DISSERTATION

ENERGY-AWARE WORKLOAD MANAGEMENT FOR GEOGRAPHICALLY DISTRIBUTED DATA CENTERS

Submitted by

Ninad Hogade

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2023

Doctoral Committee:

Advisor: Sudeep Pasricha

Howard Jay Siegel
Anthony Maciejewski
Chuck Anderson

ABSTRACT

ENERGY-AWARE WORKLOAD MANAGEMENT FOR GEOGRAPHICALLY DISTRIBUTED DATA CENTERS

Cloud service providers are distributing data centers globally to reduce operating costs while also improving the quality of service by using intelligent cloud management strategies. The development of time-of-use electricity pricing and renewable energy source models has provided the means to reduce high cloud operating costs through intelligent geographical workload distribution. However, neglecting essential considerations such as data center cooling power, interference effects from workload co-location in servers, net-metering, peak demand pricing of electricity, data transfer costs, and data center queueing delay has led to sub-optimal results in prior work because these factors have a significant impact on cloud operating costs, performance, and carbon emissions. This dissertation presents a series of critical research studies addressing the vital issues of energy efficiency, carbon emissions reductions, and operating cost optimization in geographically distributed data centers. It scrutinizes different approaches to workload management, considering the diverse, dynamic, and complex nature of these environments. Starting from an exploration of energy cost minimization through sophisticated workload management techniques, the research extends to integrate network awareness into the problem, acknowledging data transfer costs and queuing delays. These works employ mathematical and game theoretic optimization to find effective solutions. Subsequently, a comprehensive survey of state-of-the-art Machine Learning (ML) techniques utilized in cloud management is discussed. Then, the dissertation traverses into the realm of Deep Reinforcement Learning (DRL) based

optimization for efficient management of cloud resources and workloads. Finally, the study culminates in a novel game-theoretic DRL method, incorporating non-cooperative game theory principles to optimize the distribution of AI workloads, considering energy costs, data transfer costs, and carbon footprints. The dissertation holds significant implications for sustainable and cost-effective cloud data center workload management.

ACKNOWLEDGEMENTS

I am grateful to the many people who have provided consistent encouragement and important assistance in bringing this thesis to fruition.

At the fore of this journey is a person who has acted as a source of intellectual inspiration and a rock of steadfast support. Prof. Sudeep Pasricha, my advisor, deserves my heartfelt gratitude and deep admiration. His enthusiasm for the field, as well as his unique ability to demystify complex problems, sparked my own exploratory instincts. I navigated the complexities of my research with his invaluable assistance. His commitment to my personal and academic development, the time he spent developing my analytical approach, and the opportunity he gave me to achieve my true potential in achieving my Ph.D. aspirations are greatly appreciated. His method of encouraging painstaking attention to detail during my studies has considerably benefited both my academic and non-academic life. The extraordinary voyage I began under his guidance not only extended my horizons but also provided me with amazing memories and vital lessons.

I am grateful to my Ph.D. committee members, Prof. Howard Jay Siegel and Prof. Anthony A. Maciejewski. Their insightful feedback and objective perspectives were invaluable in refining my research and broadening my knowledge. Their experience and intellectual contributions considerably improved the quality of my work and pushed me to strive for academic success in the future. I'd also like to express my heartfelt gratitude to Prof. Chuck Anderson for his essential contributions as a member of my Ph.D. committee. His willingness to offer his time and skills benefited my doctoral studies and is greatly appreciated.

I discovered a group of remarkable minds in the bustling corridors of the Embedded Systems and High-Performance Computing (EPiC) research lab. Throughout this trip, the

friendship, counsel, and inventive talks with my fellow researchers have been the wind beneath my wings. I am particularly thankful to Mark Oxley, Eric Jonardi, Daniel Dauwe, Saideep Tiku, Vipin Kumar Kukkala, V. Yaswanth Raparti, Dylan Machovec, Varun Bhat, C Sai Vineel Reddy, Sai Kiran Koppu, Ishan Thakkar, Shoumik Maiti, Joydeep Dey, Febin Sunny, Asif Anwar Baig Mirza, Kamil Khan, and Ebadollah Taheri.

Without the hands-on experience I got at Fiat Chrysler Automobiles and Hewlett Packard Enterprise, I would not have been able to translate my academic knowledge into a practical realm. I am grateful to the experienced mentors at these organizations who advised me and helped shape my career path.

I am grateful to my parents, Sanjay and Sayali, for their love, unshakable faith, and unending support. This achievement is a testament to their unwavering support and faith in my talents.

Finally, I like to thank the National Science Foundation (NSF) for funding under grant CCF- 1302693, as well as Hewlett Packard for providing multiple servers. These resources have been critical to the advancement of my research.

Without the presence of all of the aforementioned individuals and institutions, this journey would not have been the enlightening and memorable experience that it was. My warmest thanks go to each of them for their substantial contributions to this pivotal moment in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF RESEARCH PUBLICATIONS

JOURNAL ARTICLES:

- **N. Hogade** and S. Pasricha, " Game-Theoretic Deep Reinforcement Learning to Minimize Carbon Emissions and Energy Costs for AI Workloads in Geo-Distributed Data Centers," *IEEE Transactions on Sustainable Computing*, June 2023, [Under Review].
- **N. Hogade** and S. Pasricha, "A Survey on Machine Learning for Geo-Distributed Cloud Data Center Management," *IEEE Transactions on Sustainable Computing*, vol. 8, no. 1, pp. 15-31, 1 Jan.-March 2022
- **N. Hogade**, S. Pasricha, and H. J. Siegel, "Energy and Network Aware Workload Management for Geographically Distributed Data Centers," *IEEE Transactions on Sustainable Computing*, Vol. 7, No. 2, pp. 400-413, June. 2021.
- **N. Hogade**, S. Pasricha, H. J. Siegel, A. A. Maciejewski, M. A. Oxley, and E. Jonardi, "Minimizing Energy Costs for Geographically Distributed Heterogeneous Data Centers," *IEEE Transactions on Sustainable Computing*, Vol. 3, No. 4, pp. 318-331, Oct.-Dec. 2018.
- C. Bash, **N. Hogade**, D. Milojicic, C. D. Patel, G. Rattihalli, "Sustainability: Fundamentals-based Approach to Paying it Forward," *IEEE Computer*, Volume: 56, Issue: 1, January 2023.

OTHER PUBLICATIONS:

- G. Rattihalli, **N. Hogade**, A. Dhakal, E. Frachtenberg, R. Pablo Hong Enriquez, Pedro Bruel, A. Mishra, and D. Milojicic, "Fine-Grained Heterogeneous Execution Framework with Energy Aware Scheduling," to appear in *IEEE Cloud*, 2023
- C. Bash, **N. Hogade**, D. Milojicic, C. D. Patel, "IT for Sustainable Smart Cities," *National Academy of Engineering, Special Issue on "Advances in Smart Cities,"* 2023.
- S. Pasricha, **N. Hogade**, H. J. Siegel, and A. A. Maciejewski, "Green Computing with Geo-Distributed Heterogeneous Data Centers," *International Green and Sustainable Computing Conference (IGSC)*, 6 pp., Oct. 2019.

1. INTRODUCTION

Cloud service providers have been resorting to the geographical distribution of data centers to intelligently distribute workloads and consequently minimize energy costs. However, this approach has given rise to a variety of challenges that span several facets of these data centers, particularly in the space of workload and resource management. This chapter provides an overview of the escalating growth of data center computing and highlights the necessity for geo-distributed data centers in the modern digital landscape. Furthermore, we delve into a thorough discussion of the challenges associated with geo-distributed cloud services and data centers, laying the groundwork for the pressing need for sophisticated workload management techniques. In addition, this chapter introduces an overview of our proposed workload management system, a solution engineered to minimize a multitude of challenges related to energy efficiency, electricity and data transfer costs, and carbon emissions in the context of geo-distributed data centers. This chapter serves as a preamble to the comprehensive contributions of this dissertation.

1.1. INCREASING INTERNET USAGE

Since the year 2000, internet usage has grown dramatically, rising by 1,355% from 2000 to 2023 [1]. The way individuals interact with one another, conduct business, and entertain, has all been dramatically impacted by this pervasive connectivity. Our reliance on the internet was further increased in 2019 because of COVID-19. The pandemic forced a shift toward eLearning, teleworking, and e-commerce, which led to an unexpected rise in internet usage. The internet's expansion has not been consistent worldwide. With over 53.1% of the world's internet users as of 2023, Asia has the highest percentage of internet users. The highest internet penetration rates, at 93.4 and 88.4%, respectively, are found in North America and Europe. With 54.4% of all traffic

worldwide coming from mobile phones, internet traffic is likewise incredibly diversified [1]. Platforms for social media and video streaming have also experienced a considerable increase in popularity.

Workloads based on artificial intelligence (AI) and machine learning (ML), as well as the Internet of Things (IoT) and edge applications, are becoming increasingly important in cloud and data center computing. The rise of AI and ML workloads in data centers is also linked to the expansion of cloud-based services. Deep learning-based services delivered via the cloud are lowering the initial costs involved with handling business operations and minimizing server maintenance responsibilities. Because of the increased demand for AI and ML-based computation, a growing number of tech behemoths and startups have begun to offer ML/AI as a cloud service

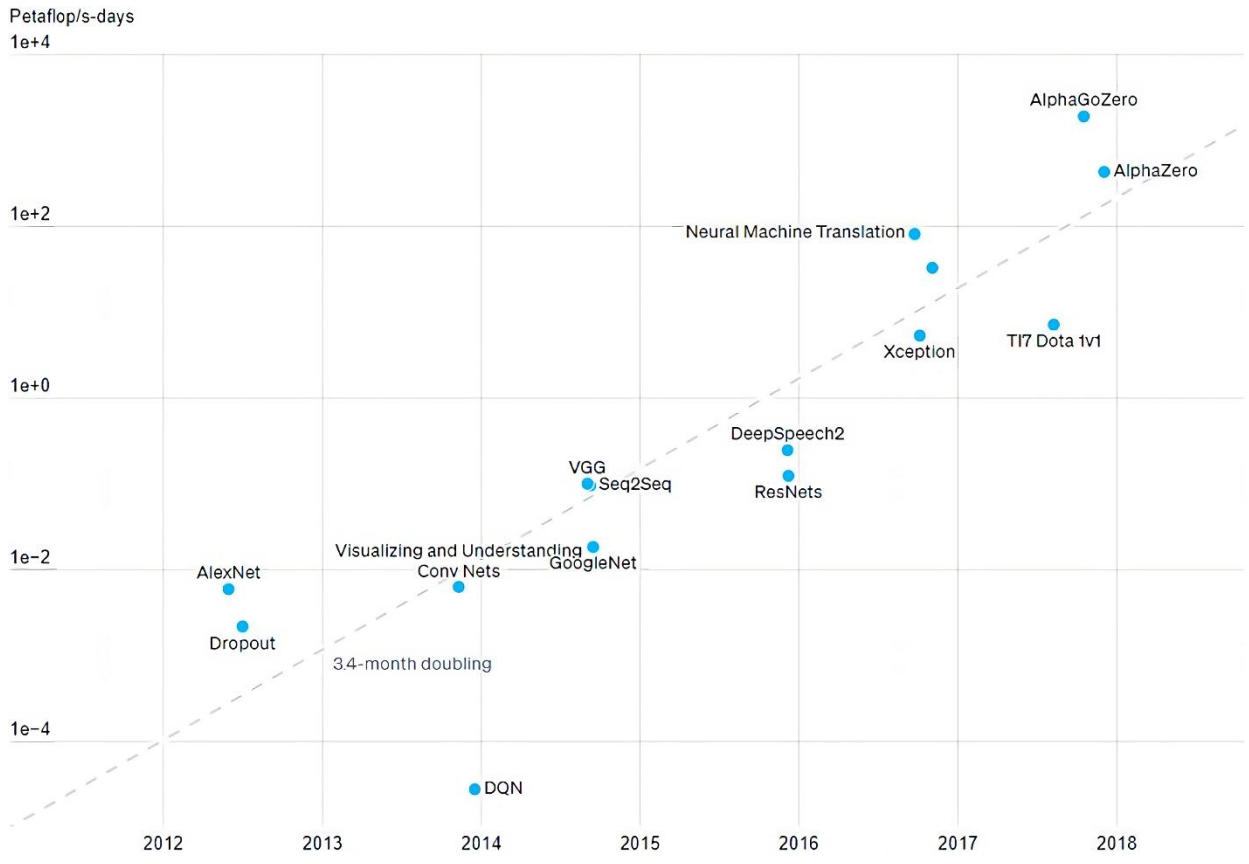


Figure 1: AlexNet to AlphaGo Zero: 300,000x increase in compute [2] (Log Scale)

[3]. The rising AI/ML workloads in data centers are also contributing to the global data center market's overall growth. The worldwide data center market was estimated at $187.35 billion in 2020 and is expected to reach $517.17 billion by 2030, growing at a 10.5% CAGR from 2021 to 2030. [4]. The amount of computing power needed to train the largest AI/ML models is growing rapidly. According to OpenAI's study [2], the amount of computation needed in the largest ML/AI training runs has been expanding exponentially since 2012, with a 3.4-month doubling time (compared to Moore's Law, which had a 2-year doubling period). This measure has increased by more than 300,000x since 2012 (a 2-year doubling period would yield only a 7x increase), as shown in Figure 1. Improvements in computation have been a significant component of AI growth, so as long as this trend continues, it's important to plan for the implications of systems that are well above today's capabilities.

Additionally, data centers have been significantly impacted by the growing number of Internet of Things (IoT) devices and applications. The volume of data collected is predicted to explode as the IoT expands, with 50% of enterprise organizations aiming to embrace IoT over the next three years [5]. Edge computing has emerged as a critical answer to this problem. In contrast to depending on a centralized data center located far away from the source of the data, edge computing puts processing and data storage closer to the site where the data is generated. This shift to edge computing allows data processing and analysis to occur in near real-time, improving both the speed and efficiency of these processes. This is especially critical for IoT applications that require quick insights from the data being collected, such as those involved in next-generation connectivity solutions like 5G networks [5]. However, the shift to edge computing does not imply that traditional data centers would be rendered useless. They will continue to play an important role in processing critical data that requires long-term storage. The data center landscape will most

certainly evolve, with a greater number of smaller data centers developed closer to population centers such as cities and business parks. This could lead to a more distributed data center infrastructure, with more storage hubs in regional markets and smaller cities, as well as mini data centers coupled with existing communications infrastructure like telecom towers [5].

1.2. GROWTH OF DATA CENTERS

Data centers house the processing power, storage, and software required to provide a variety of services, from conventional cloud storage to sophisticated machine learning software. These facilities serve as the internet's skeleton, enabling everything from streaming services to social media platforms. The need for data centers is growing exponentially along with the growth of internet traffic. The increased generation of data demands the need for reliable infrastructures for data processing and storage, which raises the need for cloud services and data centers. Because of the rising demand for data processing, digital services, data centers, and data transmission networks—the foundation of digitalization—are expanding in size and number. As a result, data centers, inter-data center data transmission networks, and other forms of resilient and dependable internet infrastructure are more important than ever. AI/ML workloads and techniques for data analytics, computational offloading of IoT data, stream data analytics, and software for IoT-edge management are among the research applications being investigated in relation to IoT-edge-cloud systems. Privacy and security of cloud systems are also important considerations. The demand for data center computing appears to be skyrocketing over the next decade. 5G, the Internet of Things, and the metaverse are all expected to drive up demand for low-latency, high-performance, and distributed computing, which will drive up demand for data centers.

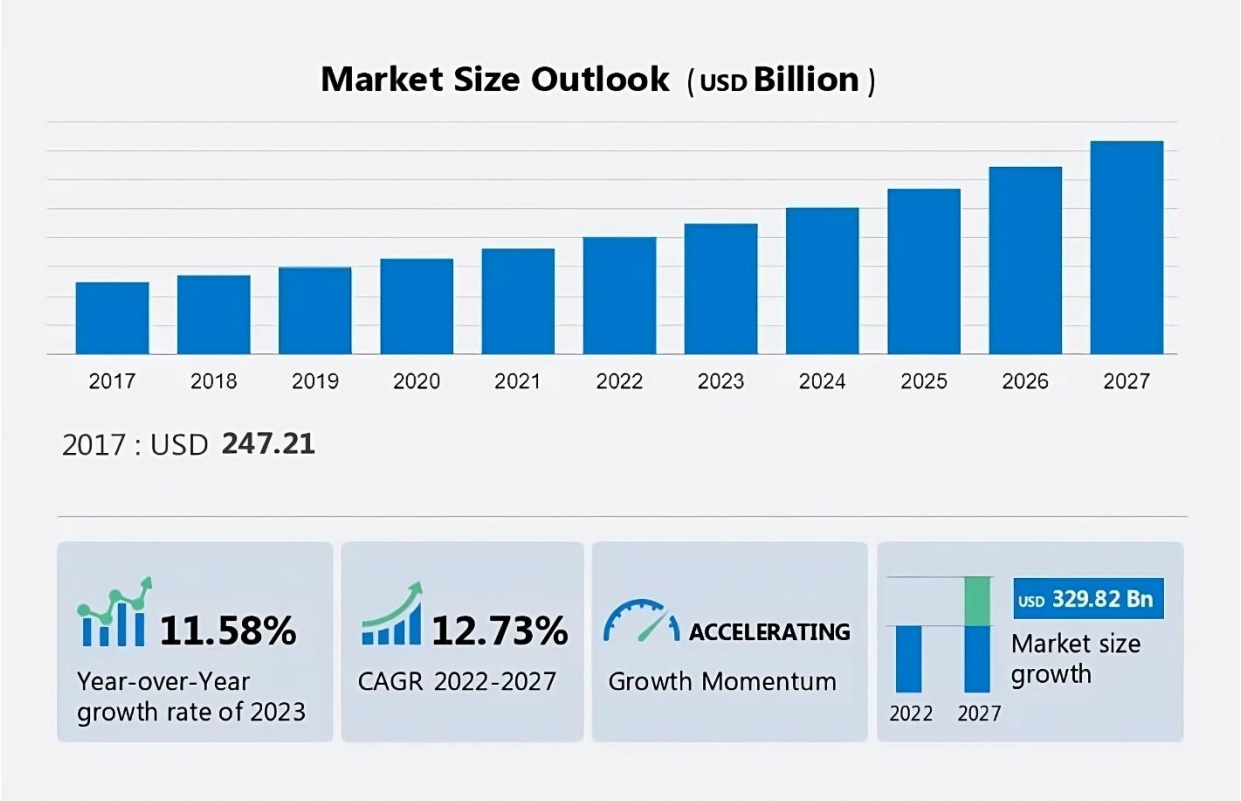


Figure 2: Global data center market revenue growth [6]

As shown in Figure 2, a recent report published by Technavio [6] shows that the data center industry is expected to grow by USD 329.82 billion at a compound annual growth rate (CAGR) of 12.73% from 2022 to 2027. This expansion is reliant on the increased adoption of multi-cloud and network upgrades to support 5G, demand planning and expansion by hyperscale data centers, and rising demand among small and medium-sized enterprises. One of the key drivers of this expansion is the adoption of multi-cloud and network improvements to enable 5G, which is driving a rise in demand for data centers. Furthermore, the global data center market experienced extraordinary demand in 2021-2022 as firms raised investments in IT infrastructure upgrades and shifted to cloud services [6]. There is also a strong trend of rising enterprise adoption of High-Performance Computing (HPC). These systems are mostly utilized for scientific and engineering purposes, and

their increasing popularity is driving up demand for high-performance data centers. The IT infrastructure category is the fastest-growing segment in the data center market, owing to an increased need for processing power and storage to accommodate increased data traffic. Enterprises throughout the world are embracing cloud technology and migrating their data from on-premises to cloud-based data centers [6].

1.3. COMPONENTS OF A DATA CENTER

A data center is a facility that houses an organization's shared IT operations and equipment in order to store, process, and distribute huge amounts of data. It is comprised of multiple critical components that work together to ensure the organization's requirements. Figure 3 shows the layout and essential components of a typical data center. All these components work in concert to

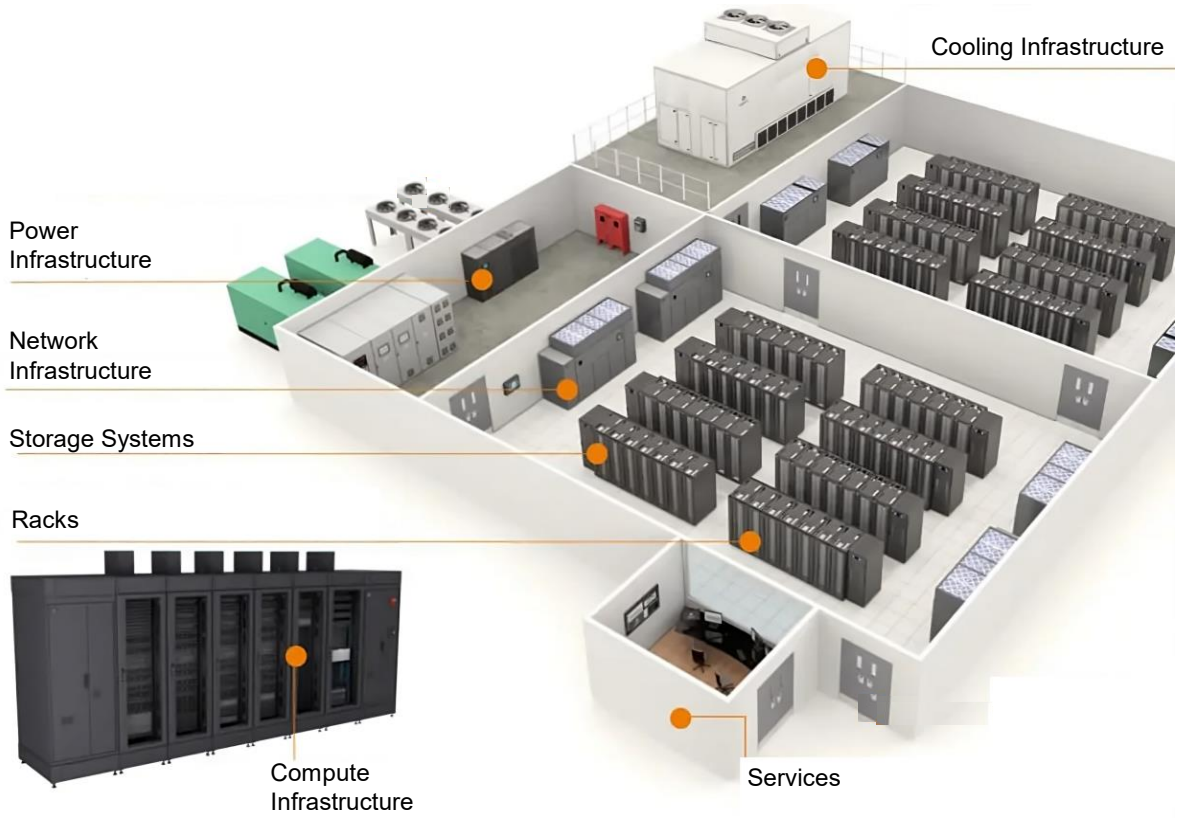


Figure 3: Layout of a typical data center and its key components

provide a robust, efficient, and secure environment for the storage, processing, and transmission of data. The exact composition and configuration of these components can vary widely depending on the specific needs and resources of the organization operating the data center. We explain these components in the following subsections.

1.3.1. COMPUTE INFRASTRUCTURE

The computing infrastructure is the foundation of any data center, providing the processing power required for a wide range of applications, from simple data storage to complex machine learning operations. It is mostly made up of servers, which are divided into two types: physical and virtual. Physical servers are self-contained machines with specialized resources such as CPU, memory, and storage. They are highly adjustable, allowing them to be tailored to individual performance needs. Physical server management, on the other hand, can be complex and costly, and they may not fully utilize their resources. Virtual servers, often known as virtual machines (VMs), are software versions of actual servers. Because of virtualization technology, multiple VMs can run on a single physical server. Each VM functions as a standalone machine, along with its own operating system and applications. Virtual servers enable more efficient resource utilization, better scaling, and simpler management. Processors, memory, and operating systems are also part of the computational infrastructure, which enables data processing and task execution. To create and run VMs in a virtualized environment, a specific sort of operating system known as a hypervisor is used. The computational infrastructure is a data center's powerhouse, providing the resources needed to process and alter data. Its configuration and management have a substantial impact on the data center's performance and efficiency.

1.3.2. COOLING INFRASTRUCTURE

A data center's cooling system is designed to control and regulate the heat produced by the servers and other hardware components in order to ensure optimal performance and lifetime. Maintaining the proper temperature and humidity levels is critical to avoiding overheating, which can result in hardware failure and data loss. In data centers, various types of cooling systems are routinely employed. The most traditional cooling method is air-based cooling. Computer Room Air Conditioners (CRACs) or Computer Room Air Handlers (CRAHs) are used to cool the air around the server racks. The hot air is drawn away from the equipment, cooled, and recirculated. This technology can be improved further by employing techniques such as hot aisle/cold aisle confinement, which effectively separates the cold supply air and the hot exhaust air from the servers. In more advanced methods, like liquid cooling, a coolant liquid is used to absorb the heat produced by the servers. This can be accomplished through either direct liquid cooling, in which the coolant comes into direct contact with the server components, or indirect liquid cooling, in which the coolant is used to cool the air surrounding the components. Liquid cooling is more efficient than air cooling and is frequently utilized in high-density data centers with large heat burdens. The free cooling method, also known as economization, uses external ambient variables to aid in cooling. For example, during the colder months, outside air can be used to cool the data center or adjacent bodies of water can be used as a source of cool air. This strategy can significantly cut energy usage and expenditures, but it is highly reliant on local climate conditions. In addition to these technologies, data centers use sophisticated monitoring and control systems to constantly monitor temperature and humidity levels and modify cooling systems as needed. These technologies can also warn operators of possible faults before they become critical, ensuring the reliability and longevity of the data center's equipment. All of these cooling solutions are intended

to improve energy efficiency, lower costs, and reduce the environmental effect of data center operations. The cooling infrastructure used is determined by a number of factors, including the size and density of the data center, the local climate, and the organization's resources and priorities.

1.3.3. NETWORKING INFRASTRUCTURE

A data center's networking infrastructure, a critical component, permits communication, data sharing, and management both within the data center and with external systems. Most enterprise networks are built around switches. They act as the controller, allowing networked devices to communicate with one another efficiently. Switches in a data center can be used to connect servers to one other or to the network, and they can support a variety of protocols and speeds. Routers are devices that transport data packets across computer networks. Routers are typically used in data centers to connect the data center network to external networks, such as the internet. A firewall is a network security system that monitors and controls network traffic based on predefined security rules. It acts as a shield between a trusted internal network and an untrustworthy external network. Load balancers distribute network traffic among numerous servers, ensuring that no single server is overburdened. This contributes to increased application availability and dependability. Network Interface Cards (NICs) are pieces of hardware that link a device to a network. Each server in a data center will normally have one or more NICs to connect to the network. Network Management Software aids in network management and monitoring. It can provide performance insights, assist in identifying and troubleshooting issues, and manage network settings and security. Cabling & Infrastructure encompasses all of the physical cables and connections that connect the network, as well as the racks, cabinets, and paths that contain and arrange network equipment. Network security systems, in addition to firewalls, include intrusion

prevention systems (IPS), virtual private networks (VPN), and other tools that help secure the network and data from threats. To manage enormous levels of data traffic, offer high availability, and support the data center's growth and evolution, the networking infrastructure must be strong, redundant, and scalable. It is an essential component in ensuring that the data center can support an organization's IT activities.

1.3.4. STORAGE SYSTEMS

Storage systems are essential components of a data center since they are in charge of storing, retrieving, and protecting data. They can come in many different forms. Direct-Attached Storage (DAS) is the most basic sort of storage, in which storage devices are linked directly to a server or a PC. While DAS is simple and inexpensive, it does not scale well, and data saved on one server may be inaccessible to other servers. Network-Attached Storage (NAS) devices are attached to a LAN and provide file-level storage services to other network devices. They provide a quick and easy way to consolidate and exchange data over a network and are commonly used for archiving and backup, media streaming, and collaboration settings. Storage Area Networks (SANs) are high-speed networks of storage devices that numerous servers can access. They offer block-level storage, which is faster and more adaptable than file-level storage. Because SANs are highly scalable and perform well, they are ideal for data-intensive applications and large-scale virtual environments. Object Storage is a newer type of data storage that handles data in the form of objects rather than files or blocks. Each object contains data, varying amounts of metadata, and a unique identity. Object storage is extremely scalable, making it ideal for storing massive amounts of unstructured data. Storage systems in a data center comprise, in addition to physical storage devices, software for managing and organizing data, as well as technologies for ensuring data

protection and integrity, such as RAID (Redundant Array of Independent Disks), backup and replication, and data deduplication. A multitude of factors influences storage system selection, including data volume and type, performance requirements, scalability requirements, and budget limitations.

1.3.5. POWER INFRASTRUCTURE

A data center's power infrastructure is intended to provide a continuous and stable power supply to all operating elements within the facility. The primary source of power for the data center is often supplied from the local grid. Some data centers, particularly those that are concerned with sustainability, may also use renewable energy sources such as solar or wind power. When the primary power source fails, an uninterruptible power supply (UPS) provides backup power. A UPS protects against input power interruptions very instantly by supplying energy stored in batteries, flywheels, or super capacitors. The length of power delivery by a UPS is typically short yet sufficient to transition to a backup power source. Backup Generators are utilized in the event of a lengthy power outage. Diesel or gas generators can offer power for a lengthy period of time. When the UPS's energy source is exhausted, these generators kick in. Electricity Distribution Units (PDUs) are in charge of delivering power to the various components of the data center. They take power from the UPS or generator and transfer it to the servers, cooling systems, and other equipment. Redundancy is an important feature of electricity infrastructure to ensure ongoing functioning. This may necessitate numerous power feeds, UPS systems, and generators. The design and management of power infrastructure require balancing the necessity for reliability and uptime with the need for cost reductions and energy efficiency.

1.3.6. MANAGEMENT AND SERVICES

A data center's services component encompasses a diverse set of software solutions and techniques that ensure the facility's efficient, secure, and smooth functioning. Management software such as Data Center Infrastructure Management (DCIM), which monitors and controls physical infrastructure, and Virtual Machine Management (VMM), which manages virtualized resources, are examples of this. Physical and cybersecurity measures are used to secure data centers. Physical security measures may include regulated access to the site, surveillance systems, and alarm systems, whereas cybersecurity measures may include firewalls, intrusion detection systems, secure access controls, encryption, and regular security audits. Furthermore, disaster recovery and business continuity planning (BCP) are critical since they ensure the data center's ability to recover from unforeseen occurrences such as natural disasters, power outages, or cyber-attacks. Data backup and replication, redundant systems, remote storage, and extensive recovery plans are examples of these solutions. IT service management (ITSM) refers to an organization's efforts in designing, planning, delivering, operating, and controlling IT services, with ITSM solutions commonly used in data centers to manage incident response, problem management, and change management, among other things. Automation and orchestration solutions are extremely important in data centers because they automate routine tasks and orchestrate complicated processes, increasing efficiency, decreasing the risk of human error, and freeing up IT professionals to focus on more strategic responsibilities. Finally, many data centers must adhere to numerous industry standards concerning data security and privacy, and compliance solutions can automate compliance reporting and monitor for any compliance concerns. All of these services are critical to the seamless operation of a data center, helping to maximize efficiency, provide security, maintain uptime, and manage the center's complex array of resources.

1.4. TYPES OF DATA CENTERS

There are several types of data centers, each with its own distinct qualities and functions, as discussed in the subsections below.

1.4.1. ENTERPRISE DATA CENTERS

Enterprises construct and run these facilities to meet their data processing and storage requirements. They are typically found on corporate campuses and are intended to serve a single organization. They are distinguished more by their ownership and purpose than by their size and capacity. Enterprise data centers are comprised of multiple sub-data centers, which can include an Internet data center (which supports the smooth operation of web applications), an Extranet data center (which supports business-to-business transactions within the enterprise data network), and an Intranet data center (retains the application and data within the data center) [7].

1.4.2. MANAGED SERVICES DATA CENTERS

In this model, a third-party data center service provider deploys, manages, and monitors the data center. They offer features and functionality similar to those of a traditional data center but through a managed service platform. Depending on the agreement between the business and the service provider, the level of management can be totally or partially delegated. For example, IBM's data center provides a wide range of managed services directly to its clients, such as security and managed network services [8] .

1.4.3. COLOCATION DATA CENTERS

Various firms needing colocation services rent space from colocation providers and provide management components such as servers, storage, and firewalls in this setup. The infrastructure is provided by colocation data centers, which include buildings, cooling, bandwidth, security, and other services. Businesses have some administrative control over data center deployment and service, and the majority of management can be done remotely [8].

1.4.4. EDGE DATA CENTERS

These are modest facilities positioned near the people they serve. Organizations can deliver information and services with little latency thanks to these data centers. They are typically located on the premises of the firm, making them similar to onsite data centers. These facilities are typically managed by an offsite company or a colocation provider. They are key components of edge computing architecture, bringing data storage and computation closer to the point of use. This sort of data center is highly compact and end-user customizable, making it perfect for supporting IoT and the autonomous car segment in order to increase processing capacity and improve customer experience [9].

1.4.5. HYPERSCALE OR CLOUD DATA CENTERS

Hyperscale data centers, commonly referred to as cloud data centers, are massive, custom-built facilities run by a single corporation. These data centers generally serve cloud service providers and huge internet enterprises that have large computation, storage, and networking requirements. They contain thousands of racks and servers and range in size from 50,000 to over 1 million square feet. This is a completely virtualized data center based on cloud computing

architecture. Companies can use cloud service providers to host both data and applications. The cloud provider monitors and maintains the physical hardware with the assistance of a third-party managed services provider. It allows clients to run apps and manage websites and data within the confines of a virtual infrastructure hosted on cloud servers. The corporation simply has to pay for the hardware resources that are used, and there is no need to worry about frequent server updates, security features, cooling costs, and so on. The monthly subscription includes the pricing for all services. It can be used by organizations of any size [8], [7], [9].

1.5. NEED FOR GEOGRAPHICALLY DISTRIBUTED DATA CENTERS

The proliferation of cloud services is likely to fuel future data center expansion. This is owing to increased demand for real-time applications, the proliferation of Internet of Things (IoT) devices, and the introduction of new technologies such as 5G. Amazon Web Services, Microsoft Azure, and Google Cloud are making significant expenditures to increase their data center infrastructure, as shown in Figure 4. The geographical spread of these data centers enables speedy data access from any place, boosting the user experience.

Figure 4: Cloud Data Center Locations Worldwide (2016) [10]

The need for geographically distributing data centers can be broken down into several key areas, as below.

1.5.1. DISASTER RESILIENCE AND BUSINESS CONTINUITY

To provide resilience and uninterrupted service, cloud service providers are increasingly distributing their data centers across several geographical locations. In the event of a natural disaster, regional power outage, or other disruption that causes a specific data center to fail, the cloud infrastructure's distributed design allows traffic to be diverted to functional data centers in other locations. This assures business continuity for clients whose operations rely on cloud service availability.

1.5.2. LOWER LATENCY

Customer experience is critical for cloud service providers. The latency experienced by end users is a critical part of this. By geographically scattering data centers, operators can ensure that users are served from the closest available region, reducing latency. This is crucial for real-time interaction-demanding applications such as online gaming, video conferencing, and real-time analytics and can dramatically improve user experience.

1.5.3. SERVING LOCATION-SPECIFIC CONTENT

Cloud service companies frequently serve a global customer base with diverse needs. By geographically dispersing data centers, suppliers can efficiently store and offer location-specific content. This enables services to be tailored to regional tastes, languages, prices, and even content

laws. It also allows you to provide content that is appropriate to a specific geographical audience, improving the user experience and satisfying a variety of customer needs.

1.5.4. COMPLIANCE WITH DATA PROTECTION REGULATIONS:

A diversity of regional and national laws and regulations, such as the European Union's General Data Protection Regulation (GDPR) [11], characterize the data governance environment. These rules frequently require that specified types of data be retained and handled within specific geographical borders. To comply with these requirements and provide compliant services to their clients, cloud providers geographically spread their data centers, ensuring that data is stored and processed in the right region.

1.6. CHALLENGES WITH GEO-DISTRIBUTED DATA CENTERS

Despite the apparent benefits, it's worth noting that managing geographically distributed data centers is a complex task. This section discusses several key challenges in geo-distributed data center management that we are trying to address in this dissertation.

1.6.1. HETEROGENEOUS WORKLOAD AND HARDWARE

The disparities in hardware and workloads in geographically distant data centers bring new issues. Different types and amounts of computer resources are required by a diverse range of applications and workloads. A mismatch between workload requirements and hardware capabilities can result in inefficient resource utilization, performance deterioration, or even service failure [12]. The performance and resource requirements of tasks can vary depending on the type of hardware on which they are executed. This requires sophisticated scheduling algorithms that

consider the specific characteristics of both the tasks and the available hardware. For instance, certain workloads may be more suited for servers with high-frequency CPUs, while others may require large amounts of memory [12]. The underlying hardware can have a significant impact on application performance. Because of differences in processor capabilities, storage access speed, and network bandwidth, heterogeneous hardware combinations can cause application performance to vary. As a result, application performance may be variable, affecting the quality of service given to end users [13].

1.6.2. HIGH POWER CONSUMPTION AND ELECTRICITY COSTS

A large-scale data center typically consumes between 20 and 50 MW of power each year, which is potentially enough to power 37,000 households [14]. In recent years, the digitization of services has accelerated, increasing the energy consumption rates of the Internet and cloud data centers. It is important to highlight that the expansion of data centers and data transmission networks presently accounts for around 3% of global electricity consumption, with forecasts indicating that this figure could rise to 4% by 2030 [15]. Nevertheless, electricity consumption in this sector has surged by 10-30% annually in recent years due to the rapid expansion of the workloads managed by major data centers [16]. For instance, server power consumption witnessed a massive increase of 266% in 2017, fueled by a surge in server electricity use as advancements in chip design and manufacturing were reaching their limits [17]. Moreover, computational demand for AI/ML training (up 150% per year) and inference (up 105% per year) at Meta (formerly Facebook) has exceeded total data center energy use (up 40% per year) [18]. Google reports that AI/ML accounted for 10-15% of total data center energy utilization, although accounting for 70-80% of overall compute demand [19].

However, the landscape of data center energy use differs greatly between countries. While the global trend in data center energy consumption has been low, countries with growing data center economies have seen exponential demand. Ireland, for example, has seen data center energy demand quadruple since 2015, accounting for 14% of total electricity consumption by 2021. Similarly, in Denmark, data center energy use is predicted to quadruple by 2025, accounting for around 7% of total electricity consumption [20].

It's clear that the escalating power consumption of data centers has substantial implications for electricity costs. Given the steep trajectory of energy use in this sector, exploring more efficient hardware, improving energy management, and implementing innovative cooling technologies are becoming increasingly crucial.

1.6.3. INTER-DATA CENTER NETWORK TRAFFIC AND COSTS

Inter-data center communication is the backbone of today's cloud-based services. It allows for the coordination and synchronization of massive volumes of data across numerous geographically distributed centers. An enormous increase in data generation has been caused by the growing reliance on digital services and the resulting internet traffic. Data center workloads have significantly increased as a result of the 20-fold increase in global internet traffic and the more than two-fold increase in internet users since 2010 [16]. In the context of such enormous data growth, geo-distributed data centers transport hundreds of petabytes of data via wide area networks (WANs). Figure 5 shows Microsoft Azure’s WAN infrastructure across the globe [21].

According to estimates, five more bits of data are transported between and within data centers for every bit of data sent from a data center to end users, illustrating the complexity of these activities [20]. As data transfers increase, so does their energy consumption. According to

the International Energy Agency (IEA) 2022 report, data transmission networks accounted for 1.1-1.4% of global electricity consumption [16]. This growth in consumption is driven by increased demand for internet services, a rise in the volume of data being transferred, and the energy requirements of the associated infrastructure, including servers, routers, switches, and undersea cables [22].

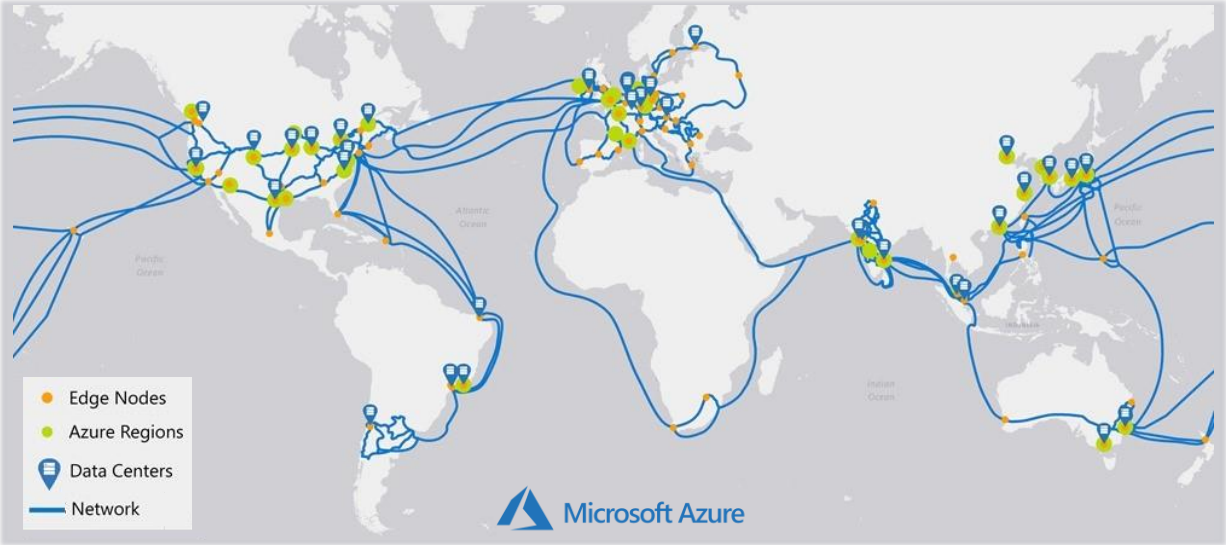


Figure 5: Microsoft Azure's WAN infrastructure (2017) [21]

While efforts are being made to improve energy efficiency through hardware and software innovations, as well as infrastructure improvements, the ongoing growth in global data demand - owing to trends such as IoT, video streaming, and cloud computing - suggests that the energy consumption of data transmission networks will remain a critical concern for the foreseeable future.

1.6.4. CARBON EMISSIONS

The global development of data centers, as well as their growing significance in our daily lives and economies, has raised substantial sustainability concerns. It is vital to remember that data

centers contribute to around 2% of total global greenhouse gas (GHG) emissions [16]. This figure is expected to climb to 5-7% of worldwide emissions in the aftermath of the recent epidemic, a troubling trajectory [16]. In order to boost efficiency and reduce environmental impact, many firms have focused on their data center and data transmission network infrastructure, which is a key contributor to energy use and GHG emissions. The data center industry is anticipated to cut emissions in half by 2030 to meet Net Zero goals [16]. In the context of continuous climate change, rising energy and water consumption pose increasingly pressing issues. To limit data center consumption and carbon footprint, the industry is projected to confront tougher regulations and increasing third-party oversight.

Balancing the increasing demand for digital services with the need for sustainability is a complex task that requires a multifaceted approach in geo-distributed cloud management, including technological innovation, regulatory measures, and corporate responsibility.

1.7. DISSERTATION OVERVIEW

Cloud service providers and researchers in this area are looking to develop an intelligent workload management system to address the aforementioned challenges. A system of this type should be capable of effectively scheduling workloads based on their different requirements, optimizing resource allocation across heterogeneous hardware, and efficiently managing inter-data center traffic to reduce data transmission costs. Furthermore, it should be capable of controlling power consumption, lowering electricity prices, and carbon emissions. This motivation originates from the desire to provide a solution that supports not just performance but also sustainability and cost-effectiveness. Meeting these challenges will undoubtedly be difficult, but the creation of a smart workload management system could be a significant step in mitigating these issues, resulting

in a more efficient and sustainable cloud architecture. Adoption and implementation of such systems have the potential to drastically alter the operations of geographically distributed data centers, making this an area ripe for investigation and discovery.

To address the challenges presented in the previous sections, in this dissertation, we propose a holistic workload management system for geographically distributed data centers, as shown in Figure 6. The figure shows inputs, output, and various resource models used along with their constraints. Our goal is to apply the insights gained from four major research chapters that address the aforementioned concerns utilizing a variety of strategies and heuristics.

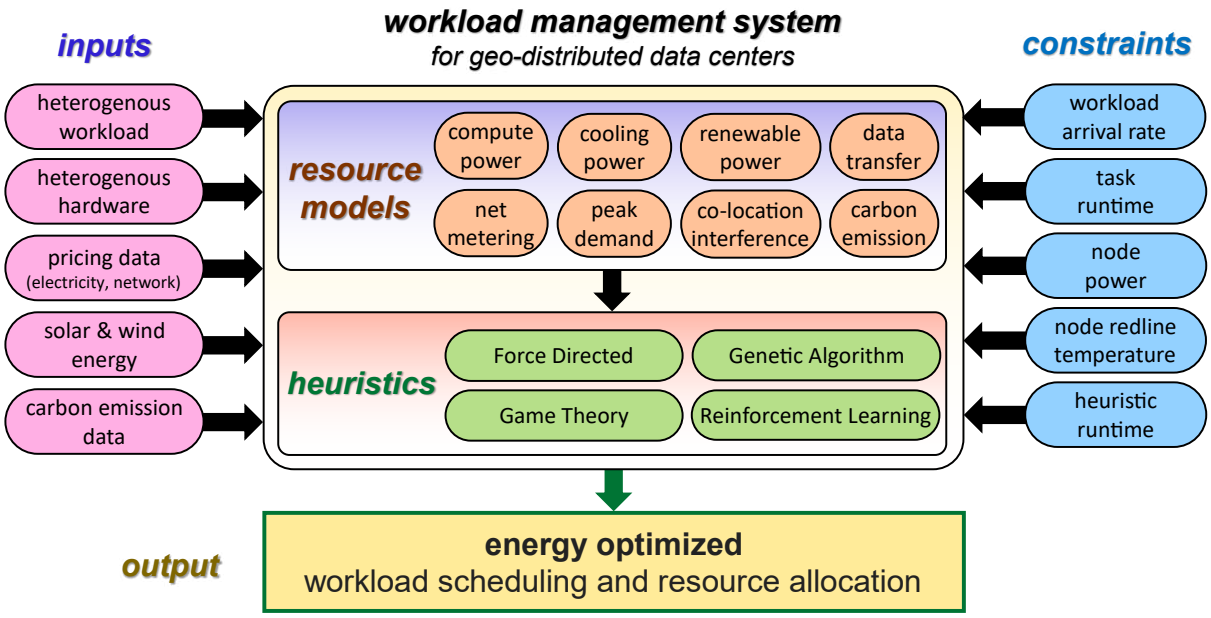


Figure 6: Overview of our workload management system for geo-distributed data centers

The second chapter, "Minimizing Energy Costs for Geographically Distributed Heterogeneous Data Centers," introduces the reader to the complexities of dealing with heterogeneous workloads and technology. We dive deeply into the development of workload management strategies that account for a variety of essential elements such as cooling power, co-

location interference, time-of-use electricity pricing, renewable energy availability, net metering, and peak demand pricing. These strategies set the framework for the intelligent workload management system presented in this research by optimizing resource allocation and effectively managing heterogeneous hardware.

In the third chapter, "Energy and Network Aware Workload Management for Geographically Distributed Data Centers," the research is expanded by adding the concept of game theory to address the issue of controlling inter-data center traffic and data transmission costs. We present a comprehensive game theory-based workload management paradigm that takes into account not just energy costs but also data transfer costs along with data center queuing delay. The insights gained from this work give realistic solutions for minimizing data transmission costs and energy efficiently managing geo-distributed data centers, thereby significantly contributing to the dissertation's goals.

The fourth chapter, "A Survey on Machine Learning for Geo-Distributed Cloud Data Center Management," provides an in-depth look at how machine learning (ML) techniques can be applied to the complexity of managing geo-distributed data center operations. This chapter emphasizes how machine learning methods can outperform traditional techniques in confronting the challenges of managing workload and hardware heterogeneity, controlling high power consumption, and reducing data transmission costs by examining the benefits and drawbacks of current ML-based techniques for cloud management.

The fifth chapter, "Game-Theoretic Deep Reinforcement Learning to Reduce Carbon Emissions and Energy Costs for AI Workloads in Geo-Distributed Data Centers," ties all of the preceding chapters together. It devises a strategy to optimize workload distribution by combining

non-cooperative game theory and deep reinforcement learning. This method reduces carbon emissions and operational expenses while maintaining performance.

These chapters work together to provide a thorough examination of the various problems connected with managing geo-distributed data centers. Each chapter provides a step toward the ultimate objective of creating an intelligent task management system capable of addressing the identified challenges. As a result, our dissertation synthesizes these findings to propose a holistic solution that promises both energy efficiency and sustainability in geo-distributed data centers. Finally, in the final chapter, we give a thorough summary of our research, accomplishments, and lessons learned, incorporating the insights and techniques presented in each chapter. The novel solutions and methodologies offered here not only add to the existing cloud computing ecosystem but also open up new opportunities for future research. Thus, we also include future work directions, acknowledging that the quickly growing nature of cloud computing technology demands ongoing research and adaptation. The potential revolution that these solutions can bring to the operations of geographically distributed data centers highlights the importance of this dissertation in the field of cloud computing.

2. MINIMIZING ENERGY COSTS FOR GEOGRAPHICALLY DISTRIBUTED HETEROGENEOUS DATA CENTERS[1]

The recent proliferation and associated high electricity costs of distributed data centers have motivated researchers to study energy-cost minimization at the geo-distributed level. The development of time-of-use (TOU) electricity pricing models and renewable energy source models has provided the means for researchers to reduce these high energy costs through intelligent geographical workload distribution. However, neglecting important considerations such as data center cooling power, interference effects from task co-location in servers, net-metering, and peak demand pricing of electricity has led to sub-optimal results in prior work because these factors have a significant impact on energy costs and performance. We propose a set of workload management techniques that take a holistic approach to the energy minimization problem for geo-distributed data centers. Our approach considers detailed data center cooling power, co-location interference, TOU electricity pricing, renewable energy, net metering, and peak demand pricing distribution models. We demonstrate the value of utilizing such information by comparing it against geo-distributed workload management techniques that possess varying amounts of system information. Our simulation results indicate that our best-proposed technique is able to achieve a 61% (on average) cost reduction compared to state-of-the-art prior work.

2.1. MOTIVATION AND CONTRIBUTION

The success of cloud computing has resulted in data center operators expanding and geographically distributing their data center locations, e.g., Google [23] and Amazon [24].

[1] The research presented in this chapter is published in [107].

Distributing data centers geographically offers several benefits to the clients such as low latency due to shorter communication distances and service resiliency. A strong motivating factor for data center operators to geographically distribute their data centers is to reduce operating expenditures by exploiting time-of-use (TOU) electricity pricing [25]. Electricity prices are not constant, but rather follow a TOU pricing model where the cost of electricity varies based on the time of day as demonstrated in Figure 7. Electricity prices are higher when total electrical grid demand is high, and fall during periods when electrical grid demand is low [26], [27]. Beyond the TOU electricity costs, most utility providers also charge a flat-rate (peak demand) fee based on the highest (peak) power consumed at any instant during a given billing period, e.g., month [28], [29]. Reducing electricity costs has been a major focus of data center management, and has continued to grow in importance as the annual electricity expenditure for powering data centers has, in some cases, surpassed the costs of purchasing the equipment itself [30].

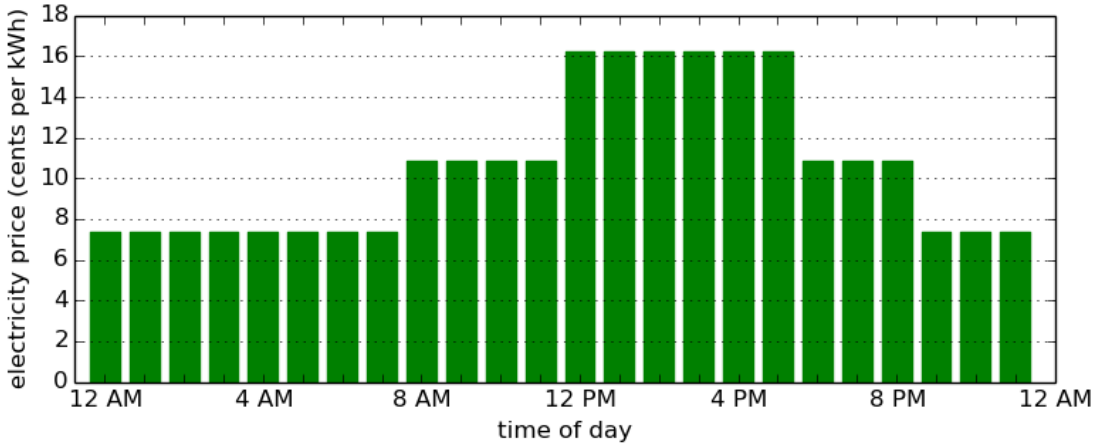


Figure 7: TOU electricity pricing, PG&E Schedule E-19 [31]

Relocating workloads among geo-distributed data centers is one effective approach to curb electricity expenditures. Workloads can be migrated to data centers located in different times zones with the goal of concentrating the workload in regions with the lowest TOU electricity and peak

demand pricing available at that time. The allocation of workloads within a data center can also reduce electricity cost by exploiting dynamic voltage and frequency scaling (DVFS) and any heterogeneity across compute nodes (e.g., different power and performance characteristics).

Due to the ever-increasing electricity consumption of data centers, the use of on-site renewable energy sources, e.g., solar and wind, has grown rapidly in recent years. Several data center operators have already built or announced plans to build "green" data centers, i.e., data centers, that are completely (or at least partially) operated with the help of renewable energy. For example, a portion of Apple's data centers in North Carolina are powered by a 60 MW solar plant [32]. McGraw-Hill operates a data center using a 14 MW solar array [33]. Similar to these examples, major global data center providers, e.g., Microsoft [34], Google [35], and Facebook [36], have invested in green energy facilities. Some data center providers have begun to use locations with large amounts of renewable energy available to reduce energy costs, or even exploit net metering. *Net metering* is a billing mechanism that gives renewable energy customers credit on their utility bills for the excess clean energy they sell back to the grid [37]. Adding on-site renewable power and exploiting net metering can reduce electricity costs [38], peak grid power costs [39], or both [40], [41]. This can provide additional opportunities for geographical load distribution (GLD) techniques to reduce overall electricity costs.

In this chapter, the goal of our research is to design techniques for geographical load distribution that will minimize the energy cost for executing incoming workloads considering many aspects of the over- all system. We use detailed models for data center cooling power, co-location interference, TOU electricity pricing, renewable energy, net metering, and peak demand pricing distribution to provide the most-accurate information possible to the geo-distributed workload manager. Co-location interference is a phenomena that occurs when multiple cores

within the same multicore processor are executing tasks simultaneously and compete for shared resources, e.g., last-level cache or DRAM. Our research is highly useful for environments where historical execution information about the types of tasks executed is readily available. Examples of such environments exist in industry, e.g., commercial companies (DigitalGlobe, Google), military computing installations (Department of Defense), and government labs (National Center for Atmospheric Research).

By considering data center cooling power, co-location interference, TOU electricity pricing, renewable energy, net metering, and peak demand pricing distribution models, we design three new workload management techniques. These techniques assume varying degrees of co-location interference characteristics to distribute or migrate the workload to low-cost data centers at regular time intervals, while ensuring that all of the workload is completed. We compare these techniques to a state-of-the-art method [42], and our previous work [43], and show that our best proposed resource management heuristic can, on average, achieve a cost reduction of 61%. Key contributions in our research can be summarized as follows:

- a hierarchical framework for the GLD problem that considers cost-minimization oriented workload management at both the geo-distributed and local heterogeneous data center level;
- a new detailed data center model that exploits net metering and peak shaving, i.e., peak power reduction, by considering information about heterogeneous compute node-types, P- states, compute node temperatures, cooling power, TOU and peak demand pricing, renew- able power sources, and performance degradation caused by co-location interference;
- the design of three resource management heuristics that possess varying degrees of co- location interference prediction information to demonstrate and motivate the use of detailed models in workload management decisions.

2.2. RELATED WORK

There have been many recent efforts proposing methods to minimize electricity costs across geo-distributed data centers, with the fundamental decisions of the optimization problem relying on a TOU electricity pricing model [44]. These models either use data analytics for pricing models or make predictions of electricity costs. Electricity costs are often much higher during peak hours of the day (typically 8 A.M. to 5 P.M.). The TOU electricity cost models, sometimes in combination with a model for revenue generated from computation of a workload, motivates the use of optimization techniques to minimize energy cost or, if provided a revenue model, to maximize total profit.

Workload distribution for geo-distributed data centers has been studied in prior work such as [42], [43], [45], [46], [47], [48], [49], [50]. Information about TOU pricing is typically used to either minimize electricity costs across all geo-distributed data centers, e.g., [42], [43], [45], [47], [48], [49], [50], or to maximize profits when a revenue model is included for computational work performed, e.g., [46]. A quality of service (QoS) constraint of some form is recognized in most of the aforementioned works. Typically, this is incorporated into the model as a queuing delay constraint [45], [47], [49]. Other works incorporate QoS violations into a cost function, where a monetary penalty is associated with violating service level agreements (SLAs) due to excessive queuing delay [46], latency [48], or migration penalties [42]. The detail of each model varies significantly among approaches in the prior work. Some works include DVFS in decision making [47], some include power consumption of the cooling system in addition to the computing system [48], [49], and others consider real-world TOU pricing data [46], [47].

Our research considers all of the aforementioned modeling aspects to assist in workload management decisions: (a) DVFS to exploit the power/performance trade-offs of P-states; (b) cooling system power and thermal properties to reduce cooling cost; (c) TOU and peak demand pricing data to reduce monetary electricity cost. To the best of our knowledge, our research is the first to integrate all of these aspects within the GLD problem. In addition, unlike any prior work in GLD, we consider performance degradation caused by co-location interference as part of our load distribution techniques.

With respect to energy usage, some studies have considered renewable energy for energy optimization of geo-distributed data centers [42], [43], [51], [52]. Similar to [31], our research uses information about TOU electricity pricing, peak demand pricing and peak shaving, renewable energy usage, and net metering. However, reference [31] proposes a solution for a single data center rather than for a group of geo-distributed data centers. Moreover, it does not consider heterogeneous compute node-types, different performance states of cores, compute node temperatures, cooling power, or co-location interference. Our research considers all these factors at a geo-distributed level. Similar to our work, reference [51] proposes a workload scheduling technique that uses both peak shaving and renewable energy prediction models. However, it does not consider net metering and detailed cooling power and co-location interference models. Similar to [42] and our prior work in [43], our study considers a renewable energy source at each geo-distributed data center, a cooling system at each data center, and migration penalties associated with moving already-assigned workloads to different data centers. We differ significantly from [42] by including TOU electricity pricing traces, considering DVFS P-state decisions, and integrating interference caused by the co-location of multiple tasks to cores that share resources in

our management techniques. We extend our prior work significantly from [43] by including a peak demand price model and also exploiting peak shaving and net metering.

2.3. SYSTEM MODEL

2.3.1. OVERVIEW

We propose a hierarchical framework for a geo-distributed resource manager (GDRM) that consists of a high-level manager to distribute incoming workload requests and migrate already-allocated requests to geographically distributed data centers. The goal of the GDRM is to minimize the total energy cost of the system while servicing all requests. Each data center has its own local workload management system that takes the workload assigned to it by the GDRM and maps requests to compute cores within the data center. We first describe the system model at the geo-distributed level and then provide further details into the models of components at the data center level.

2.3.2. GEO-DISTRIBUTED LEVEL MODEL

We consider a rate-based workload management scheme, where workload arrival rate can be predicted over the decision interval called an epoch [53], [54]. In our work, an epoch length T^e is one hour, and a 24-epoch period represents a full day. Over the course of an epoch, the workload arrival rates can be reasonably approximated as constant, e.g., the Argonne National Lab Intrepid log shows mostly-constant arrival rates over large intervals of time [55].

We assume that the beginning of each epoch represents a steady-state scheduling problem where we assign execution rates, i.e., reciprocal of the execution time, of a set of I workload task-types to D data centers. A task-type $i \in I$ is characterized by its arrival rate AR_i, and the estimated

time required to complete a task of task-type i on each of the heterogeneous compute nodes in each P-state. The assignment problem at the geo-distributed level is to assign execution rates for each task-type i to each data center $d \in D$ such that total energy cost across all data centers is minimized, with the constraint that the execution rates of all task-types meet their arrival rates, i.e., all tasks complete without being dropped or unexecuted. To fulfill this constraint, for each epoch τ, we assign a maximum data center execution rate $ER_{d,i}^{DC}$ for each task-type i to each data center d such that the total execution rate for all task-types exceed (or equal) the corresponding arrival rate, AR_i, thus ensuring the workload can be completed. That is,

$$\sum_{d=1}^{D} ER_{i,d}^{DC}(\tau) \geq AR_i(\tau), \quad \forall i \in I. \tag{1}$$

2.3.3. DATA CENTER LEVEL

2.3.3.1. ORGANIZATION OF EACH DATA CENTER

Each data center d houses NN_d number of nodes that are arranged in hot aisle/cold aisle fashion (Figure 8), and a cooling system comprised of NCR_d number of computer room air conditioning (CRAC) units. Heterogeneity exists among compute nodes, where nodes vary in their execution speeds, power consumption characteristics, and number of cores. Cores within a compute node are homogeneous, and each core is DVFS-enabled to allow independent configuration of its P-states. The number of cores in node n is NCN_n, and NT_k is the node type to which core k belongs.

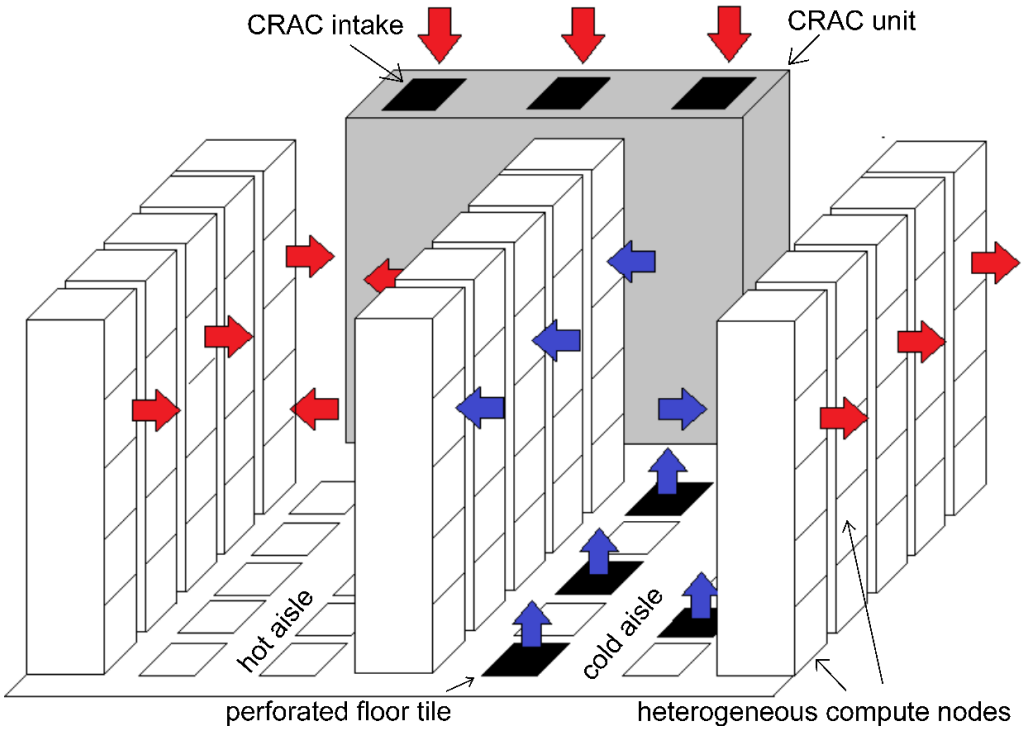


Figure 8: Data center in hot aisle/cold aisle configuration [56]

2.3.3.2. COMPUTE CORE EXECUTION RATES

Recall that our GDRM determines the distribution of tasks of each task-type among all data centers. At each data center d, the sum of execution rates of all cores that are assigned to execute task-type i must exceed or equal $ER_{d,i}^{DC}(\tau)$. We assume that we know the estimated computational speed (ECS) of any task of type i on a core of node-type n in P-state p, $ECS(i,n,p)$, determined using historical, experimental, or analytical techniques [57], [58].

For epoch τ, we assign a desired fraction $DF_{i,k}(\tau)$ of time each core k will spend executing tasks of type i and the P- state $PS_{i,k}(\tau)$ each core k is configured when executing tasks of type i. We assume tasks will run serially until completion. That is, a core sharing its time among multiple tasks implies that a scheduler will assign different tasks to execute on the core in such a manner that, over a long period of time (i.e., steady-state), the amount of time a core k spends executing a

task of type i would equal its assigned $DF_{i,k}(\tau)$ value. The core execution rate $ER_{i,k}^{core}$ of tasks of type i on core k is

$$ER_{i,k}^{core}(\tau) = DF_{i,k}(\tau) \cdot ECS(i, NT_k, PS_{i,k}(\tau)). \quad (2)$$

At the data center level, we assign $DF_{i,k}(\tau)$ and $PS_{i,k}(\tau)$ such that power is minimized (see Section 2.3.3.3), and the execution rates of all task-types on cores in all data centers equal the arrival rate ensuring that the arriving workload is fully executed. That is,

$$\sum_{d=1}^{D} \sum_{n=1}^{NN_d} \sum_{k=1}^{NCN_n} ER_{i,k}^{core}(\tau) = AR_i(\tau), \quad \forall i \in I, \forall d \in D. \quad (3)$$

2.3.3.3. COMPUTE NODE POWER MODEL

The power consumption of a compute node consists of the static overhead power consumption (equal to the amount of power consumed when the system is idle) and the additional dynamic power consumed when cores are executing tasks. We define O_n as the overhead power consumption of compute node n (a constant value independent of the workload resulting from system components such as storage, network interfaces). Let APC(i, NT_k, $PS_{i,k}(\tau)$) be the average power consumed by core k in a node of type NT_k when executing tasks of type i in P-state $PS_{i,k}(\tau)$ during epoch τ. The power consumption of node n during epoch τ, $PN_n(\tau)$, is

$$PN_n(\tau) = O_n + \sum_{k=1}^{NCN_n} \sum_{i=1}^{I} APC(i, NT_k, PS_{i,k}(\tau)) \cdot DF_{i,k}(\tau). \quad (4)$$

2.3.3.4. COOLING POWER MODEL

The heat generated by compute nodes is removed by the CRAC units. The airflow within the data center causes heat generated from nodes to propagate to other nearby nodes, thereby increasing the inflow temperature of those nodes. Using the notion of thermal influence indices [59] that were derived using computational fluid dynamics simulations, we can calculate the steady-state temperatures at compute nodes and CRAC units in each data center. Because we assume the same physical layout for each of the data centers (Figure 8), we use thermal influence indices derived for one data center layout based on an average workload that would be executed by the data center.

The outlet temperature of each compute node is a function of the inlet temperature, the power consumed, and the air flow rate of the node. The inlet temperature of each compute node is a function of the outlet temperatures of each CRAC unit and the outlet temperatures of all compute nodes of the same data center [56]. The ASHRAE guidelines have designated the inlets of IT equipment as the common measurement point for temperature compliance [60], and therefore we consider thermal constraints at the inlets of compute nodes. For all nodes, the inlet temperature of each node is constrained to be less than or equal to the red-line temperature (maximum allowable node temperature).

The power consumed by a CRAC unit is a function of the heat removed at that CRAC unit and the Coefficient of Performance (CoP) of the CRAC unit [61]. Let ρ_d be the density of air and C_d be the specific heat capacity of air at data center d. Let $TC_{d,c}^{in}(\tau)$, $TC_{d,c}^{out}(\tau)$, and $AFC_{d,c}(\tau)$ be the inlet temperature, outlet temperature, and air flow rate, respectively, of CRAC unit c in data center d during epoch τ. Then the power consumed by this CRAC unit, $PCR_{d,c}(\tau)$, can be calculated as [61]

$$PCR_{d,c}(\tau) = \frac{\rho_d \cdot C_d \cdot AFC_{d,c}(\tau) \cdot \left(TC_{d,c}^{in}(\tau) - TC_{d,c}^{out}(\tau)\right)}{CoP_d\left(TC_{d,c}^{out}(\tau)\right)}. \quad (5)$$

2.3.3.5. NODE ACTIVATION/DEACTIVATION POWER OVERHEAD

At each data center, the number of nodes of each node-type that are in-use frequently changes among epochs. Inactive nodes are placed in a sleep state, but entering and exiting this sleep state takes some time due to the actions required in both hardware and software to transition the system between states. Each node that is active is considered to be active for the entire epoch, which requires that any node transitioning to/from a sleep state do so during the epoch following/before the current epoch, respectively.

For each data center d, let $N_{d,j}^{trans}(\tau)$ be the number of nodes of type j that activate or deactivate during epoch τ. Let P_j^{Sleep} be the average sleep power for node-type j. Let P_j^D be the average peak dynamic power for node-type j. It is calculated by averaging over all task-types the peak power for each task-type i executing on node-type j. The average node utilization of node-type j defined as μ_j. Let CoP of the CRAC unit at data center d be CoP_d. Without loss of generality, we assume that each data center contains the same number of nodes, however each data center is heterogeneous in the sense that the number of nodes belonging to each node-type among data centers varies. Let J_d be the set of node-types in data center d. Let T^S be the time required for a node to transition to/from a sleep state. Recall that T^e is the duration of an epoch. The node activation/deactivation power ADP_d for data center d during epoch τ is then calculated as

$$ADP_d(\tau) = \sum_{j \in J_d} \left[\left(\mu_j P_j^D - P_j^{Sleep}\right) \cdot N_{d,j}^{trans}(\tau)\right] \cdot \left(1 + \frac{1}{CoP_d}\right) \cdot \frac{T^S}{T^e}. \quad (6)$$

2.3.3.6. RENEWABLE POWER MODEL

Each data center is equipped with and partially powered by a renewable energy source. Every location can have either solar power, wind power, or some combination of both. Solar power P_d^{solar} and wind power P_d^{wind} (both have units of kW) are calculated as [62] for each data center d as an average per epoch τ. The total renewable power, $PR_d(\tau)$, available at data center d during epoch τ is the sum of the wind and solar power available at that time. We use these models with historical data [63] to predict the renewable power available at each data center:

$$PR_d(\tau) = P_d^{solar}(\tau) + P_d^{wind}(\tau). \tag{7}$$

2.3.3.7. OVERALL DATA CENTER POWER MODEL

Let Eff_d be an approximation of the power overhead coefficient in data center d due to the inefficiencies of power supply units. Eff_d is always greater than or equal to 1. The total non-renewable power consumed throughout data center d during epoch τ, $PD_d(\tau)$, is calculated as

$$PD_d(\tau) = \left(\sum_{c=1}^{NCR_d} PCR_{d,c}(\tau) + \sum_{n=1}^{NN_d} PN_n(\tau) + ADP_d(\tau) \right) \cdot Eff_d - PR_d(\tau). \tag{8}$$

For epoch τ, $PD_d(\tau)$ can be negative if the renewable power available at the data center, $PR_d(\tau)$, is greater than the cooling and computing power.

2.3.3.8. NET METERING MODEL

Net metering allows data center operators to sell back the excess renewable power generated on-site to the utility company. When the excess power is added into the grid, utility companies pay a fraction of the retail price. This fraction is called the net metering factor, α.

2.3.3.9. PEAK DEMAND MODEL

Most utility providers charge a flat-rate (peak demand) fee based on the highest (peak) power consumed at any instant during a given billing period, e.g., month. The peak demand price per kW at data center d is denoted as P_d^{price}. We define $Pc_d^{peak}(\tau)$ as the highest grid power consumed since the beginning of the current month, including the current epoch τ. We define $Pp_d^{peak}(\tau)$ as the highest grid power consumption since the beginning of the current month until the start of the current epoch τ. The peak power increase at data center d during epoch τ, $\Delta_d^{peak}(\tau)$, is then defined as

$$\Delta_d^{peak}(\tau) = Pc_d^{peak}(\tau) - Pp_d^{peak}(\tau) \tag{9}$$

if $Pc_d^{peak}(\tau) \geq Pp_d^{peak}(\tau)$ else it is equal to 0. The peak power increase $\Delta_d^{peak}(\tau)$ is calculated in each epoch τ and summed over all epochs in a billing period to calculate total peak power.

2.3.3.10. SYSTEM ELECTRICITY COST

The electricity price per kWh at data center d during epoch τ is defined as $E_d^{price}(\tau)$. Data center operators can use net metering if total non-renewable power consumed throughout the data center, $PD_d(\tau)$, is negative. For such conditions, the total power/electricity cost $PC_d(\tau)$ for data center d during epoch τ can be defined as

$$PC_d(\tau) = E_d^{price}(\tau) \cdot PD_d(\tau) + P_d^{price} \cdot \Delta_d^{peak}(\tau) \tag{10}$$

if $PD_d(\tau)$ is positive. If $PD_d(\tau)$ is negative then

$$PC_d(\tau) = E_d^{price}(\tau) \cdot \alpha \cdot PD_d(\tau) + P_d^{price} \cdot \Delta_d^{peak}(\tau) \quad (10)$$

where α=0 if net metering is not available. The first term in Equation (10) represents the TOU electricity cost and the second term represents the peak demand cost.

2.3.3.11. CO-LOCATION INTERFERENCE MODEL

Tasks competing for shared memory in multicore processors can cause severe performance degradation, especially when competing tasks are memory-intensive [64]. The memory-intensity of a task refers to the ratio of last-level cache misses to the total number of instructions executed. We employ a linear regression model from [65] that combines a set of disparate features (i.e., inputs that are correlated with task execution time) based on the current tasks assigned to a multicore processor to predict the execution time of a target task i on core k in the presence of performance degradation due to interference from task co-location. These features are, the number of applications co-located on that multicore processor, the base execution time, the clock frequency, the average memory intensity of all applications on that multicore processor, and the memory intensity of application i on core k.

In our linear model, the output is a linear combination of all features and their calculated coefficients. We classify the task-types into memory-intensity classes on each of the node-types, and calculate the coefficients for each memory-intensity class using the linear regression model to determine a co-located execution rate for task-type i on core k, $CER_{i,k}^{core}(\tau)$. The total execution rate for task-type i in epoch τ is therefore given by

$$CER_i(\tau) = \sum_{d=1}^{D} \sum_{k=1}^{NC_d} CER_{i,k}^{core}(\tau). \quad (11)$$

Because co-location interference degrades the execution rate of task-types, some tasks may be unable to finish if task-types are allocated to cores based on Equation (2) that does not consider degradation effects. To allocate tasks to cores when considering co-location interference, some of our techniques use information about $CER_{i,k}^{core}$ to judge actual execution rates more accurately than techniques that do not consider co-location interference. When considering co-location at a data center d, the data center execution rate constraint becomes

$$\sum_{k=1}^{NC_d} CER_{i,k}^{core}(\tau) \geq ER_{d,i}^{DC}(\tau), \quad \forall i \in I, \forall d \in D. \tag{12}$$

The linear regression model was trained using execution time data that was collected by executing benchmarks from the PARSEC [66] and NAS parallel [67] benchmark suites on a set of server class multicore processors that define the nodes used in our study (see Table 1 in Section 2.6 for details about each node). This model for execution time prediction under co-location interference is derived from real workloads and machines, and results in a mean prediction error of approximately 7% [65].

2.4. PROBLEM FORMULATION

We consider a scenario with multiple data centers sharing a single workload. The system is assumed to be under-subscribed in the sense that the system is expected to have enough computation resources to complete the workload without requiring that any tasks be dropped. Though the system is under-subscribed, individual data centers may be executing at full capacity. The tasks originate off-site from the data centers, and we make the simplifying assumptions that the transmission time and cost from a task origin to a data center is equivalent for all data centers. The objective of a GDRM is to minimize monetary electricity cost of the geo-distributed system

(the sum of Equation (10) across all data centers) while ensuring that the workload is completed according to the constraints defined by Equations (1) and (12).

The problem is especially challenging when considering the variable amount of renewable power available at each data center, the heterogeneity of compute nodes within a data center, and the additional constraint that the entire workload must complete without dropping any tasks. Having information about TOU electricity pricing, peak demand pricing, a prediction of the amount of renewable power, net metering policy at each data center, the incoming workload, and the execution speeds of task-types on the heterogeneous compute nodes allows our GDRM to make intelligent decisions for allocating the workload, as described next in Section 2.5.

2.5. HEURISTICS DESCRIPTIONS

2.5.1. OVERVIEW

The GDRM allocates the incoming workload not only to individual data centers, but also to specific nodes within each data center. The GLD problem is NP-hard [42], and therefore we propose three resource management heuristics for GDRM, with each having different levels of detail of the system model available to it.

2.5.2. FORCE DIRECTED LOAD DISTRIBUTION HEURISTICS

Force-directed load distribution (FDLD) is a variation of force-directed scheduling [68], a technique often used for optimizing semiconductor logic synthesis. FDLD is an iterative heuristic that selectively performs operations to minimize system forces until all constraints are met. We adapt the FDLD approach proposed in [42] to the rate-based allocation environment we have

outlined in Section 2.3, and enhance it to propose two new FDLD based heuristics to solve our problem.

ALGORITHM 1: PSEUDO-CODE FOR FDLD HEURISTICS

```
1.  allocate an instance of each task-type to every node in every data center in every epoch
2.  operations-remaining = true
3.  while operations-remaining
4.     for each node with tasks still allocated to it
5.        for each task-type on the node
6.           temporarily remove task-type from node
7.           if FDLD-CL
8.              estimate execution rates using Equation (11) (CER_i)
9.           else if FDLD-TAO
10.             estimate execution rates using Equation (15) and φ_i^C
11.          else if FDLD-SO
12.             calculate execution rates using Equation (15) and φ^C
13.          calculate estimated power costs PC_d^E(τ)
14.          calculate F^S from F^ER and F^C
15.          if execution rate constraints are not violated
                   (Equation (12) for FDLD-CL, Equation (16) for
                   FDLD-SO and FDLD-TAO)
16.             add to set of possible task removal operations
17.          restore task-type to node
18.       end for
19.    end for
20.    if set of possible task removal operations is empty
21.       operations-remaining = false
22.    else
23.       choose and implement the task-type removal
                operation that would result in the lowest F^S
24. end while
25. return final allocation solution
```

Our baseline FDLD heuristic is the one proposed in [42], which we enhance with simple over-provisioning (FDLD-SO) to compensate for performance degradation due to co-location. This allows the FDLD heuristic to meet the execution rate constraint at a given data center. This heuristic over-provisions all task-types equally by scaling estimated task execution rates by a factor ϕ^C. Our first new heuristic improves upon FDLD-SO by using task aware over-provisioning

(FDLD-TAO) to estimate co-location effects for each task-type by a factor specific to each task-type i, ϕ_i^C. For both FDLD-SO and FDLD-TAO, the degree of over-provisioning (ϕ^C and ϕ_i^C, respectively) is determined empirically through simulation studies to provide values that give the system the best possible performance. Lastly, our second new heuristic uses the co-location (FDLD-CL) models given in Section 2.3.3.11 to account for co-location effects when calculating task execution rates.

The fundamental operation of all FDLD variants is described in Algorithm 1. To generate the initial solution, every node in every data center in every epoch is assigned to execute all task-types (step 1). Each iteration of the FDLD removes one instance of one task-type from a single node, selecting the task to remove, resulting in the lowest total system force F^S (steps 4-23). After getting the final allocation solution from the heuristic, we calculate the final execution rates and the system electricity cost by summing the power costs across all data centers. The rest of this section presents the derivation of the total system force F^S, Equation (15) (used in steps 10 and 12 in Algorithm 1), and Equation (16) (used in step 15 in Algorithm 1).

As per Equation (2), the task execution rate is a function of the P-state of the node the task is executing at, but FDLD is not designed to make DVFS decisions to set the execution rates of task-types, and therefore we assume that it is going to make the decisions based on the node utilization. An average execution rate must be determined for all task-types using the average node utilization factor μ_j for each node-type j. Let $ER_{j,i}(P_{MAX})$ and $ER_{j,i}(P_0)$ be the execution rates of task-type i running on a single core of a node of type j in the highest numbered P-state and lowest numbered P-state, respectively. Therefore, the equivalent single core execution rate $R_{j,i}$ of task-type i on node-type j is

$$R_{j,i} = ER_{j,i}(P_{MAX}) + \left[ER_{j,i}(P_0) - ER_{j,i}(P_{MAX})\right]\mu_j. \quad (13)$$

The single core execution rate of the application is equal to the minimum execution rate executing at the highest numbered P-state (first term in Equation (13)) plus a performance factor (second term in Equation (13)). The node utilization will affect the single core execution rate such that the execution rate will be proportional to the node utilization, e.g., $R_{j,i}$ will be $ER_{j,i}(P_0)$ for $\mu_j = 1$ and $ER_{j,i}(P_{MAX})$ for $\mu_j = 0$.

Let $NN_{d,j}$ be the number of nodes of type j in data center d. Let $S_{d,j,n}(\tau)$ be the set of instances of different task-types placed on a node n of node-type j in data center d during epoch τ. An instance of a task-type is a task that belongs to the specific task-type. Let $Q_{d,j,i}(\tau)$ be the equivalent number of nodes of type j running task-type i in data center d during epoch τ. We assume that the compute time of each core on a node is evenly divided among its assigned tasks. Therefore the equivalent number of nodes $Q_{d,j,i}(\tau)$ represent the compute time allocated to task-type i on node-type j for data center d, given by

$$Q_{d,j,i}(\tau) = \sum_{n=1}^{NN_{d,j}} \begin{cases} \frac{1}{\left|S_{d,j,n(\tau)}\right|} & if\ i \in S_{d,j,n}(\tau) \\ 0 & otherwise \end{cases}. \quad (14)$$

For example, for a data center with twelve nodes and three task-types assigned, each task-type would get the equivalent of four nodes out of twelve, i.e., 13 of the available compute resources. This will be assigned such that each task-type gets 13 of the execution time of each of the twelve nodes.

To compensate for performance degradation due to co-location effects, node over-provisioning is accomplished by the factor ϕ. Let K_j be the number of cores in a node of type j.

The average estimated execution rate $ER_{j,i}^{E}(\tau)$ of task-type i on node-type j during epoch τ, when using either the FDLD-SO or FDLD-TAO versions, is given by

$$ER_{j,i}^{E}(\tau) = \sum_{d=1}^{D} \sum_{j \in J_d} K_j \cdot R_{j,i} \cdot \phi \cdot Q_{d,j,i}(\tau) \tag{15}$$

subject to the constraint

$$ER_{j,i}^{E}(\tau) \geq AR_i(\tau) \quad \forall i \in I, \tag{16}$$

where ϕ is replaced by either ϕ^C or ϕ_i^C in Equation (15) when using either FDLD-SO or FDLD-TAO, respectively.

Let Z be the term that is replaced by $CER_i(\tau)$ when considering the FDLD-CL heuristic and is replaced by $ER_{j,i}^{E}(\tau)$ when using either FDLD-SO or FDLD-TAO. The execution rate force F^{ER} is calculated using

$$F^{ER}(\tau) = \sum_{i \in I} \left[e^{\left(\frac{Z}{AR_i(\tau)} - 1\right)} - 1 \right]. \tag{17}$$

Observe that $F^{ER}(\tau)$ will decrease to zero as the ratio of Z to $AR_i(\tau)$ decreases to 1.

Let $P_{d,j,n}^{E}$ be the estimated average power for node n of node-type j in data center d, calculated as

$$P_{d,j,n}^{E} = \begin{cases} P_j^{Sleep} & if \left| S_{d,j,n}(\tau) \right| = 0 \\ \mu_j P_j^D + P_j^S & otherwise \end{cases}. \tag{18}$$

For all FDLD variants, let $PD_d^E(\tau)$ be the estimated non-renewable power consumed at data center d during epoch τ, calculated as

$$PD_d^E(\tau) = \left[\sum_{j \in J_d} \sum_{n=1}^{NN_{d,j}} P_{d,j,n}^E \cdot \left(1 + \frac{1}{CoP_d}\right) + ADP_d(\tau)\right] \cdot Eff_d - PR_d(\tau), \tag{19}$$

For all FDLD variants, let $PC_d^E(\tau)$ be the estimated power cost at data center d during epoch τ. $PC_d^E(\tau)$ is calculated as shown in Equation (10), where $PD_d(\tau)$ is replaced by its estimated version, i.e., $PD_d^E(\tau)$.

Let $PC_d^{max}(\tau)$ be the maximum real power cost possible at data center d, calculated using

$$PC_d^{max}(\tau) = E_d^{price}(\tau) \cdot \sum_{j \in J_d} NN_{d,j} \cdot \left(\mu_j P_j^D + P_j^S\right) + P_d^{price} \cdot \Delta_d^{peak}(\tau). \tag{20}$$

The cost force F^C can then be calculated with

$$F^C(\tau) = \sum_{d=1}^{D} \left[e^{\left(\frac{PC_d^E(\tau)}{PC_d^{max}(\tau)}\right)} - 1\right]. \tag{21}$$

Observe that the value of F^C goes to zero as the ratio of $PC_d^E(\tau)$ to $PC_d^{max}(\tau)$ decreases to zero.

Let N^τ be the total number of epochs being considered. The total system force across all epochs, F^S, is calculated as

$$F^S = \sum_{\tau=1}^{N^\tau} F^{ER}(\tau) + F^C(\tau). \tag{22}$$

2.5.3. GENETIC ALGORITHM HEURISTIC

We designed a third heuristic: a genetic algorithm load distribution with co-location awareness (GALD-CL). The Genitor style [69] GALD-CL heuristic has two parts: a genetic algorithm based GDRM (Algorithm 2) and a local data center level greedy heuristic (Algorithm 3) that is used to calculate the fitness value of the genetic algorithm.

ALGORITHM 2: PSEUDO-CODE FOR GALD-CL HEURISTIC

1. create an initial population of chromosomes
2. **while** within time limit
3. perform selection, crossover and mutation to create new chromosomes
4. **for** each new chromosome
5. use greedy heuristic to perform allocations at data center level
6. **end for**
7. find the fitness values (total energy costs) of the population
8. trim population to its original size by eliminating the least-fit chromosomes
9. **end while**
10. return best chromosome from population and use final allocation from it

The initial population for GALD-CL (Algorithm 2) is generated by randomly partitioning the global arrival rate AR_i for each task-type i in the epoch across all of the data centers (step 1). Each data center d gets a desired fraction of arrival rate $DAR_{d,i}$ for each task-type i such that,

$$\sum_{d=1}^{D} DAR_{d,i}(\tau) = AR_i(\tau), \quad \forall i \in I. \tag{23}$$

Each chromosome is a matrix of I x D genes, where each gene is a $DAR_{d,i}$ and d pair, representing the desired arrival rate of each task-type i at each data center d. We perform selection, crossover, and mutation that alter existing chromosomes to generate new offspring chromosomes (step 3). We use a roulette wheel selector, a two-point crossover, and a simple real range mutator [70]. After the generation of two offspring, for each new chromosome, a local greedy heuristic discussed in the next paragraph is used to perform allocations at the data center level (steps 4-6).

After the greedy heuristic, the fitness value (total energy cost including CRAC energy) for each chromosome is calculated (step 7). The population is trimmed to its original size by eliminating the least-fit (highest cost) chromosomes (step 8). After reaching the time limit, the algorithm ends and the final allocation of arrival rates is obtained from the best (lowest cost) chromosome.

Local Greedy Heuristic: This greedy approach (Algorithm 3) is similar in concept to Min-min in [71], [72] and is used to assign the desired arrival rate $DAR_{d,i}$ of each task-type i to cores in data center d. For this heuristic, cores are dedicated to a single particular task-type, i.e., a core cannot execute more than one task-type at a time. Our greedy heuristic iteratively assigns task-types to cores to find the most efficient mapping, where we define efficiency $EFF_{i,k}$ for a task mapping of type i on a node of type NT_k in P-state $PS_{i,k}(\tau)$ as

$$EFF_{i,k}(\tau) = \frac{ECS(i, NT_k, PS_{i,k}(\tau))}{APC(i, NT_k, PS_{i,k}(\tau))}. \quad (24)$$

For each data center, we start this greedy heuristic by finding the task-type/node-type pair with the highest $EFF_{i,k}(\tau)$ value (Equation (24)) among all possible pairs (step 2). Our heuristic uses the most efficient P-state for a given task-type/node-type pair. All I x J_d task-type to node-type pairs are then sorted by their efficiency in descending order to create a list (step 3). At each iteration of the heuristic, the first (most efficient) pair is selected (step 5). Then the heuristic checks if any unassigned cores within the selected node-type exist (step 6). If so, we assign fraction of the $DAR_{d,i}$ for the selected task-type to a core in the selected node-type (based on $ECS(i, n, p)$) (step 7) and that core is removed from consideration (step 8). The CRAC outlet temperatures are set to the red-line temperature, and then the outlet temperatures of all CRAC units are iteratively decreased by one degree until the thermal constraint (the inlet temperature of the node, with the

selected core, should be less than the red-line temperature) is met (step 9). After the assignment, the execution rate $ER_{d,i}^{DC}$ is updated (step 10). If the $ER_{d,i}^{DC}$ for the selected task-type exceeds its $DAR_{d,i}$ (step 11), all task-type/node-type pairs with that task-type are removed from the list (step 12). The heuristic uses Equation (11) to estimate core execution rates with constraint defined by Equation (12). If no unassigned cores within the selected node-type exist (step 13), that task-type/node-type pair is removed from consideration (list) (step 14). This iterative part (steps 4-15) of the heuristic stops when there are no more task-type/node-type pairs to consider.

ALGORITHM 3: PSEUDO-CODE FOR LOCAL GREEDY HEURISTIC

1. **for** each data center
2. find most efficient task-type/node-type pair among all possible pairs
3. sort all task-type/node-type pairs by efficiency in descending order
4. **while** sorted list is not empty
5. select first task-type/node-type pair
6. **if** unassigned core within selected node-type exist
7. assign fraction of $DAR_{d,i}$ for the selected task-type to a core from the selected node-type
8. remove that core from future consideration (assigned)
9. set CRAC outlet temperatures to temperatures such that node inlet temperature is less than red-line temperature
10. update $ER_{d,i}^{DC}$
11. **if** $ER_{d,i}^{DC}$ for the selected task-type exceeds its $DAR_{d,i}$
12. remove all task-type/node-type pairs with that task-type
13. **else** //no more cores from node-type exist
14. remove that task-type/node-type pair from the list
15. **end while**
16. **end for**
17. **for** each data center
18. **for** each task-type
19. **if** $DAR_{d,i}$ exceeds the data center assignment $ER_{d,i}^{DC}$ //solution invalid
20. adjust $DAR_{d,i}$ value by reducing it by 10% for data center where solution is invalid
21. normalize $DAR_{d,i}$ values to modify chromosome until valid solution reached, return to step 1
22. **end for**
23. **end for**

After mapping the desired arrival rates $DAR_{d,i}$ of each task-type i to cores in each data center d, if the $DAR_{d,i}$ exceeds the data center assignment $ER_{d,i}^{DC}$ (checked for each task-type in each data center), the solution is invalid (step 19). If so, we adjust $DAR_{d,i}$ value by reducing it by 10% for the data center at which the solution is invalid (step 20). To calculate normalized arrival rate value $DAR_{d,i}^{norm}$, we then normalize the $DAR_{d,i}$ values such that the sum of $DAR_{d,i}^{norm}$ values for each task-type matches the global arrival rate AR_i

$$DAR_{d,i}^{norm}(\tau) = DAR_{d,i}(\tau) \cdot \frac{AR_i(\tau)}{\sum_{d=1}^{D} DAR_{d,i}(\tau)}. \quad (25)$$

We then replace $DAR_{d,i}$ with $DAR_{d,i}^{norm}$ for all data centers d with the given task-type i to modify the chromosome until a valid solution can be reached (step 21), i.e., $ER_{d,i}^{DC}$ exceeds (or equals) $DAR_{d,i}$.

In summary, the GALD-CL heuristic addresses two potential shortcomings of the FDLD variants. First, the nature of the decision making within the FDLD variants prevents them from making any kind of DVFS decisions, therefore a single P-state is chosen for each node-type regardless of the tasks executing on the node. The greedy heuristic within the GALD-CL approach chooses the most efficient P-state for each task-type on each node-type [56]. Second, the FDLD variants are susceptible to becoming trapped in local minima. The genetic algorithm portion of the GALD-CL approach intrinsically enables escape from local minima, allowing a more complete search of the solution space.

2.6. SIMULATION ENVIRONMENT

Experiments were conducted for three geo-distributed data center configurations containing four, eight, and sixteen data centers. Locations of the data centers in the three configurations were selected from major cities around the continental United States to provide a variety of wind and solar conditions among sites and at different times of the day (see Figure 9).

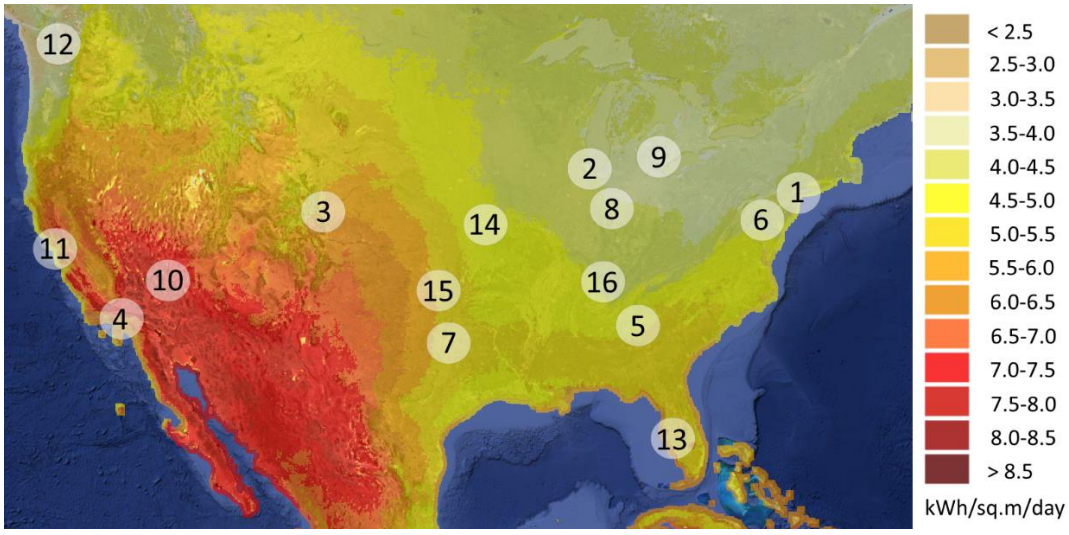


Figure 9: Location of simulated data centers overlaid on solar irradiance intensity map (average annual direct normal irradiance); wind data collected but not shown [63]

Experiments for the configuration with four data centers used locations one through four from Figure 9, while experiments using configuration with eight and sixteen data centers used locations one through eight and one through sixteen, respectively. The sites of each configuration were selected so that each configuration would have a fairly even east coast to west coast distribution to better exploit TOU pricing, peak demand pricing, net metering, and renewable power. Each data center consists of 4,320 nodes arranged in four aisles, and is heterogeneous within itself, having nodes from either two or three of the node-types given in Table 1, with most locations having three node-types and per-node core counts that range from 4-12 cores. For CRAC

units, the red-line temperature was set to 30C, which is on the high end of ASHRAE's temperature guidelines [60].

Table 1: Node Processor Types Used in Experiments

Intel processor	# cores	L3 cache	frequency range
Xeon E3-1225v3	4	8MB	0.8 - 3.20 GHz
Xeon E5649	6	12MB	1.60 - 2.53 GHz
Xeon E5-2697v2	12	30MB	1.20 - 2.70 GHz

Nodes placed in a sleep state by a heuristic are considered to be in the Advanced Configuration and Power Interface (ACPI) node sleep state S3, where RAM remains powered on. Sleep state S3, also commonly referred to as suspend or standby, allows greatly reduced power consumption while still possessing a small latency to return to an active operating state. Sleep power for all nodes is calculated as a fixed percentage of static power for each node-type, assumed to be 16% based on a study of node power states [73]. The average node utilization factor used during FDLD allocations, μ_j, is set as 0.75. The Coefficient of Performance (CoP) of the CRAC unit was determined empirically by simulating workloads with different memory intensity classes at each data center location, and its value ranges between 1.43 and 2.08 for different configurations. The time of each epoch τ was set to be one hour. The time required to transition a node to or from a sleep state, T^S, was conservatively assumed to be five minutes.

The electricity prices used during experiments, as shown in Figure 7, were taken directly from Pacific Gas and Electric (PG&E) Schedule E-19, which is for commercial locations consuming between 500kW and 1MW [31]. The peak demand prices per kW are given in Table 2 are.

Table 2: Peak Demand Prices Used in Experiments

data center location	peak demand price ($/*kW*)	data center location	peak demand price ($/*kW*)
New York	11.04	Detroit	14.54
Chicago	3.82	Las Vegas	8.25
Denver	6.75	San Francisco	13.01
LA	8.91	Seattle	3.29
Atlanta	8.11	Tampa	10.25
Baltimore	3.84	Kansas City	6.39
Dallas	11.88	Oklahoma City	6.20
Indianapolis	10.57	Nashville	5.09

We assume that each data center has peak renewable power generating capacity equivalent to its maximum power consumption [32], [33], [74]. Renewable power at each location was either wind power, solar power, or a combination of the two. For example, a location with high average solar irradiance but low average wind speed would be restricted to having solar power only. Solar and wind data was obtained from the National Solar Radiation Database [63]. An example of the renewable power available at different locations is given in Figure 10.

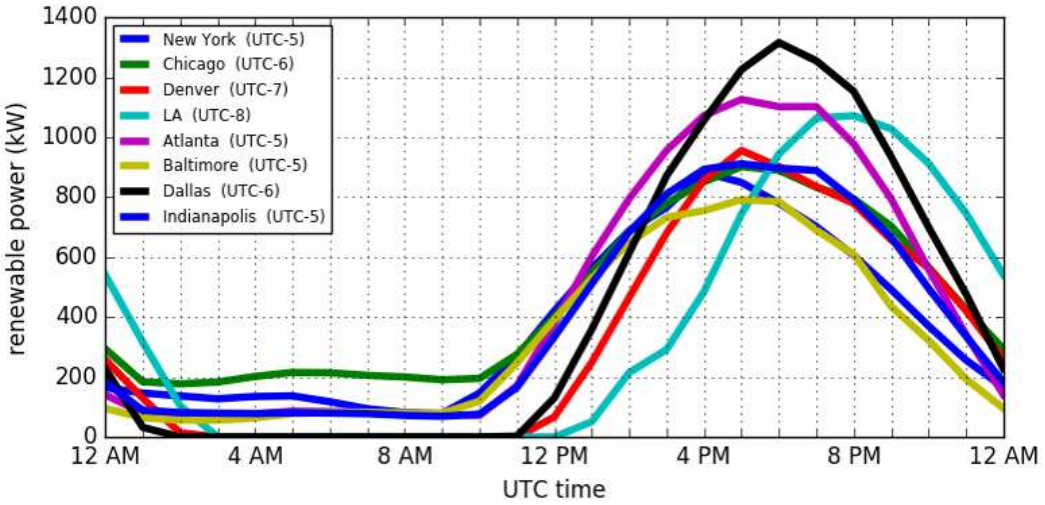


Figure 10: Renewable power available at eight locations

In net metering, data centers send excess power back to the grid and utility companies pay back the customer a fraction of retail price α. In most cases, the net metering factor α is 1; in very few cases, it is less than 1; and in some cases, it is 0, i.e., net metering is not available at that location [37], [75].

Each task-type used in our experiment is representative of a different benchmark from the PARSEC [66] and NAS parallel [67] benchmark suites. Task execution times and co-located performance data for the task of the different memory intensity classes were obtained from running the benchmark applications on the nodes listed in Table 1 [65]. Synthetic task arrival patterns were constructed and the baseline pattern is shown in Figure 11. For reference, the figure also shows TOU prices for an east coast and a west coast site.

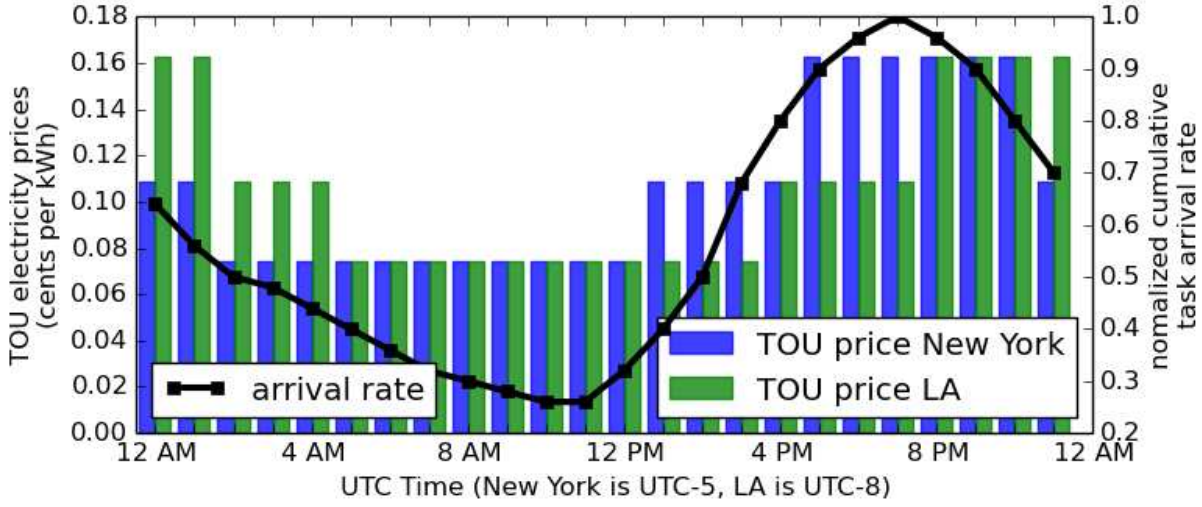


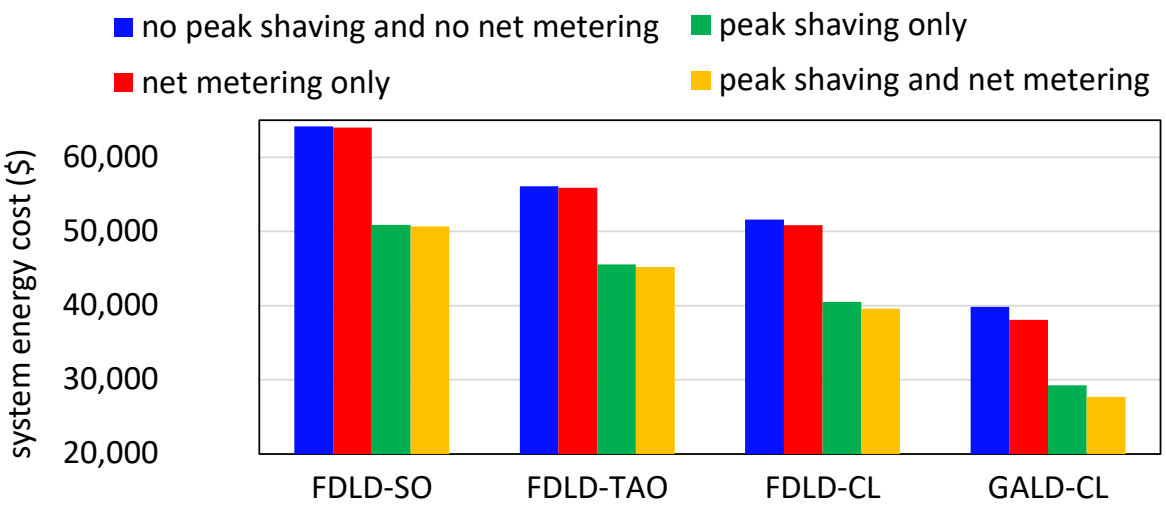
Figure 11: Baseline task arrival rate and TOU prices at two sites (New York, Los Angeles) over 24 hours

2.7. EXPERIMENTS

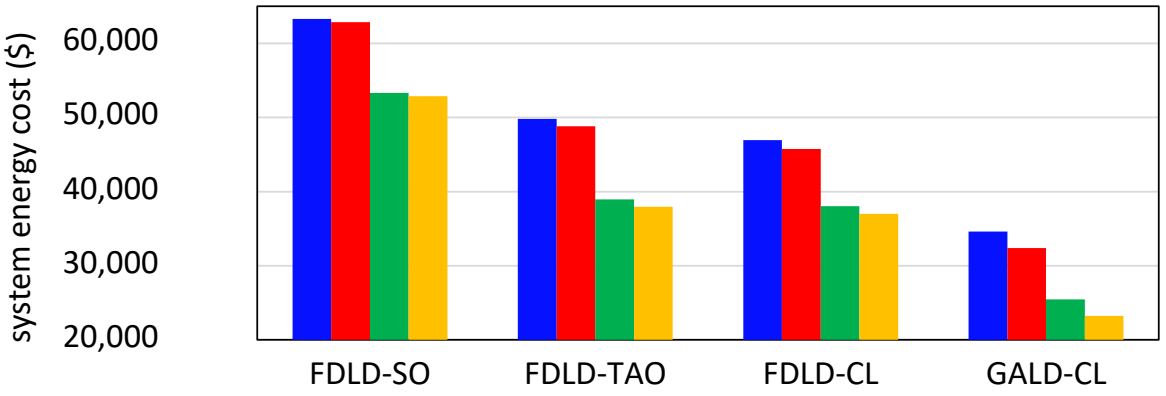
2.7.1. COST COMPARISON OF HEURISTICS

Our first set of experiments analyzes the total system energy cost for each heuristic described in Section 2.5 with and without peak shaving and net metering. For each heuristic, we evaluate four variants: (i) without both peak shaving and net metering, (ii) with net metering only, (iii) with peak shaving only, and (iv) with both peak shaving and net metering. Heuristic variants that are referred to as "without peak shaving" do not include the peak demand pricing factor in their objective functions but consider it while calculating the total monthly electricity cost at the end of the billing period. These experiments use a data center configuration consisting of eight locations and a workload that was a hybrid mix of memory-intensive and CPU-intensive task-types (discussed in Section 2.7.2). The system energy costs are estimated over a duration of one day. The results are shown in Figure 12(a).

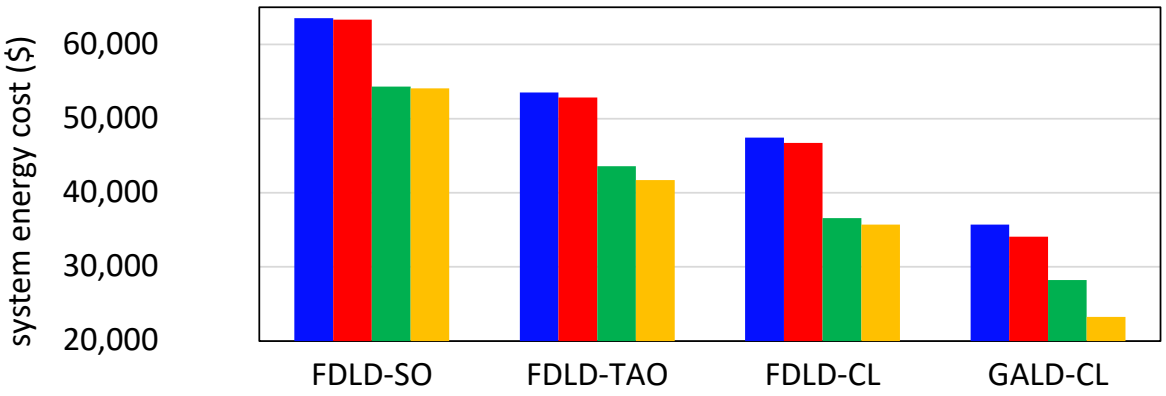
For each individual heuristic, considering both net metering and peak shaving produced the best results, while experiments without these produced worse results. This validates our consideration of peak shaving and net metering during geo-distributed data center workload management, to more effectively minimize energy costs. It can also be observed that the FDLD-CL heuristic, using the co-location models, performs the best among the FDLD variants. The FDLD-SO heuristic performed the worst, severely over provisioning nodes and resulting in high operating costs. The GALD-CL heuristic with net metering and peak shaving outperformed all other approaches. This heuristic has complete information about the entire system model, including the co-location models and task-node power (DVFS) models, allowing it to make better placement decisions. With additional execution time and a larger population of chromosomes even better solutions can be found.



(a) hybrid workload



(b) CPU-intensive workload



(c) memory-intensive workload

Figure 12: System energy costs for each heuristic over a day for (a) hybrid, (b) CPU-intensive, and (c) memory-intensive workloads, for eight data center locations

The decision option related to DVFS P-states gives the GALD-CL heuristic a strong advantage over the FDLD variants but it also takes longer to execute. The GALD-CL heuristic was limited to a run time of approximately one hour per epoch simulated. Alternatively, the FDLD heuristics completed in approximately six minutes per epoch simulated. While not performing as well as the GALD-CL, the FDLD variants have the advantage of reaching a solution more quickly, which may be beneficial in some cases.

2.7.2. WORKLOAD TYPE ANALYSIS

The previous experiment used a workload that was a hybrid mix of memory-intensive and CPU-intensive task-types. Figure 12 shows experiments for all of the FDLD and GALD-CL heuristics for a group of eight data centers where two additional workload types were evaluated: one where all of the tasks are highly memory-intensive (using data from *canneal*, *cg*, *ua*, *sp*, and *lu* benchmarks), and one where the tasks are highly CPU-intensive (using data from *fluidanimate*, *blackscholes*, *bodytrack*, *ep*, and *swaptions* benchmarks) [66], [67]. The composition of data center workloads can vary greatly and can impact the resource requirements, and these experiments show that the techniques presented in our research will perform well for a variety of workload types. Observe that the CPU-intensive workload typically costs the least as it experiences the least co-location degradation and therefore requires fewer nodes. However, the memory-intensive tasks cause performance degradation because they compete for shared memory in multicore processors.

2.7.3. SCALABILITY ANALYSIS

2.7.3.1. DATA CENTER SCALABILITY ANALYSIS

In this experiment, we analyze heuristic performance for additional problem sizes. Simulations running hybrid workloads were conducted for four and sixteen data center

configurations in addition to the previously discussed eight data center configuration. For each configuration, the average performance improvement of each heuristic over the FDLD-SO heuristic with no peak shaving and no net metering is given in Table 3. The GALD-CL heuristic was limited to a run time of approximately one hour. FDLD heuristics for four, eight, and sixteen locations completed on average in two, six, and eighteen minutes per epoch simulated, respectively. These experiments confirm that all FDLD heuristics can perform well for smaller and larger problem sizes but the GALD-CL heuristic consistently performs the best for all problem sizes.

Table 3: Energy Cost Reduction Comparison

	heuristic	**no PS and no NM**	**NM only**	**PS only**	**PS and NM**
4 data centers	FDLD-SO	0.0%	0.2%	23.4%	23.6%
	FDLD-TAO	12.3%	14.5%	33.4%	34.0%
	FDLD-CL	15.0%	15.7%	36.7%	37.9%
	GALD-CL	49.5%	58.2%	67.3%	75.8%
8 data centers	FDLD-SO	0.0%	0.3%	20.8%	21.1%
	FDLD-TAO	16.9%	16.8%	31.5%	32.5%
	FDLD-CL	19.6%	20.8%	36.9%	38.3%
	GALD-CL	40.9%	45.7%	54.5%	60.8%
16 data centers	FDLD-SO	0.0%	0.4%	21.7%	24.5%
	FDLD-TAO	11.1%	12.2%	30.9%	32.9%
	FDLD-CL	14.2%	16.1%	33.4%	36.6%
	GALD-CL	33.1%	36.4%	44.0%	46.6%

PS = peak shaving, NM = net metering

2.7.3.2. GA RUN TIME SCALABILITY ANALYSIS

Table 3 shows similar energy cost reduction results for all FDLD variants in the cases of the data center configurations containing four, eight, and sixteen data centers running hybrid workloads. But for GALD-CL, we notice that the energy cost reduction decreases with the increasing number of data centers. Here, as the number of data centers in the group grows larger, the problem size increases and the number of GALD-CL generations that can take place within the time limit (one hour by default) decreases, which decreases the performance of GALD-CL.

To better understand how the GALD-CL solution quality is impacted by the heuristic's run time, we increase the GALD-CL run time in proportion to the increase in number of data centers. We execute GALD-CL for about one hour for four data centers, about two hours for eight data centers, and about four hours for sixteen data centers. The results from Table 4 show that GALD-CL is capable of performing well for larger problem size, when given more time. For comparison, the Table 4 also includes results for GALD-CL executed for about one hour for a group of four, eight, and sixteen data centers. It should be noted that, even when allowing the GALD-CL to execute for one hour, it still provides the system a significant energy cost reduction in comparison to the all FDLD heuristics as shown in Table 3.

Table 4: Impact of GALD-CL Run Time

Configuration	GA epoch	no PS, no NM	NM only	PS only	PS and NM
4 data centers	1 hour	49.5%	58.2%	67.3%	75.8%
8 data centers	1 hour	40.9%	45.7%	54.5%	60.8%
	2 hours	46.7%	55.2%	62.5%	72.2%
16 data centers	1 hour	33.1%	36.4%	44.0%	46.6%
	4 hours	40.1%	49.4%	61.6%	69.9%

2.7.3.3. EPOCH-BASED ANALYSIS

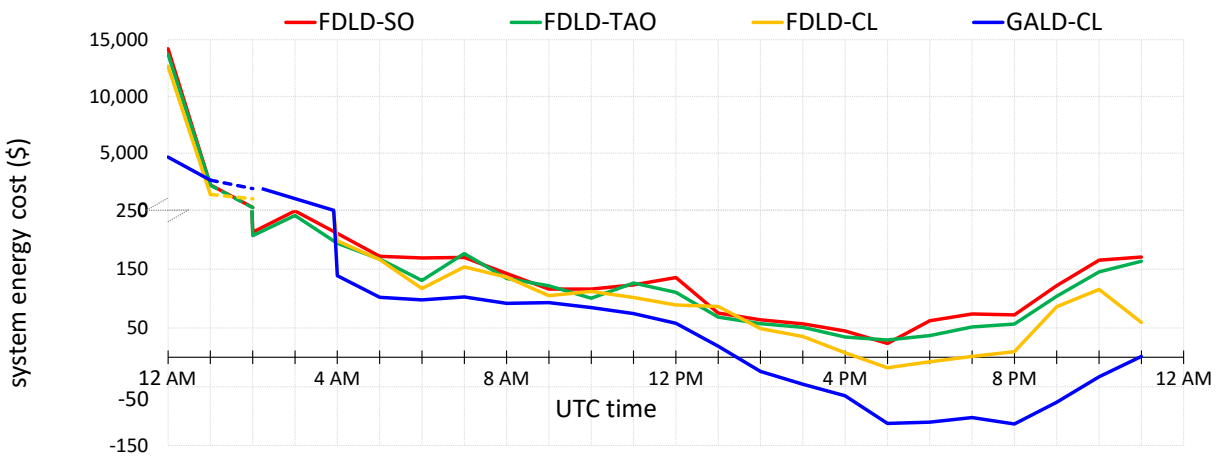
For most of our experiments, we analyzed the total system cost for each heuristic over one day. Figure 13 shows a more detailed view of the system operating cost at one-hour intervals over the course of a day for four, eight, and sixteen data centers executing a hybrid workload. The four resource management heuristics in this study consider both peak shaving and net metering.

Net metering causes the plots to go into the negative region in certain epochs, which represents the case when the system earns money by selling excess renewable power back to the utility companies. The operating cost for each heuristic is very high during the first epoch because the period for which the results are shown represents the first day of the month where the initial peak demand cost is added. This effect would not be present for other days of the month. After a few epochs, the performance of the FDLD-CL came close to the GALD-CL, but was not able to surpass its performance.

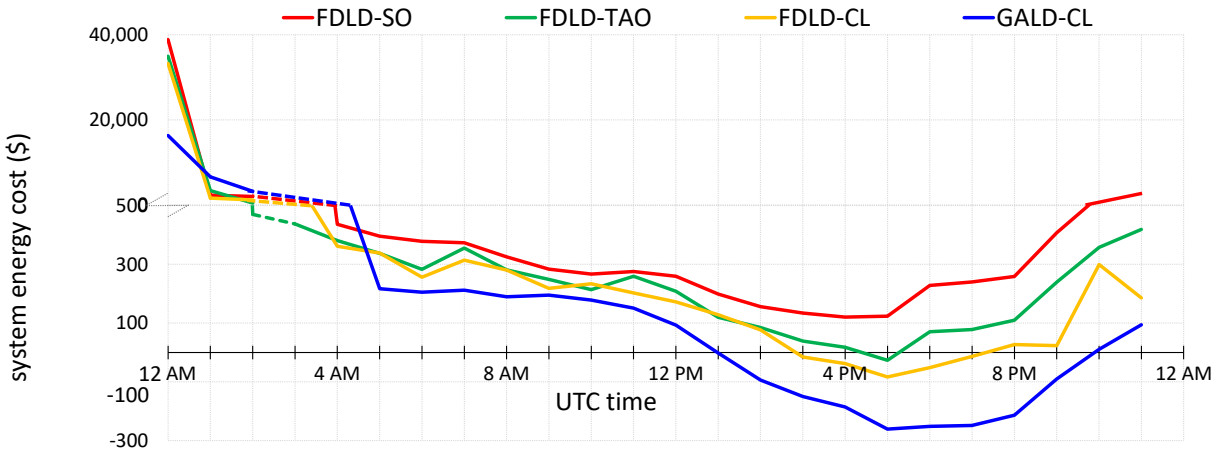
2.7.4. SENSITIVITY ANALYSIS

2.7.4.1. NET METERING FACTOR

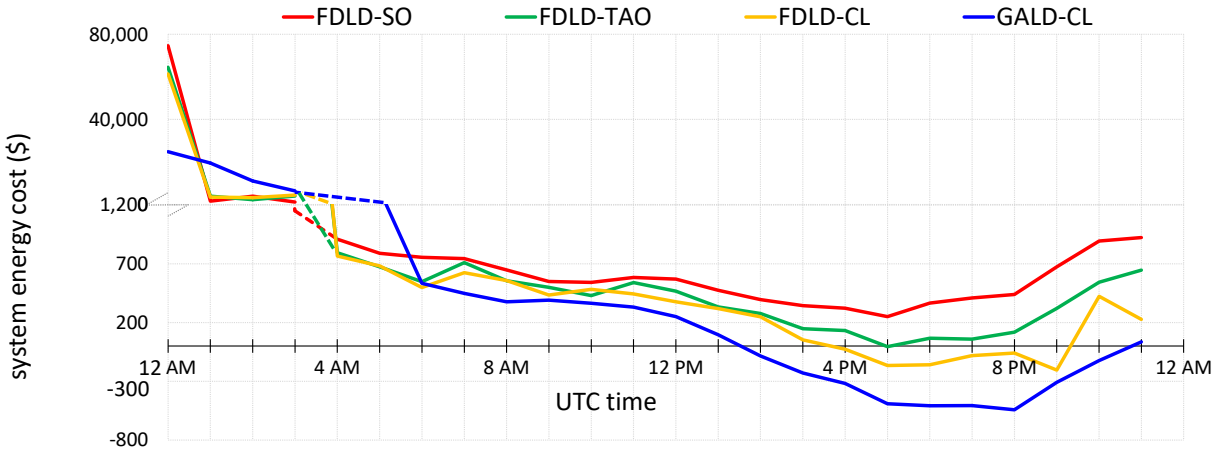
Renewable power generation changes throughout the year depending on a location. As discussed in Section 2.6, data centers generate different amounts of renewable power at each location. Different states have different net metering laws and utility companies from those states have different energy buy-back rates for net metering. In this section, we analyze the impact of net metering (with no peak shaving) and study how the net metering factor impacts energy costs and the behavior of the heuristics. For these experiments, we consider a configuration with eight data centers executing a hybrid workload. The experiments analyze the total system cost for each heuristic utilizing only net metering. Peak shaving is ignored because this study focuses only on the sensitivity analysis of the net metering factor.



(a) four data centers



(b) eight data centers



(c) sixteen data centers

Figure 13: System energy costs for each heuristic over a day for epoch-based analysis, for a configuration with (a) four, (b) eight, and (c) sixteen data center locations running the hybrid workloads

Net metering factor values for data centers were randomly sampled (with uniform distribution) from three value ranges, where buy-back costs for renewable energy are low ($0 \leq \alpha 1 < 0.4$), medium ($0.4 \leq \alpha 2 < 0.7$), and high ($0.7 \leq \alpha 3 \leq 1$). Furthermore, we consider five simulation runs for each set of values and plot standard deviations as shown in Figure 14. In the figure, the system energy costs decrease, but not dramatically as the net metering factor values increase. However, we can observe that GALD-CL is able to exploit net metering better than the other heuristics as α values increase.

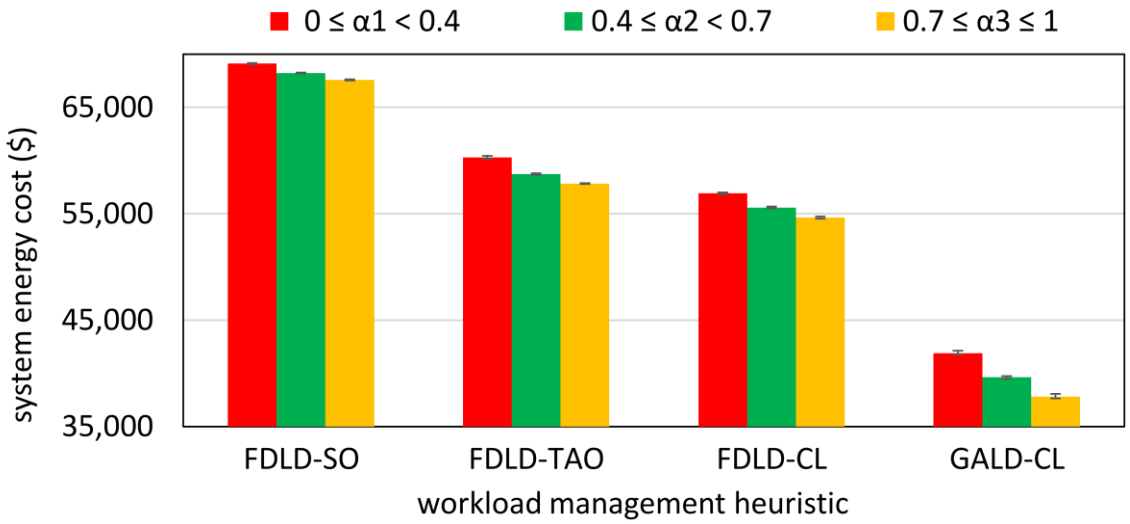


Figure 14: System energy costs for each heuristic over a day for net metering factor sensitivity analysis, for eight data center locations running a hybrid workload

2.7.4.2. PEAK DEMAND PRICE

As discussed in Section 2.6, the set of data centers we are considering for these experiments are heterogeneous and therefore consume different amounts of peak power. Utilities at different locations have different peak prices as shown in Table 2. In this section, we analyze the impact of peak shaving (with no net metering) and study the impact of peak demand price. For these experiments, we consider eight data centers executing a hybrid workload. The experiments analyze

the total system cost for each heuristic utilizing only peak shaving. Net metering is not considered because this study focuses only on the sensitivity analysis of the peak demand price.

Peak demand pricing values for data centers were randomly sampled (with uniform distribution) from three value ranges, where these values are low ($3 \leq p1 < 7$), medium ($7 \leq p2 < 11$), and high ($11 \leq p3 \leq 15$). Furthermore, we consider five simulation runs for each set of values and plot standard deviations as shown in Figure 15. As expected, the system energy costs increase as the peak demand pricing values increase. We observe a significant cost change across $p1$, $p2$, and $p3$ because peak demand cost is one of the two major components of the total system cost (see Equation (10)) and peak demand pricing has a major impact on the peak demand cost component.

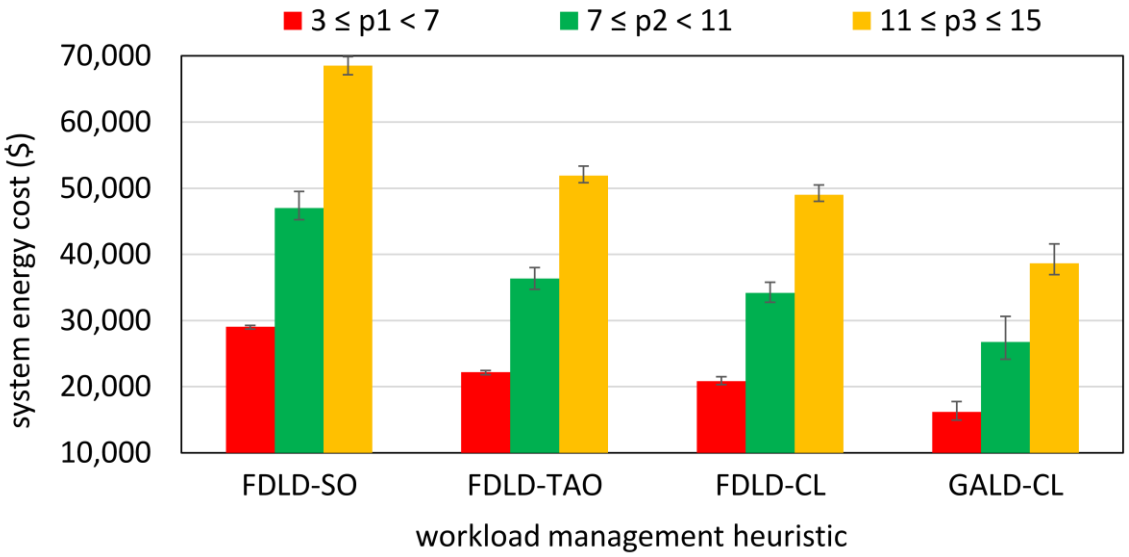


Figure 15: System energy costs for each heuristic over a day for peak demand price sensitivity analysis, for eight data center locations running a hybrid workload

The wide standard deviation for peak shaving shows that the system is highly sensitive to the peak demand price. By comparing both net metering factor and peak demand price sensitivity analyses, we can observe that peak shaving has a bigger impact on system cost than net metering.

The major reason behind this is realistic assumptions of renewable power generation capabilities at each data center. If we consider self-sustainable green data centers [32], [33], [74] and the trend of increasing on-site renewable (solar/wind) power farms, more renewable power will be available for net metering which will have a more significant impact on the system.

2.7.5. TASK ARRIVAL RATE PATTERN ANALYSIS

All of our experiments so far have assumed a sinusoidal task arrival rate pattern as shown in Figure 11. This kind of pattern exists in environments where workload traffic depends on user/consumer interaction and follows their demand during the day, e.g., Netflix [76], Facebook [77]. However, for the environments where continuous computation is needed and the workload pattern is non user/consumer interaction specific, the task arrival rate pattern is usually flat (nearly-constant). Examples of such environments exist in military computing installations (Department of Defense), government research labs (National Center for Atmospheric Research), etc. We conducted a set of simulations to analyze the impact of varying the task arrival pattern. We consider a configuration with eight data centers executing a hybrid workload with both sinusoidal and flat arrival patterns.

Recall that each task-type is characterized by its arrival rate and the estimated time required to complete the task on each of the heterogeneous compute nodes in all P-states. The four GDRM heuristics map execution rates to minimize total energy cost across all data centers with the constraint that the execution rates of all task-types meet their arrival rates (see Equation (1)). Therefore the assignment of execution rates alters with the change in arrival rates, which further affects the system cost. The results shown in Figure 16(c) and Figure 16(d) indicate that the geo-distributed system responds differently for sinusoidal and flat arrival rate patterns.

Overall system energy cost is higher for the sinusoidal workload arrival pattern because it produces higher peak data center power, which further increases the peak demand cost, than the flat workload arrival pattern. Peak shaving significantly reduces the energy costs for both workload patterns. The sinusoidal workload arrival pattern roughly aligns with the pattern of the renewable power generation (see Figure 10), allowing data centers to utilize all of the available renewable power.

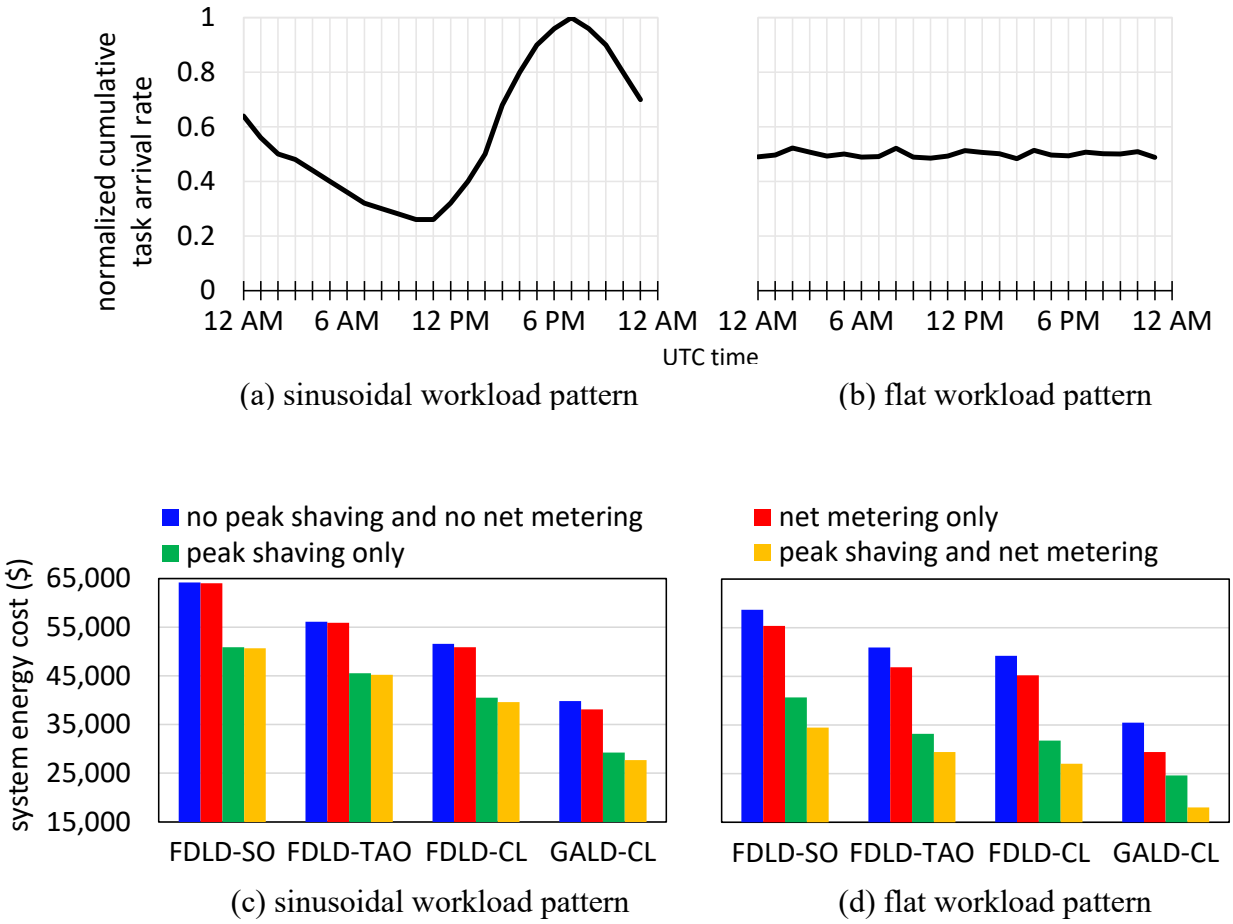


Figure 16: Comparison of (a) sinusoidal and (b) flat workload arrival rate patterns; comparison of system energy costs among heuristics over a day for (c) sinusoidal and (d) flat workload arrival rate patterns for eight data center locations running a hybrid workload analysis, for eight data center locations running a hybrid workload

The flat (nearly-constant) workload arrival rate pattern does not align with the pattern of the renewable power generation. This leaves data centers with excess renewable energy in the second half of the day, which allows heuristics to exploit net metering heavily and produce more cost savings. Thus, heuristics that consider net metering perform better for the flat arrival rate pattern as compared to the sinusoidal arrival rate pattern.

2.8. CONCLUSIONS

We studied the problem of minimizing the energy costs for geographically distributed heterogeneous data centers with the constraint that all tasks complete without being dropped. Renewable energy, peak demand, and co-location interference at data centers have a significant impact on energy consumption. We capture these effects by including net metering, peak shaving, and co-location models in our workload distribution techniques. We analyzed several techniques that possess varying degrees of co-location interference prediction information: (i) a force-directed scheduling technique, FDLD-TAO, that uses task aware over-provisioning to estimate co-location effects for each task-type, (ii) a force-directed scheduling technique, FDLD-CL, that uses co-location models when calculating task execution rates, and (iii) a genetic algorithm combined with a local search technique, GALD-CL, that has information about the co-location models and DVFS P-states.

The primary contributions of our research are to provide analyses of the aspects of the problems and solutions associated with renewable energy, peak demand, and co-location aware resource management in heterogeneous computing systems. We used new peak demand and net metering models that directly consider real-world peak demand prices and net metering policies in calculation of the system electricity cost, and a new co-location interference model created from a

linear regression technique using data from real servers. We demonstrated that including this additional information in the decision process of the resource management heuristics resulted in a lower energy cost. This is achieved by reducing or eliminating node over-provisioning while still meeting all required workload execution rates. Ignoring interference effects altogether can be especially detrimental to overall performance and energy overheads when task execution times deviate far from those expected due to interference. We also demonstrated the importance of net metering and peak shaving by illustrating the impact of the awareness of these factors on reducing operating costs for geographically distributed data centers.

We compared our techniques across various workload profiles, performing a scalability assessment, examining sensitivity to renewable power availability and peak demand pricing parameters, and testing system behavior for different task arrival patterns. Our proposed FDLD-CL and GALD-CL heuristics resulted in 37% and 61% lower operational costs on average than an approach from prior work (represented by the FDLD-SO heuristic) [42]. However, to implement our approach in a real system, it needs to be used with some form of a workload prediction technique, e.g., [78], [79]. Additionally, due to scalability issues of GALD-CL, we recommend using FDLD-CL with a workload prediction technique in systems where the workload profile changes rapidly and therefore requires short epochs, e.g., a few minutes. When the workload profile is not changing rapidly and the workload distribution decisions are provided more time, e.g., an hour, using GALD-CL with a workload prediction technique is a more suitable heuristic.

3. ENERGY AND NETWORK AWARE WORKLOAD MANAGEMENT FOR GEOGRAPHICALLY DISTRIBUTED DATA CENTERS[2]

Cloud service providers are distributing data centers geographically to minimize energy costs through intelligent workload distribution. With increasing data volumes in emerging cloud workloads, it is critical to factor in the network costs for transferring workloads across data centers. For geo-distributed data centers, many researchers have been exploring strategies for energy cost minimization and intelligent inter-data-center workload distribution separately. However, prior work does not comprehensively and simultaneously consider data center energy costs, data transfer costs, and data center queueing delay. In this chapter, we propose a novel game theory-based workload management framework that takes a holistic approach to the cloud operating cost minimization problem by making intelligent scheduling decisions aware of data transfer costs and the data center queueing delay. Our framework performs intelligent workload management that considers heterogeneity in data center compute capability, cooling power, interference effects from task co-location in servers, time-of-use electricity pricing, renewable energy, net metering, peak demand pricing distribution, and network pricing. Our simulations show that the proposed game-theoretic technique can minimize the cloud operating cost more effectively than existing approaches.

3.1. MOTIVATION AND CONTRIBUTION

In recent years, the ever-increasing use of smartphones, wearable devices, portable computers, Internet-of-Things (IoT) devices, etc., has fueled the use of cloud computing. Data-intensive

[2] The research presented in this chapter is published in [137].

applications in the domains of artificial intelligence, distributed manufacturing systems, smart and connected energy, and autonomous vehicles are beginning to leverage cloud computing extensively [80]. To support these applications, cloud service providers are increasingly deploying new data centers that can bolster the capabilities of their cloud computing offerings. While centralized data centers were common in the past, more recently there has been a trend towards deploying data centers across geographically diverse locations [23], [24]. Distributing data centers geographically across the globe leads to many benefits. It brings them closer to customers offering better performance (e.g., latency) and lower network costs. It also provides better resilience to unpredictable failures (e.g., environmental hazards) due to the redundancy that distributed data centers enable.

Another strong motivation to geographically distribute data centers is to reduce electricity costs by exploiting time-of-use (TOU) electricity pricing [25]. The cost of electricity varies based on the time of day and follows the TOU electricity pricing model [26]. Electricity prices are higher when the total electrical grid demand is high and falls during periods when electrical grid demand is low [27]. For non-residential or commercial customers, utilities also charge an additional flat-rate (peak demand) charge based on the highest (peak) power consumed at any instant during a billing period, e.g., month [28]. In some cases, such peak demand charges can be higher than the energy use portion of the electric bill. Moving workloads among geo-distributed data centers is one effective approach to reduce electricity expenditures. This approach allows cloud service providers to allocate workloads to data centers with lower energy costs. This not only reduces operating costs for cloud service providers, but also can reduce cloud computing costs for customers.

Distributing workloads geographically entails an overhead: network costs for transferring workloads and their data between data centers. Many workloads hosted in geographically distributed data centers need to transfer data among data centers for data replication, collection, and

synchronization. Such data movement significantly increases the cloud networking costs. Even though the price of Internet data transfers continues to decline by approximately 30% per year [81], inter-data center traffic is exploding [82]. Moreover, operating energy-efficient data centers at high capacity can sometimes overwhelm the intra-data center servers and networking equipment, causing a delay in processing of the incoming workload. We call this the data center queueing delay, and this reduces cloud provider performance and revenues.

Data centers today are energy intensive, and are estimated to account for around 1% of worldwide electricity use. The total energy used by the world's data centers has doubled over the past decade and some studies claim that it will triple or even quadruple within the next decade [83]. We define cloud operating costs as the dollar energy costs for all geo-distributed data centers plus the dollar inter-data center data transfer costs. Minimizing such costs is important as the annual electricity expenditure for powering data centers is growing rapidly: China's data centers alone are on track to use more energy than all of Australia by 2023 [84]. These energy expenses annually can sometimes even surpass the costs of purchasing all of the data center equipment.

It should be noted that geo-distributed data centers have significant heterogeneity in operating costs and performance, because of aspects including: (a) inter-data center network costs, that are affected by the amount of data transferred among sites; (b) queueing delay caused by the intra-data center network congestion in the data centers executing at high capacity; (c) variable TOU pricing, as data centers are often located in different times zones; (d) the use of on-site green/renewable energy sources, e.g., solar and wind, to various extents across sites; (e) the availability (or absence) of net metering, which is a mechanism that gives renewable energy customers credit on their utility bills for the excess clean energy they sell back to the grid [37]; (f) variable peak demand pricing from utility providers across sites; (g) the availability of diverse

resources within a data center, such as cooling infrastructure, and heterogeneity across compute nodes (e.g., from the perspective of power and performance characteristics); and (h) co-location interference, a phenomenon that occurs when multiple cores within the same multicore processor are executing tasks simultaneously and compete for shared resources, e.g., last-level caches or DRAM. It is important to factor in this heterogeneity while making decisions about workload management in geo-distributed data centers.

The goal of this chapter is to design and evaluate a geographical workload distribution solution for geo-distributed data centers that will minimize the cloud operating cost for executing incoming workloads considering all of the aspects mentioned above. This research is applicable to environments where execution information about the workloads is readily available or can be predicted by some form of a workload prediction technique, e.g., [85], [86]. Examples of such environments that exist in the industry include commercial companies (DigitalGlobe, Google), military computing installations (Department of Defense), and government labs (National Center for Atmospheric Research). More specifically, we propose a novel game theory-based workload management framework that distributes workload among data centers over time, while meeting their performance requirements. The novel contributions of our research can be summarized as follows:

- we formulate the cloud workload distribution problem as a non-cooperative game and design a new Nash equilibrium based intelligent game-theoretic workload distribution framework to minimize the cloud operating cost;
- our framework simultaneously considers energy and network cost minimization, while satisfying workload performance goals;

- our decisions leverage detailed models for a comprehensive set of characteristics that impact cloud operating costs and workload performance, including data center compute and cooling power, co-location performance interference, TOU electricity pricing, renewable energy, net metering, peak demand pricing distribution, data center queueing delay, and the costs involved with inter-data center data transfers.

3.2. RELATED WORK

3.2.1. DATA CENTER ELECTRICITY COST, RENEWABLE POWER, AND PEAK DEMAND

There have been many recent efforts proposing methods to minimize electricity costs across geo-distributed data centers, with the fundamental decisions of the optimization problem relying on a TOU electricity pricing model [44], [87]. Electricity costs are often much higher during peak hours of the day (typically 8 A.M. to 5 P.M.). The electricity cost models, sometimes in combination with a model that considers renewable energy, motivates the use of optimization techniques to minimize energy cost or, if provided a revenue model, to maximize total profit [88], [89]. A few other papers consider peak demand pricing models often in conjunction with energy storage devices/batteries to optimize electricity costs [90], [91].

3.2.2. DATA CENTER NETWORK MANAGEMENT

Many recent papers focus on intelligent workload scheduling among cloud centers to optimize the quality of service (QoS) while increasing cloud operating profits [92], [93]. As data migration among cloud data centers increases network costs, some efforts try to address this issue by proposing techniques that intelligently allocate workloads to reduce overall network costs [94],

[95]. In [96] and [97], the authors propose a multi-objective framework that simultaneously optimizes the resource wastage and migration costs while meeting QoS requirements.

3.2.3. GAME THEORY FOR DATA CENTER RESOURCE MANAGEMENT

A few efforts consider game-theoretic approaches for cost optimization in geo-distributed cloud data centers. In [98], the authors propose a technique to allocate computing resources according to the service subscriber's requirements by using non-cooperative game theory. The authors in [99], [100] propose a bandwidth resource management technique for geo-distributed cloud data centers. In [101], the authors propose an algorithm that tries to reduce data migration costs and maximize the profit of all cloud service providers involved in a federation. In [102], the authors use a cooperative game theory to model the cooperative electricity procurement process of data centers as a cooperative game and show the cost-saving benefits of aggregation. In [103], an approach is proposed to address the tradeoff between QoS and energy consumption in cloud data centers. The approach uses a game-theoretic formulation that allows for appropriate loss of QoS while maximizing the cloud service provider's profits. In [104], the authors consider a cloud data center and smart grid utility demand-response program. The work proposes a cooperative game theory-based technique to migrate workloads between data centers and better utilize the benefits provided by demand response schemes over multiple data center locations. In [105] and [106], the authors propose a non-cooperative game-theoretic workload management technique to minimize data center energy costs, while considering TOU electricity pricing information.

Similar to [105] and [106], our research uses information about TOU electricity pricing and develops a non-cooperative game-theoretic workload management technique to minimize the cloud operating cost. Our framework also uses information about renewable power and net

metering policies at various data center locations like [105]. However, both [105] and [106] do not consider detailed models for data center compute and cooling power, co-location interference, and peak demand pricing distribution. Moreover, these efforts do not consider heterogeneity in workload (applications/task types executing in the data center) and heterogeneity within the data center (types of servers, server rack arrangements, etc.). Our previous work [107] considers many of the above-mentioned aspects. We extend [107] by developing a new inter-data center network cost model, a data center queueing delay model, and considering these factors in our resource management problem. Moreover, we propose a new game theory-based workload management framework that takes a holistic approach to the cloud operating cost minimization problem, which is shown to be more effective than the multiple resource management strategies presented in [107] (Section 3.7 presents a detailed analysis to quantify the improvement).

3.3. SYSTEM MODEL

3.3.1. OVERVIEW

Our framework comprises a geo-distributed-level Cloud Workload Manager (CWM) that distributes incoming workload requests to geographically distributed data centers. Each data center has its own local Data center Workload Manager (DWM) that takes the workload assigned to it by the CWM and maps requests to compute nodes within the data center. We first describe the system model at the geo-distributed level and then provide further details into the models of components at the data center level.

3.3.2. GEO-DISTRIBUTED LEVEL MODEL

We consider a rate-based workload management scheme, where the workload arrival rate

can be predicted over a decision interval called an epoch [85], [86]. In our work, an epoch length T^e is one hour, and a 24-epoch period represents a full day. Within the short duration of an epoch, the workload arrival rates can be reasonably approximated as constant, e.g., the Argonne National Lab Intrepid log shows mostly constant arrival rates over larger intervals of time [55].

Let D be a set of $|D|$ data centers and let d represent an individual data center. We assume that a cloud infrastructure (Figure 17) is composed of $|D|$ data centers, that is, $d = 1,2, \dots, |D|$ and $d \in D$. Let I be a set of $|I|$ task types and i represents an individual task type. We consider $|I|$ task (i.e., workload) types, that is, $i = 1,2, \dots, |I|$ and $i \in I$. A task type $i \in I$ is characterized by its arrival rate and its execution rate, i.e., reciprocal of the estimated time required to complete a task of task type i on each of the heterogeneous compute nodes, in each performance state (P-state). We assume that the beginning of each epoch τ represents a steady-state scheduling problem where the CWM splits the global arrival rate $GAR_i(\tau)$ for each task type i into the local data center level arrival rate $AR_{i,d}(\tau)$ and assigns it to each data center $d \in D$. That is,

$$\sum_{d=1}^{|D|} AR_{i,d}(\tau) = GAR_i(\tau), \quad \forall i \in I. \tag{26}$$

The CWM performs this assignment such that the total cloud operating (energy and network) cost across all data centers is minimized, with the constraint that the execution rates of all task types exceed their arrival rates at each data center, i.e., all tasks complete without being dropped or unexecuted. At the start of each epoch, the CWM calculates (explained in Section 3.3.3.2) a co-location aware data center maximum execution rate $ER_{i,d}$ for each task type i at each data center d. Later, it splits global arrival rate $GAR_i(\tau)$ for each task type i into local data center level arrival rates $AR_{i,d}(\tau)$ such that the data center maximum execution rate $ER_{i,d}(\tau)$ exceeds the corresponding arrival rate $AR_{i,d}(\tau)$, thus ensuring the workload is completed. That is,

$$ER_{i,d}(\tau) > AR_{i,d}(\tau), \quad \forall i \in I, \forall d \in D. \tag{27}$$

Note that $ER_{i,d}(\tau)$ must be strictly greater than $AR_{i,d}(\tau)$ to minimize the expected average queueing delay for task type i at data center d, as discussed later in Section 3.3.3.8.

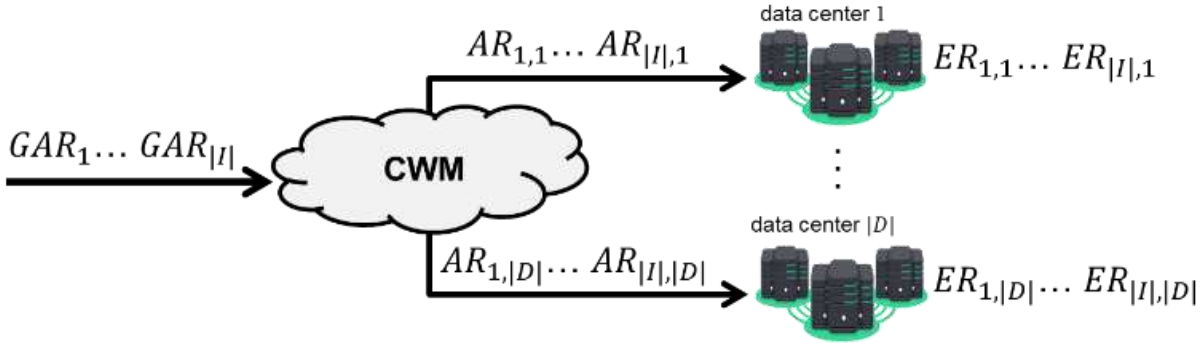


Figure 17: Cloud workload manager (CWM) performing geo-distributed level task (workload) assignment to data centers analysis, for eight data center locations running a hybrid workload

3.3.3. DATA CENTER LEVEL MODEL

3.3.3.1. ORGANIZATION OF EACH DATA CENTER

Each data center d houses NN_d number of nodes and a cooling system comprised of computer room air conditioning (CRAC) units. Let NCR_d be the number of CRAC units. Heterogeneity exists across compute nodes, where nodes vary in their execution speeds, power consumption characteristics, and the number of cores. The number of cores in node n is NCN_n, and NT_k is the node type to which core k belongs.

3.3.3.2. CO-LOCATION AWARE EXECUTION RATES

Tasks competing for shared memory in multicore processors can cause severe performance degradation, especially when competing tasks are memory intensive. The memory-intensity of a

task refers to the ratio of last-level cache misses to the total number of instructions executed. We use a linear regression model that combines a set of disparate features based on the current tasks assigned to a multicore processor to predict the execution time of a target task i on core k in the presence of performance degradation due to interference from task co-location [65].

We classify the task types into memory-intensity classes on each of the node types, and calculate the coefficients for each memory-intensity class using the linear regression model to determine a co-located execution rate for task type i on core k, $CER_{i,k}^{core}(\tau)$ [65]. When considering co-location at a data center d, the co-location aware data center execution rate for task type i is given by:

$$ER_{i,d}(\tau) = \sum_{n=1}^{NN_d} \sum_{k=1}^{NCN_n} CER_{i,k}^{core}(\tau). \quad (28)$$

The linear regression model was trained using execution time data that was collected by executing benchmarks from the BigDataBench 5.0 [108] benchmark suite on a set of server-class multicore processors that define the nodes used in our study (see Table 5 in Section 3.6.2 for node type details). This model for execution time prediction under co-location interference is derived from real workloads and real server machines and has a mean prediction error of approximately 7% [65].

3.3.3.3. DATA CENTER POWER MODEL

We use the detailed data center power model from our prior work [107]. Let $PCR_{d,c}(\tau)$ be the power consumed by a CRAC unit c in data center d. $PN_n(\tau)$ is the power consumption of node n. Each data center is partially powered by either solar power, wind power, or some combination of both. The total renewable power, $PR_d(\tau)$, available at data center d is the sum of the wind and

solar power available at that time. $PR_d(\tau)$ can be zero if no renewable power is available. Let Eff_d be an approximation of the power overhead coefficient in data center d due to the inefficiencies of power supply units. Eff_d is always greater than or equal to 1. The net total power consumed throughout data center d, $PD_d(\tau)$, is calculated as:

$$PD_d(\tau) = \left(\sum_{c=1}^{NCR_d} PCR_{d,c}(\tau) + \sum_{n=1}^{NN_d} PN_n(\tau) \right) \cdot Eff_d - PR_d(\tau). \quad (29)$$

For epoch τ, $PD_d(\tau)$ can be negative if the renewable power available at the data center, $PR_d(\tau)$, is greater than the cooling and computing power.

3.3.3.4. NET METERING MODEL

Net metering allows data center operators to sell back the excess renewable power generated on-site to the utility company. When the excess power is added into the grid, utility companies pay a fraction of the retail price. The value of this factor varies by data center location. This fraction is called the net metering factor, α_d.

3.3.3.5. PEAK DEMAND MODEL

Most utility providers charge a flat-rate (peak demand) fee based on the highest (peak) power consumed at any instant during a given billing period, e.g., month. The peak demand price per kW at data center d is denoted as P_d^{price}. We define $Pc_d^{peak}(\tau)$ as the highest grid power consumed since the beginning of the current month, including the current epoch τ. We define $Pp_d^{peak}(\tau)$ as the highest grid power consumption since the beginning of the current month until

the start of the current epoch τ. The peak power cost increase at data center d, $\Delta_d^{peak}(\tau)$, is then defined as:

$$\Delta_d^{peak}(\tau) = P_d^{price} \cdot \left(Pc_d^{peak}(\tau) - Pp_d^{peak}(\tau)\right) \quad \text{if } Pc_d^{peak}(\tau) \geq Pp_d^{peak}(\tau), \text{ else it is equal to 0.} \tag{30}$$

The peak power cost increase $\Delta_d^{peak}(\tau)$ is calculated in each epoch τ and summed over all epochs in a billing period to calculate total peak power cost.

3.3.3.6. NETWORK COST MODEL

Two principal components of the network cost are the network price per data traffic unit ($/GB), N^{price}, and the amount of data volume (GB) for the number of tasks migrated (outward). We assume that N^{price} is same across all data centers. Let S_i be the size (in GB) of a task type i. The network cost at data center d, $NC_d(\tau)$, is calculated as

$$NC_d(\tau) = \sum_{i=1}^{|I|} \sum_{n=1}^{NN_d} \left(N^{price} \cdot S_i\right). \tag{31}$$

3.3.3.7. DATA CENTER COST MODEL

The electricity price per kWh at data center d is defined as $E_d^{price}(\tau)$. Data center operators can use net metering if the net total power consumed throughout the data center, $PD_d(\tau)$, is negative. For such conditions, the total data center cost $DC_d(\tau)$ for data center d can be defined as

$$DC_d(\tau) = E_d^{price}(\tau) \cdot \alpha_d \cdot PD_d(\tau) + \Delta_d^{peak}(\tau) + NC_d(\tau) \tag{32}$$

where α_d= 1 if $PD_d(\tau)$ is positive and $0 \leq \alpha_d \leq 1$ otherwise.

The first term in (32) represents the TOU electricity cost, the second term represents the peak demand cost, and the third represents the network cost.

3.3.3.8. DATA CENTER QUEUEING DELAY MODEL

To reduce operating costs, cloud service providers tend to allocate more workload to the more energy-efficient data centers, often running them at very high capacity. In some cases, this can overwhelm the data center servers and network equipment causing a queueing delay [109]. This manifests as delay experienced by users of the cloud service. Because of queueing delay, data centers fail to service the incoming workload in a timely manner, which causes revenue loss. There is evidence that lost revenue has a linear relationship to the data center queueing delay for sites such as Google, Bing, and Shopzilla [110]. These assumptions are satisfied by most standard queueing methods, e.g., the 95th percentile of delay under the M/M/1 queue and the average delay under the M/GI/1 processor sharing (PS) queue [106] and [109]. Let $d_{i,d}^{queue}(\tau)$ be the expected average queueing delay for the task type i at data center d, is calculated as:

$$d_{i,d}^{queue}(\tau) = \frac{1}{ER_{i,d}(\tau) - AR_{i,d}(\tau)}. \quad (33)$$

3.4. PROBLEM FORMULATION

We consider a scenario with multiple data centers maintained by a cloud service provider. The data centers can share the workload coming in from various locations. The system is assumed to be under-subscribed in the sense that the system is expected to have enough computation resources to complete the workload without requiring that any tasks be dropped or terminated

before completion. The tasks originate off-site from the data centers, and we do not consider the transmission time and cost from a task origin to a data center. If the CWM migrates a task from one location to another location, there is a data transfer cost (as discussed in Section 3.3.3.6) associated with it. The objective of a CWM is to allocate the workload across geo-distributed data centers to minimize the monetary cloud operating (energy and network) cost of the system (the sum of (32) across all data centers) while ensuring that the workload is completed according to the constraint defined by (27).

The complexity of the CWM workload allocation problem makes it NP-hard [42], and therefore we propose a game theory-based workload management heuristic. Here, we model the cloud workload distribution problem as a non-cooperative game. In a non-cooperative game, there could be a finite (or infinite) number of players who aim to maximize/minimize their objective independently but ultimately reach an equilibrium. For a finite number of players, this equilibrium is called the Nash equilibrium [111].

3.5. NASH EQUILIBRIUM-BASED HEURISTIC

3.5.1. OVERVIEW

Our proposed Nash equilibrium-based intelligent load distribution (NILD) heuristic is a non-cooperative game-theoretic load balancing approach. The following subsections describe the components of the heuristic in more detail.

3.5.2. OBJECTIVE FUNCTION

NILD jointly optimizes the estimated data center (energy and network) cost and the estimated delay cost, as discussed in the following subsections.

3.5.2.1. ESTIMATED DATA CENTER COST

Let J_d be a set of node types in a data center d and j represents an individual node type. In a data center d, there are a total of $NN_{d.j}$ nodes of node type j. Let P_j^D be the average peak dynamic power for node type j. It is calculated by averaging (over all task types) the peak power for each task type i executing on node type j. Let PD_d^{max} be the maximum power dissipation possible at data center d, is calculated as:

$$PD_d^{max} = \left(NCR_d \cdot PCR_{d,c}^{max} + \sum_{j \in J_d} NN_{d,j} \cdot P_j^D \right) \cdot Eff_d. \tag{34}$$

Recall that $PR_d(\tau)$ is the total renewable power available at data center d. Let $PD_{i,d}^E(AR, \tau)$ be the estimated power dissipation possible for each task type i at data center d, is calculated as:

$$PD_{i,d}^E(AR, \tau) = \left(\frac{PD_d^{max} \cdot AR_{i,d}(\tau)}{ER_{i,d}(\tau)} - PR_d(\tau) \right). \tag{35}$$

Let $NC_{i,d}^{max}$ be the maximum network cost possible for each task type i at data center d, is calculated as:

$$NC_{i,d}^{max} = N^{price} \cdot NN_d \cdot S_i \tag{36}$$

Let $NC_{i,d}^E(AR, \tau)$ be the estimated network cost possible for each task type i at data center d, is calculated as:

$$NC_{i,d}^E(AR, \tau) = \frac{NC_{i,d}^{max} \cdot AR_{i,d}(\tau)}{ER_{i,d}(\tau)}. \tag{37}$$

Observe that, at data center d with zero $PR_d(\tau)$ for each task type i, $PD_{i,d}^E(AR, \tau)$ in (35) and $NC_{i,d}^E(AR, \tau)$ in (37) will increase to PD_d^{max} and $NC_{i,d}^{max}$, respectively as the data center arrival

rates $AR_{i,d}(\tau)$ approaches to its maximum execution rate $ER_{i,d}(\tau)$. Thus, we can say that both $PD^{E}_{i,d}(AR,\tau)$ and $NC^{E}_{i,d}(AR,\tau)$ are functions of AR and τ.

Then the estimated data center cost $DC^{E}_{i,d}(AR,\tau)$ incurred by the task type i with a data center maximum execution rate $ER_{i,d}(\tau)$ at data center d, can be calculated by modifying (32) as:

$$DC^{E}_{i,d}(AR,\tau) = E^{price}_{d}(\tau)\cdot\alpha_{d}\cdot PD^{E}_{i,d}(AR,\tau) + \Delta^{peak}_{d}(\tau) + NC^{E}_{i,d}(AR,\tau) \tag{38}$$

where α_d= 1 if $PD^{E}_{i,d}(AR,\tau)$ is positive and $0 \le \alpha_d \le 1$ otherwise.

3.5.2.2. ESTIMATED DELAY COST

The estimated delay cost is associated with the data center queueing delay (33) and captures the lost revenue incurred from the delay experienced by the requests. For this loss, we use the model from [106]. NILD considers the queueing delay within the data center. Let β be a constant known as a delay cost factor. As per (33), the estimated delay cost $DelC^{E}_{i,d}(AR,\tau)$ for the task type i at the data center d, can be calculated as

$$DelC^{E}_{i,d}(AR,\tau) = \beta\cdot AR_{i,d}(\tau)\cdot d^{queue}_{i,d}(\tau). \tag{39}$$

Substituting the value of $d^{queue}_{i,d}(\tau)$ from (33), we get:

$$DelC^{E}_{i,d}(AR,\tau) = \frac{\beta\cdot AR_{i,d}(\tau)}{ER_{i,d}(\tau) - AR_{i,d}(\tau)}. \tag{40}$$

Observe that $DelC^{E}_{i,d}(AR,\tau)$ will increase (to infinity) as the difference $(ER_{i,d}(\tau) - AR_{i,d}(\tau))$ decreases to zero. This shows that the data centers running at very high capacity result in a very high estimated delay cost. Because of this, NILD avoids assigning the

highest possible arrival rates to data centers. Therefore, with a constant $ER_{i,d}(\tau)$ and β, the estimated delay cost becomes a function of AR and τ.

3.5.2.3. ESTIMATED OVERALL COST INCURRED

The goal of NILD is to minimize the overall cost incurred, which has two components: estimated data center cost, defined in (38), and the estimated delay cost, defined in (40). The estimated overall cost $OC_i^E(AR, \tau)$ incurred for the task type i, can be calculated as:

$$OC_i^E(AR, \tau) = \sum_{i=1}^{|I|} \left(DC_{i,d}^E(AR, \tau) + DelC_{i,d}^E(AR, \tau) \right). \quad (41)$$

3.5.3. LOAD DISTRIBUTION AS A NON-COOPERATIVE GAME

In our workload distribution problem, the objective of NILD is to split the global arrival rate $GAR_i(\tau)$ for each task type i into local data center level arrival rates $AR_{i,d}(\tau)$ and assign it to each data center $d \in D$. We model this problem as a non-cooperative game that is played among a set of players. Each player has a set of strategies and the estimated overall cost that results from using each strategy. In this game, each task type i is a player. NILD uses Algorithm 4, as discussed later in Section 3.5.5, to find the strategy AR_i of each player i. Then NILD uses Algorithm 5, as discussed later in Section 3.5.6, to find a feasible cloud workload distribution strategy $AR = \{AR_1, AR_2, \dots, AR_{|I|}\}$ such that $OC_i^E(AR, \tau)$ is minimized. The components of our non-cooperative game are:

- **Players:** a finite set of players, where each task type i is a player; $i = 1,2, \dots, |I|$ (as per Section 3.3.2)
- **Strategy sets:** load distribution strategy of a player $AR_i = \{AR_{i,1}, AR_{i,2}, \dots, AR_{i,|D|}\}$

- **Cost:** each player wants to minimize the overall cost $OC_i^E(AR, \tau)$ associated with its strategy.

3.5.4. NASH EQUILIBRIUM

To obtain the workload distribution strategy for the cloud data centers, the game described in Section 3.5.3 can be solved using a Nash equilibrium, which is the most widely used solution for such games. The Nash equilibrium of our non-cooperative game is a load distribution strategy of the entire cloud $AR = \{AR_1, AR_2, \dots, AR_{|I|}\}$ such that for each player i:

$$AR_i \in \underset{AR_i}{argmin}\, OC_i^E(AR, \tau). \tag{42}$$

If no player can further minimize the overall cost incurred by changing its current strategy to another one, the strategy AR is a Nash equilibrium. In this equilibrium, a player i cannot decrease the overall cost incurred by choosing a different workload distribution strategy AR_i given the other players' workload distribution strategies.

3.5.5. BEST REPLY STRATEGY

In the Nash equilibrium, the strategy profile AR is such that each player's workload distribution strategy is the best reply given to the other players' strategies. The best reply for a player provides a minimum overall cost for that player's workload allocation strategy given the other players' strategies. Therefore, by finding the best reply strategy AR_i for each player i we determine the load distribution strategy of the entire cloud $AR = \{AR_1, AR_2, \dots, AR_{|I|}\}$. The NILD assigns arrival rates to a data center incrementally. After each iteration, the available capacity (maximum execution rate) of a data center decreases. This leftover capacity (execution rate) is

known as the available execution rate. Let $ER_{i,d}^{av}$ be the available execution rate for a task type player i at data center d, is calculated as

$$ER_{i,d}^{av}(\tau) = ER_{i,d}(\tau) - AR_{i,d}(\tau). \tag{43}$$

The problem of finding the best reply strategy depends on finding the optimal workload distribution for a system with each player, i.e., task type i $(\forall i \in I)$, and $|D|$ distributed data centers with available execution rates $ER_{i,d}^{av}$ $(\forall d \in D)$. We can represent the above problem in the following optimization equation to find the best reply:

$$\min_{AR_i} \; OC_i^E(AR, \tau) \tag{44}$$

subject to the constraints defined by (26) and (27).

The decision variable involved in this optimization is $AR_i = \{AR_{i,1}, AR_{i,2}, \dots, AR_{i,|D|}\}$, because we assume that, at equilibrium, the strategies of other players are constants. As our optimization framework has the objective of minimizing the estimated operating costs, we try to assign larger workloads to less costly data centers. We determine whether a data center is expensive by calculating the value for a metric called capacity factor [106]. A lower capacity factor represents a less expensive data center, and vice versa. The capacity factor is nothing but the estimated data center maximum cost that is independent of the arrival rate. Let $CF_{i,d}^E(\tau)$ be the estimated capacity factor of a data center d for task type i. Modifying (38) we get:

$$CF_{i,d}^E(\tau) = E_d^{price}(\tau) \cdot \alpha_d \cdot \left(\frac{PD_d^{max}}{ER_{i,d}^{av}(\tau)} - PR_d(\tau)\right) + \Delta_d^{peak}(\tau) + \left(\frac{NC_{i,d}^{max}}{ER_{i,d}^{av}(\tau)}\right) \tag{45}$$

where α_d=1 if PD_d^{max} is positive and $0 \leq \alpha_d \leq 1$ otherwise.

First sort the data centers based on their estimated capacity factors $CF_{i,d}^{E}(\tau)$. As per [106], let q be the smallest integer (number of data centers) that satisfies

$$\sum_{d=1}^{q}\sqrt{\frac{\beta \cdot ER_{i,d}^{av}(\tau)}{\lambda(\tau)-CF_{i,d}^{E}(\tau)}} < \sum_{d=1}^{q} ER_{i,d}^{av}(\tau) - GAR_i(\tau). \tag{46}$$

where $\lambda(\tau)$ is a Lagrangian multiplier [106], whose value is given by modifying (45) as:

$$\lambda(\tau) = E_q^{price}(\tau)\cdot\alpha_q\cdot\left(\frac{PD_q^{max}}{ER_{i,q}^{av}(\tau)} - PR_q(\tau)\right) + \Delta_q^{peak}(\tau) + \left(\frac{NC_{i,q}^{max}}{ER_{i,q}^{av}(\tau)}\right) + \frac{\beta}{ER_{i,q}^{av}(\tau)} \tag{47}$$

where α_d= 1 if $PD_d(\tau)$ is positive and $0 \le \alpha_d \le 1$ otherwise.

To further simplify (46), let γ be the right side and δ be the left side of (46), their values are given by:

$$\gamma = \sum_{d=1}^{q} ER_{i,d}^{av}(\tau) - GAR_i(\tau). \tag{48}$$

$$\delta = \sum_{d=1}^{q}\sqrt{\frac{\beta \cdot ER_{i,d}^{av}(\tau)}{\lambda(\tau)-CF_{i,d}^{E}(\tau)}}. \tag{49}$$

Therefore, (46) can be rewritten as:

$$\delta < \gamma \tag{50}$$

As per [106], the arrival rate $AR_{i,d}(\tau)$ for a player i at data center d can be calculated as:

$$AR_{i,d}(\tau) = ER_{i,d}^{av}(\tau) - \frac{\gamma}{\delta}\cdot\sqrt{\frac{\beta \cdot ER_{i,d}^{av}(\tau)}{\lambda(\tau)-CF_{i,d}^{E}(\tau)}} \tag{51}$$

if $1 \le d < q$ else it is equal to 0.

The constraint (50) ensures that the workload is distributed starting from the least expensive data center first to the expensive one at the end. For the data center with the lowest $CF_{i,d}^{E}$, the second term in (51) will be the lowest, assigning larger $AR_{i,d}$ to inexpensive data centers, and vice versa. Equation (51) is used to find the best reply strategy $AR_i = \{AR_{i,1}, AR_{i,2}, \dots, AR_{i,|D|}\}$.

Based on (50) and (51), the following $Best\text{-}Reply$ algorithm is formulated to find the optimal load distribution strategy, i.e., the best reply of player i.

ALGORITHM 4: PSEUDO-CODE FOR *Best-Reply* STRATEGY

inputs: global arrival rate: $GAR_i(\tau)$
available execution rates: $ER_{i,1}^{av}(\tau), \dots, ER_{i,|D|}^{av}(\tau)$
capacity factors: $CF_{i,1}^{E}(\tau), \dots, CF_{i,|D|}^{E}(\tau)$
output: load distribution strategy for a player : $AR_i = \{AR_{i,1}, \dots, AR_{i,|D|}\}$

1. sort data centers in ascending order of capacity factors
   i.e., $CF_{i,1}^{E}(\tau) \leq \cdots \leq CF_{i,d}^{E}(\tau) \leq \cdots \leq CF_{i,|D|}^{E}(\tau)$
2. $q = |D|$
3. initialize γ and δ using (48) and (49), respectively
4. **while** $\gamma > \delta$
5. $\quad \gamma = \gamma - ER_{i,q}^{av}(\tau)$
6. $\quad AR_{i,q} = 0$
7. $\quad q = q - 1$
8. $\quad$ update δ using (49)
9. **for** each data center in $\{1,2, \dots, q\}$
10. $\quad$ calculate $AR_{i,d}(\tau)$ using (51)

The $Best\text{-}Reply$ algorithm takes global arrival rate, available execution rates, and capacity factors of player i as inputs. As the output, it determines the best reply strategy $AR_i = \{AR_{i,1}, AR_{i,2}, \dots, AR_{i,|D|}\}$. First, the data centers are sorted based on their capacity factors (step 1). The heuristic aims to assign larger workloads to less expensive data centers. It initializes the values of q, γ, and δ (steps 2 and 3). We use the while loop (steps 4-8) to find the smallest index data center that satisfies (50). In the loop, the algorithm does not assign any workload to the last (most expensive) data center i.e., $AR_{i,q} = 0$, if the $(q-1)^{th}$ data center's available execution rate is

capable of satisfying (50). This while loop updates the values of q, γ, and δ and continues until the condition (50) or $\gamma > \delta$ is met. Finally, the algorithm determines $AR_{i,d}(\tau) \in AR_i$ for each data center using (51).

3.5.6. NILD HEURISTIC

We designed the final NILD algorithm to compute the Nash equilibrium of the non-cooperative game. It uses the $Best\text{-}Reply$ algorithm explained in the previous section. The CWM uses Algorithm 5 shown below to assign (i.e., split) the global workload arrival rates to data center level arrival rates for all task types. Each player i computes its optimal $Best\text{-}Reply$ strategy (steps 3-8), i.e., it determines $AR_{i,d}(\tau)$ for each data center i. The heuristic then calculates the $norm$, by summing the absolute value of the difference among its operating costs across all players (task types) from the current l^{th} iteration and the previous $(l-1)^{th}$ iteration. This continues until the

ALGORITHM 5: PSEUDO-CODE FOR NILD HEURISTICS

inputs: global arrival rates: $GAR_1(\tau), \dots, GAR_i(\tau), \dots, GAR_{|I|}(\tau)$
data center maximum execution rates: $ER_{1,1}(\tau), \dots, ER_{i,d}(\tau), \dots, ER_{|I|,|D|}(\tau)$
output: cloud load distribution strategy: $AR = \{AR_1, \dots, AR_i, \dots, AR_{|I|}\}$

1. **initialize:**
   iteration number: $l = 0$
   load distribution strategy: $AR_i = 0$
   estimated overall cost incurred: $OC_i^E(AR, \tau) = 0$
   tolerance: $\varepsilon = 10^{-3}$
   $flag = continue$
2. **while** $flag = continue$
3. **for** each i in $\{1,2, \dots, |I|\}$
4. **for** each d in $\{1,2, \dots, |D|\}$
5. calculate $ER_{i,d}^{av}(\tau)$ using (33)
6. calculate $CF_{i,d}^E(\tau)$ using (45)
7. find optimal AR_i using $Best\text{-}Reply$ algorithm
8. calculate $OC_i^E(AR, \tau)$ using (41)
9. $norm = \sum_{i=1}^{|I|} \left| OC_i^{E(l-1)}(AR, \tau) - OC_i^{E(l)}(AR, \tau) \right|$
10. **if** $norm < \varepsilon$
11. $flag = stop$

algorithm comes to an equilibrium i.e., the difference in the total operating cost across all the players in successive iterations (i.e., $norm$) is less than the pre-defined stopping criterion (tolerance ε). The value of ε is determined empirically through simulation studies to provide a value that gives the system the best possible performance. Here, each player determines its $Best\text{-}Reply$ strategy using the current load distribution strategies of other players and updates its strategy.

3.6. SIMULATION ENVIRONMENT

3.6.1. COMPARISON HEURISTICS

We compare our proposed NILD heuristic with (a) co-location aware force-directed load distribution (FDLD) [107], (b) genetic algorithm load distribution (GALD) [107], and (c) Nash equilibrium based simple load distribution (NSLD) [106]. FDLD [107] is a variation of force-directed scheduling [68], frequently used for optimizing semiconductor logic synthesis. FDLD is an iterative heuristic that selectively performs operations to minimize system forces until all constraints are met. The Genitor style [69] GALD has two parts: a genetic algorithm-based CWM and a local data center level greedy heuristic that is used to calculate the fitness value of the genetic algorithm. The local greedy heuristic has information about task-node power dynamic voltage and frequency scaling (DVFS) models [107]. [106] also proposes a game-theoretic formulation for the load distribution problem. However, it addresses a much simpler problem, with homogeneous workloads and data centers, and a simplistic data center cost model. We adapt the approach proposed in [106] to our heterogeneous environment we have outlined in Section 3.3, and create the NSLD. The simpler models used NSLD affect the data center and delay costs. They also affect the data center capacity factor and arrival rate calculation, which changes the $Best\text{-}Reply$ strategy

of a player (task type). In contrast, our proposed NILD framework integrates detailed models for data center compute and cooling power, co-location interference, net metering, peak demand, and network pricing distribution. Moreover, we also consider heterogeneity in workload (applications/task types executing in the data center) and heterogeneity within the data center (types of servers, server rack arrangements, etc.), with more detailed models to calculate the latency and power overhead for computation and communication.

3.6.2. EXPERIMENTAL SETUP

Experiments were conducted for three geo-distributed data center configurations containing four, eight, and sixteen data centers. Locations of the data centers in the three configurations were selected from major cities around the continental United States to provide a variety of wind and solar conditions among sites and at various times of the day (Figure 18). The sites of each configuration were selected so that each configuration would have an even east coast to west coast distribution to better exploit TOU pricing, peak demand pricing, net metering, and renewable

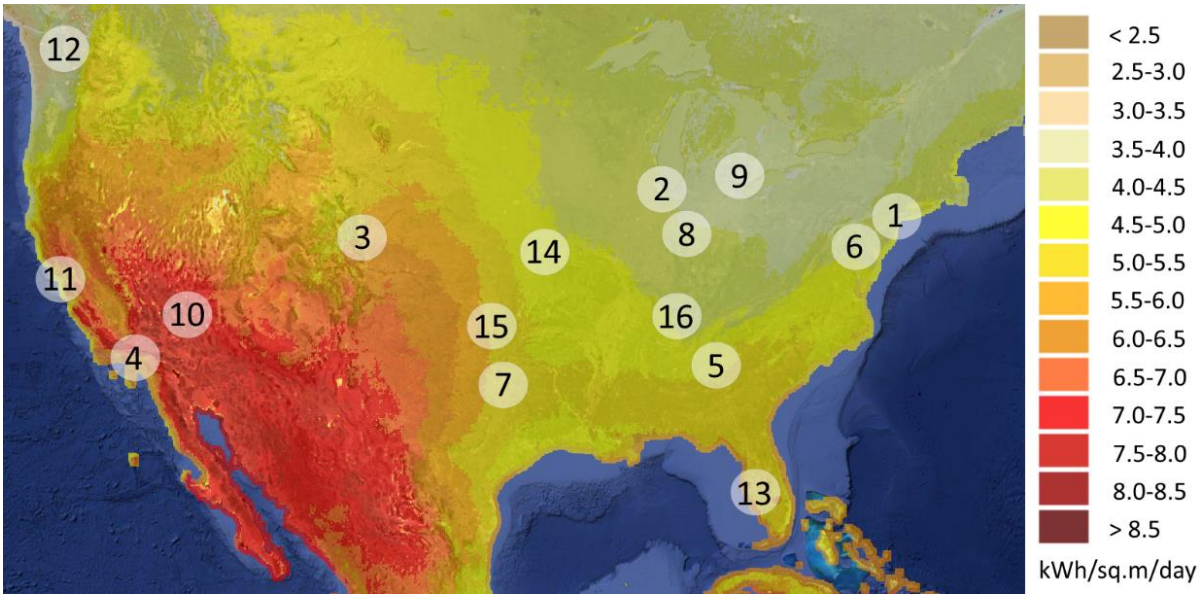


Figure 18: Location of simulated data centers overlaid on solar irradiance intensity map (average annual direct normal irradiance); wind data collected but not shown [41]

power. Each data center comprises 4,320 nodes arranged in four aisles, and is heterogeneous within itself, having nodes from either two or three of the node types given in Table 5, with most locations having three node types and per-node core counts that range from 4 to 12 cores.

Table 5: Node Processor Types Used in Experiments

Intel processor	# cores	L3 cache	frequency range
Xeon E3-1225v3	4	8MB	0.8 - 3.20 GHz
Xeon E5649	6	12MB	1.60 - 2.53 GHz
Xeon E5-2697v2	12	30MB	1.20 - 2.70 GHz

The delay cost factor (β) used during NILD allocations was determined empirically (see Section 3.7.4.2) and set to 0.1. The time of each epoch τ was set to be one hour. The network prices used in the experiments were taken from Amazon Web Services [112]. The TOU electricity prices and the peak demand prices were taken directly from the utility. We assume that each data center has peak renewable power generating capacity equivalent, or slightly more at some locations, to its maximum power consumption. Solar and wind data were obtained from the National Solar Radiation Database [63]. At most locations, the net metering factor α_d is 1; in rare cases, it is less than 1; and in some cases, it is 0, i.e., net metering is not available at that location [37], [75].

Table 6: Task Types Used in Experiments

benchmark	workload type	dataset	size
LDA	offline analytics	Wikipedia entries	233 MB
K-means	offline analytics	Facebook network	650 MB
Naive Bayes	offline analytics	Amazon reviews	500 MB
Image-to-Text	artificial intelligence	Microsoft COCO	600 MB
Image-to-Image	artificial intelligence	cityscapes	100 MB

Organizations are heavily using platform as a service (PaaS) offerings from cloud providers. These services mainly involve data warehouse, data analytics, and machine learning/artificial intelligence workloads [113], [114]. Due to the popularity of such workloads among cloud providers, we use data intensive (e.g., offline analytics and artificial intelligence) workloads from the BigDataBench 5.0 [108] benchmark suite. Table 6 summarizes the workloads from this suite that we consider in our work. Task execution times and co-located performance data for tasks of the different memory intensity classes were obtained from running the benchmark applications on the nodes listed in Table 5 [65]. Figure 19 depicts a synthetic sinusoidal task arrival rate pattern and a flat task arrival rate pattern with line plots. The left y-axis is normalized to the highest total global workload arrival rate. A sinusoidal task arrival rate pattern exists in environments where the workload traffic depends on consumer interaction and follows their demand during the day, e.g., Netflix [76], Facebook [77]. However, for the environments where continuous computation is needed and the workload pattern is non-user/consumer interaction specific, the task arrival rate pattern is usually flat (nearly constant). Examples of such

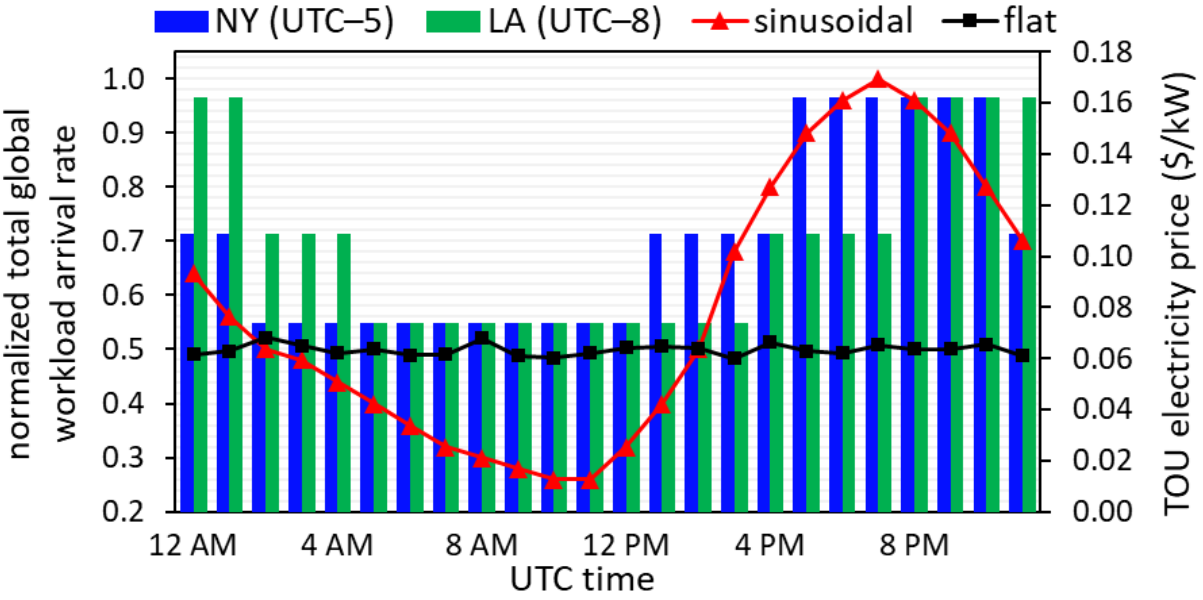


Figure 19: Baseline task arrival rate and TOU prices at two sites (New York, Los Angeles)

environments exist in military computing installations (Department of Defense) and government research labs (National Center for Atmospheric Research). For reference, Figure 19 also depicts TOU prices for New York (east coast) and Los Angeles (west coast) data center locations with bar plots.

3.7. EXPERIMENTS

3.7.1. COST COMPARISON OF HEURISTICS

Our first set of experiments analyzes the total system energy cost for each heuristic described in Section 3.6.1. For each heuristic, we consider four variants: (a) peak shaving, net metering, and network (PS, NM, NW) unaware; (b) only network (NW) aware; (c) only peak shaving and net metering (PS, NM) aware; and (d) peak shaving, net metering, and network (PS, NM, NW) aware. Heuristic variants that are referred to as "peak shaving unaware" or "network unaware" do not include the peak demand pricing factor or the network cost, respectively, in their objective functions but consider them while calculating the total monthly operating cost at the end of the

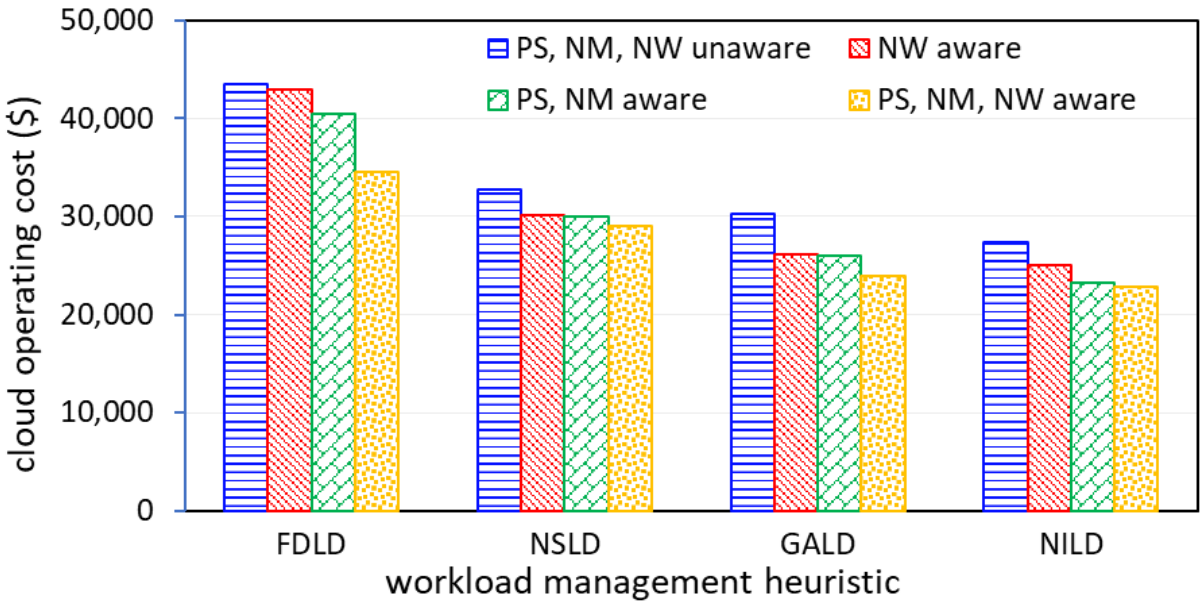


Figure 20: Cloud operating costs for each heuristic over a day, for a configuration with four data centers

billing period. This experiment used a data center configuration with four locations running a sinusoidal workload pattern. The cloud operating costs were calculated over a duration of a day. The results are shown in Figure 20.

For each heuristic; the 'PS, NM, NW aware' variant produced the best results. This validates our consideration of peak shaving, net metering, and network cost during the cloud workload distribution to more effectively minimize overall energy costs. It can also be observed that the FDLD heuristic performed the worst, severely over-provisioning nodes, and resulting in high operating costs. The GALD heuristic has information about task-node power (DVFS) models [107], allowing it to make better task placement decisions than FDLD. The NSLD [106] heuristic considers a simplistic view of data center compute and cooling power, while also ignoring co-location interference, net metering, and peak demand pricing distributions, and because it was designed for a different system model it is unable to perform as well as our NILD framework. The performance of the 'PS, NM aware' variant of NILD heuristic came close to its 'PS, NM, NW aware' variant. The 'PS, NM, NW aware' variant of NILD heuristic outperformed all other approaches. This heuristic minimizes the operating costs for all players/task types independently but ultimately reaches an equilibrium. Because of the non-cooperative nature of this method, each player/task type determines the lowest possible operating cost with its workload allocation strategy while considering the current load distribution strategies of other players.

3.7.2. DATA CENTER SCALABILITY ANALYSIS

3.7.2.1. COST REDUCTION COMPARISON

In this experiment, we analyze heuristic performance for larger problem sizes. Simulations with the sinusoidal workload patterns were analyzed for eight and sixteen data center configurations in addition to the previously discussed four data center configuration. For each configuration, the percentage performance improvement of each heuristic over the 'PS, NM, NW unaware' FDLD variant is given in Table 7.

Table 7: Cloud Operating Cost Reduction Comparison

	heuristic	**PS, NM, NW unaware**	**NW aware**	**PS, NM aware**	**PS, NM, NW aware**
4 data centers	FDLD	0.0%	1.3%	6.9%	20.5%
	NSLD	24.7%	30.8%	31.1%	33.1%
	GALD	30.3%	39.8%	40.3%	44.8%
	NILD	37.0%	42.5%	46.5%	47.5%
8 data centers	FDLD	0.0%	0.6%	23.5%	31.0%
	NSLD	27.1%	29.1%	29.5%	31.4%
	GALD	33.7%	32.7%	48.6%	49.2%
	NILD	47.0%	50.5%	53.8%	54.3%
16 data centers	FDLD	0.0%	0.4%	22.6%	34.7%
	NSLD	32.6%	34.5%	34.7%	36.8%
	GALD	31.7%	22.4%	40.3%	40.4%
	NILD	46.9%	50.6%	50.5%	54.2%

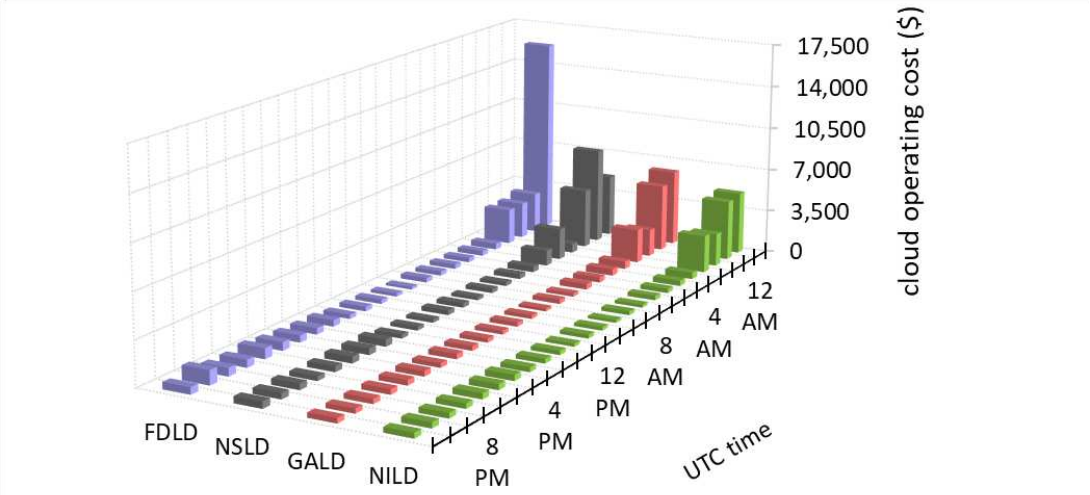
PS = peak shaving, NM = net metering, NW = network

3.7.2.2. EPOCH BASED ANALYSIS

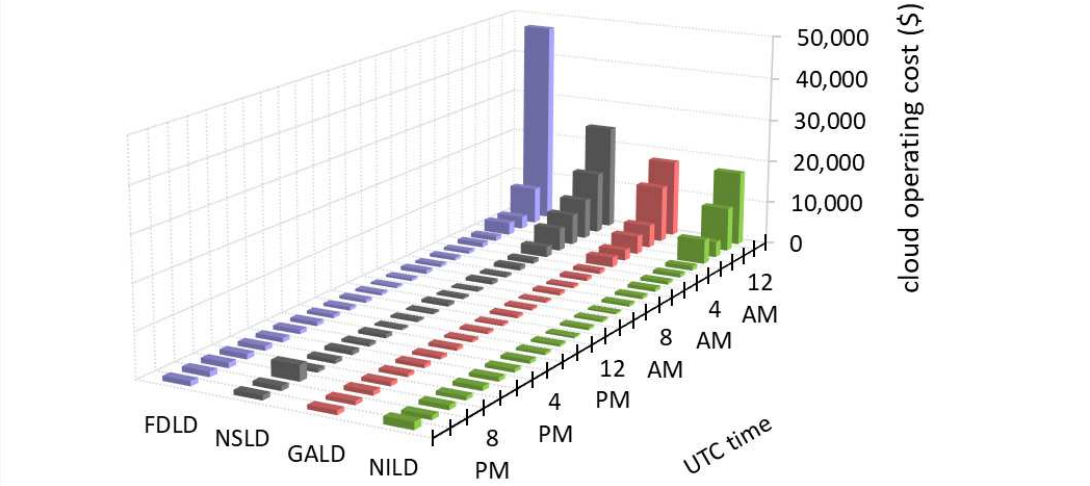
For most of our experiments, we analyzed the total system cost for each heuristic over one day. Figure 21 shows a detailed view of the cloud operating cost at one-hour intervals over a day

for four, eight, and sixteen data centers executing a sinusoidal workload. We consider the 'PS, NM, NW aware' variants of all workload management heuristics in this study. The operating cost for each heuristic is very high during the first epoch in the figure because the period for which the results are shown represents the first day of the month where the initial peak demand cost is added. This effect stays there for the first day and would not be present for other days of the month. After a few epochs, the performance of the GALD came close to the NILD but could not surpass its performance in general.

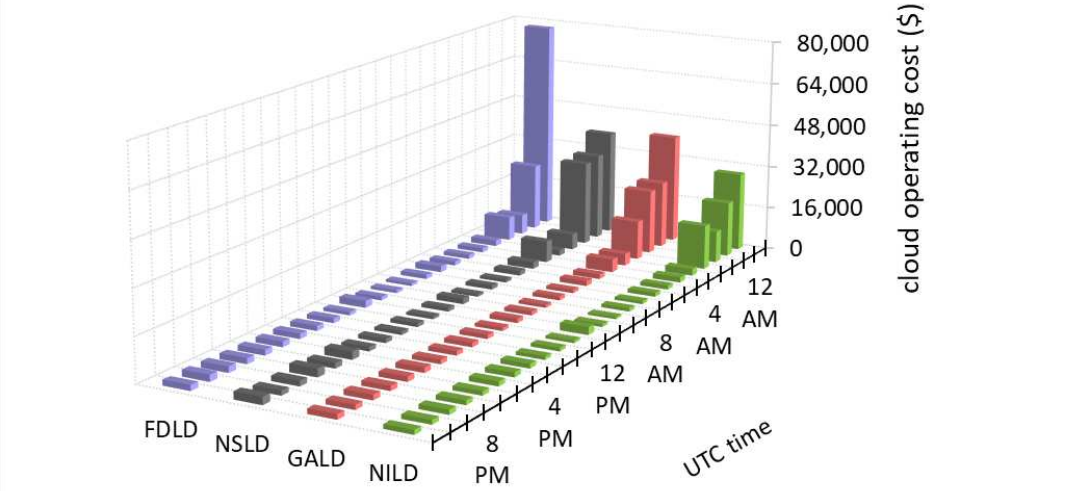
We discussed the notion of the data center queueing delay in Section 3.3.3.8. It is calculated using (33). We also discussed how important it is for cloud service providers to minimize this delay. For this experiment, we considered the 'PS, NM, NW aware' variant of each heuristic for a group of eight data centers executing sinusoidal workload over a day. In Figure 22, the colored data points represent the queueing delay (summed over 8 data centers) values for all heuristics for each epoch over a day. As per (33), the queueing delay increases as the denominator's value (the difference '$ER_{i,d}(\tau) - AR_{i,d}(\tau)$') decreases. The increase in the queueing delay across heuristics indicates the times during which the data centers are running at very high or nearly maximum capacity ($ER_{i,d}(\tau)$) due to the resource allocation decisions ($AR_{i,d}(\tau)$) made by the heuristics. The results shown in Figure 22 reveal that the FDLD, NSLD, and GALD heuristics fail to keep the delay small for certain epochs, whereas the NILD heuristic maintains a lower queueing delay compared to the other heuristics over the day (on average) across epochs.



(a) four data centers



(b) eight data centers



(c) sixteen data centers

Figure 21: Cloud operating costs for each heuristic over a day, for a configuration with (a) four, (b) eight, and (c) sixteen data centers

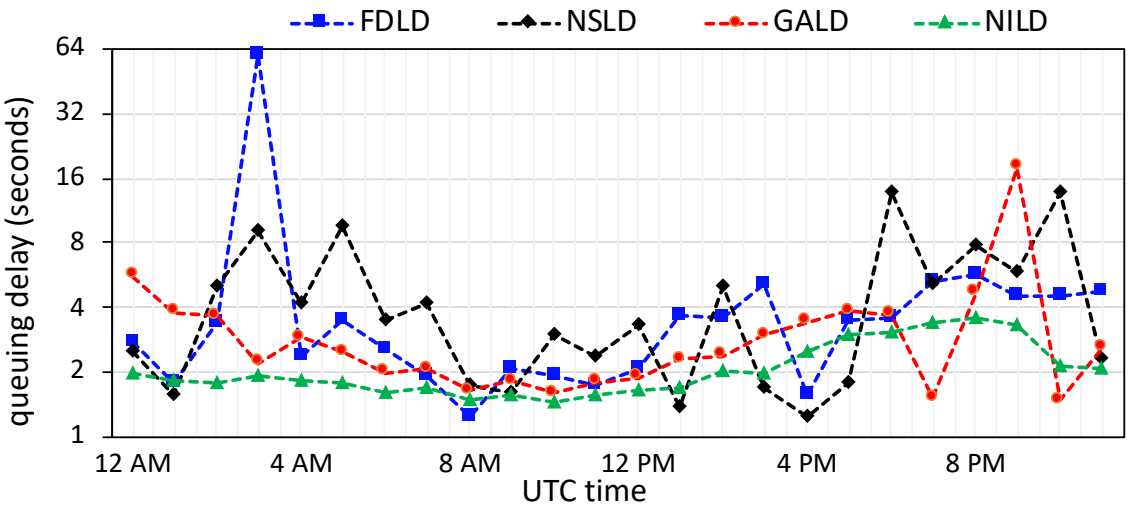


Figure 22: Data center queueing delay comparison among heuristics over a day executing sinusoidal workload for eight data centers

3.7.2.3. HEURISTIC RUNTIME ANALYSIS

The GALD heuristic was limited to a runtime of approximately one hour (epoch length) for all experiments. The ability to change P-states via DVFS gives the GALD a powerful advantage over the FDLD heuristic (which does not take advantage of different P-states in compute nodes) but it also means that GALD takes longer to execute. Table 8 shows the approximate execution time of each heuristic for various problem sizes. The FDLD heuristic does not show better cost reduction than GALD but it has the advantage of reaching a solution more quickly. However, NSLD and NILD took a few seconds to execute. This is beneficial in cases where the workload manager must make the allocation decisions quickly. The NSLD heuristic reached the solution

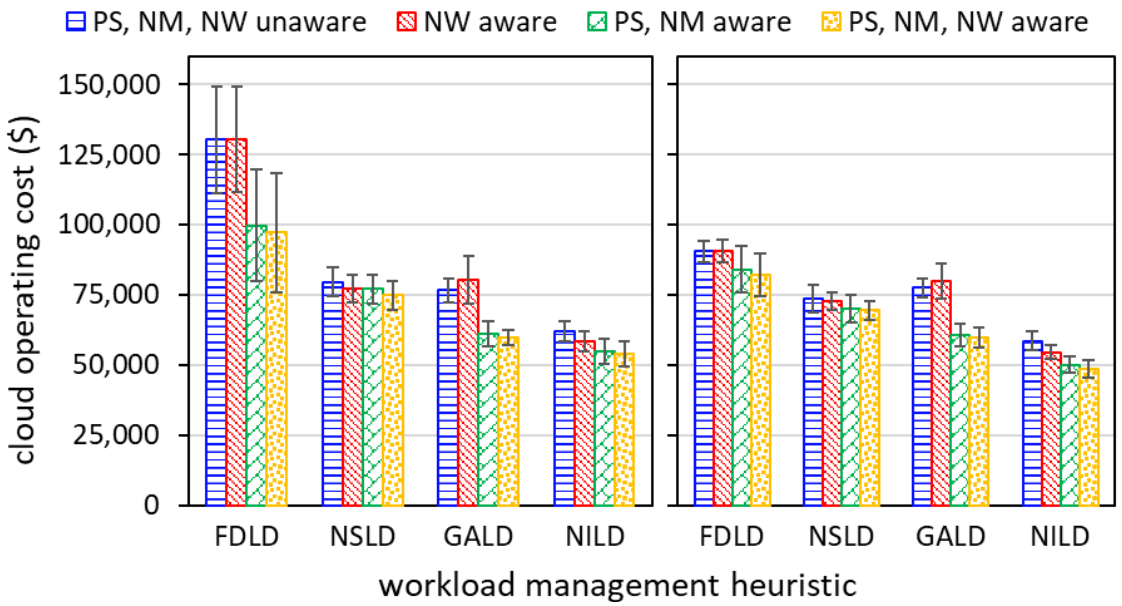
Table 8: Heuristic Runtime Comparison

# data centers	FDLD runtime	NSLD runtime	NILD runtime
4	~ 1 min	~ 1 seconds	~ 2 seconds
8	~ 3 mins	~ 2 seconds	~ 8 seconds
16	~ 12 mins	~ 7 seconds	~ 30 seconds

quicker than NILD because it uses a simplistic data center cost model. The runtime of FDLD, NSLD, and NILD can be further reduced by running simulations on powerful machines.

3.7.3. TASK ARRIVAL RATE PATTERN ANALYSIS

For this experiment, we considered a configuration with eight data centers executing both sinusoidal and flat workload arrival patterns as shown in Figure 19. We conducted 10 simulation runs for each workload pattern and analyzed its impact. For each run, the arrival rate values were randomly sampled (with a normal distribution). We used the original arrival rate values, shown in Figure 19 as mean and used 20% of the mean as standard deviation. We analyzed the cloud operating cost variation for each heuristic over a day by plotting the mean cost with the standard error bars as shown in Figure 23. Recall that each task type is characterized by its arrival rate and the estimated time required to complete the task on each of the heterogeneous compute nodes. The



(a) sinusoidal workload pattern (b) flat workload pattern

Figure 23: Comparison of cloud operating costs among heuristics over a day for (a) sinusoidal and (b) flat workload arrival rate patterns for eight data centers

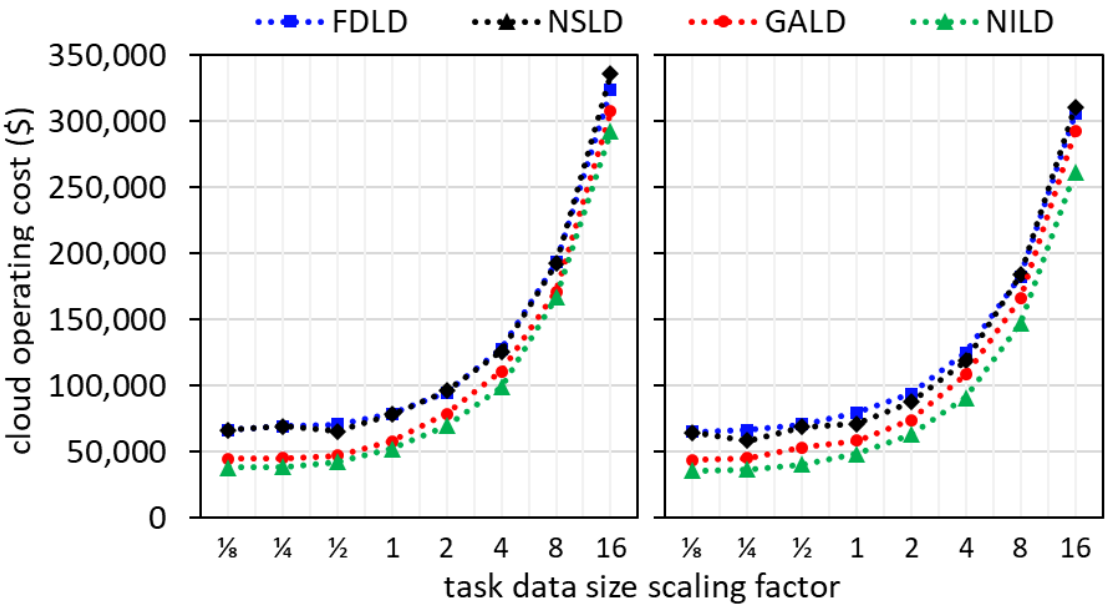
heuristics distribute the workload to minimize total energy cost across all data centers with the constraint that the execution rates of all task types meet their arrival rates (27). Therefore, the workload assignment was altered with the change in incoming arrival rate pattern, which further affected the system cost. The results are shown in Figure 23(a) and Figure 23(b) indicate that the geo-distributed system responded differently for sinusoidal and flat arrival rate patterns. The percentage difference between sinusoidal and flat arrival rate patterns' results for 'PS, NM, NW aware' variants (yellow bars) of FDLD, NSLD, GALD, and NILD are 16%, 7%, 1%, and 10%, respectively.

Overall cloud operating cost was higher for the sinusoidal workload arrival pattern because it produced higher peak demand costs and consumed the most renewable power available during the day as compared to the flat workload arrival pattern. The wide standard deviation for FDLD variants shows that the FDLD heuristic is sensitive to the varying arrival rates. For FDLD, slight variations in arrival rates change final allocations significantly that affect peak power costs. This causes the operating cost to fluctuate significantly. The NSLD and GALD heuristics are not as sensitive as FDLD, while NILD is the most consistent among all heuristics. We also notice that some simulation results for all three heuristics show that the NW aware heuristic variants perform worse than their NW unaware counterparts. This happens mostly with the GALD heuristic executing sinusoidal workload pattern. In such cases, the heuristic only considers the network costs while making allocation decisions and does not consider the peak shaving and net metering. This increases the peak demand costs and also does not allow heuristics to sell the excess energy back increasing the overall cloud operating cost.

3.7.4. SENSITIVITY ANALYSIS

3.7.4.1. TASK DATA SIZE

We performed this experiment to analyze the impact of task data size on the cloud operating costs. Recall that, as per (31), two principal components of the network cost are the price per data traffic unit ($/GB), N^{price}, and the amount of data volume (GB) for the number of tasks migrated (outward). Here the data volume depends on the amount of data migrated per task. If a task migrates from one data center location to another, it transfers various types and amounts of data for each task type. We used a workload that was a mixture of offline analytics and artificial intelligence task types as shown in Table 6. This experiment considered all heuristics for a group of eight data centers executing both sinusoidal and flat workloads with different task data sizes. We considered the 'PS, NM, NW aware' variants of all workload management heuristics in this study.



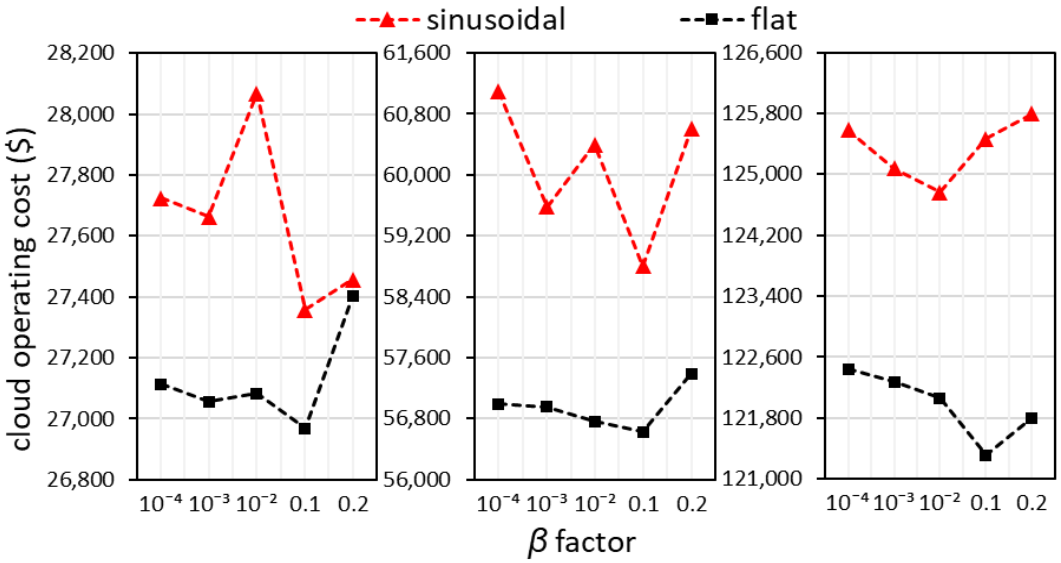
(a) sinusoidal workload pattern (b) flat workload pattern

Figure 24: Impact of task data size scaling on cloud operating costs among heuristics over a day for (a) sinusoidal and (b) flat workload arrival rate patterns for eight data centers

The results from Figure 24 show that the cloud operating costs increased with the increase in task data size. Here, the network costs increased with an increase in the amount of data transferred. For all task data sizes, our proposed NILD framework performed the best overall. For the small values of task data sizes, both FDLD and NSLD performed the worst, while GALD performed better but not as good as NILD. For the large values of task data sizes, GALD's performance degraded (as compared to its performance for the small data sizes) but NSLD performed the worst.

3.7.4.2. NILD DELAY COST FACTOR (β)

As discussed in Section 3.5.5, NILD uses the $Best\text{-}Reply$ algorithm to determine arrival rates. It uses the delay cost factor, β, while calculating the arrival rates (51). In this experiment, we study the impact of β on the cloud operating cost. We considered configurations with four, eight, and sixteen data centers executing both sinusoidal and flat workload arrival patterns. We considered the 'PS, NM, NW unaware' variants of all workload management heuristics in this study.



(a) four data centers (b) eight data centers (c) sixteen data

Figure 25: Impact of the delay cost factor, β, on cloud operating costs among heuristics over a day for both sinusoidal and flat workload arrival rate patterns, for a configuration with (a) four, (b) eight, and (c) sixteen data centers

Results from this experiment in Figure 25 show that the cloud operating costs for both sinusoidal and flat workloads increased with the increase in β. As per (51), the larger values of β make large adjustments in the arrival rate, and vice versa. For the configuration with sixteen data centers, when we increased β beyond 0.2, the NILD failed to reach equilibrium. Whereas for the other configurations, NILD reached equilibrium quickly and CWM could not make fine adjustments in the arrival rate to further minimize the operating costs. When we decreased the value of β very low (smaller than 0.1), NILD reached equilibrium slowly and could not make appropriate adjustments in the arrival rates. The best value of β found from this experiment (0.1) was the one we used for all experiments.

3.8. CONCLUSIONS

In this chapter, we studied the problem of workload distribution among geo-distributed cloud data centers to minimize cloud energy and network costs while ensuring all tasks complete without being dropped. We used a new cloud network model that directly considers the real-world data transfer prices and the amount of data migrated out of the cloud data centers. Our framework considers heterogeneity within the data centers and in task types used in the workload. It is aware of data center cooling power, time-of-use (TOU) electricity pricing, green renewable energy, net metering, peak demand pricing distribution, inter-data center network, and data center queueing delay. We formulated the cloud workload distribution problem as a non-cooperative game. We proposed a game-theoretic workload management technique for minimizing the cloud operating cost. However, to implement our approach in a real system, it needs to be used with some form of a workload prediction technique, e.g., [85], [86], to avoid delays with workload allocation. For our complex (NP-hard) problem and system environment, it should also be noted that the Nash

equilibrium-based game-theoretic technique may not always achieve global optima solutions. However, the obtained solutions are still superior to those obtained with state-of-the-art frameworks, as demonstrated in our experiments. Apart from the data analytics and artificial intelligence workloads, the real-world cloud workloads also include web-service, search, mobile services, etc. Such time-critical workloads cannot be transferred without migration penalties, extra network processing, and scheduling costs. However, our framework can be extended for such time-critical workloads, e.g., by using a priority scheduling approach where time-critical workloads are given higher priority for local scheduling at (or near) the source of the request, to minimize latencies. We compared our new game-theoretic NILD technique with three state-of-the-art techniques (FDLD, GALD [107], and NSLD [106]). We analyzed their performance by comparing the cloud operating cost reduction, performing a scalability assessment, examining sensitivity to task data size, testing system behavior for different task arrival patterns, and comparing data center queueing delays. We demonstrated that NILD performed the best when including information about peak demand charges, net metering policies, network costs, and intra-data center queueing delay. The best performing NILD heuristic from this research achieved 43%, 16%, and 33% cost reductions on average than the FDLD and GALD from [107], and NSLD from [106], respectively. Additionally, the runtime of the NILD heuristic is much lower than the FDLD and GALD heuristics, and is suitable in case of time-critical workload scheduling.

4. A SURVEY ON MACHINE LEARNING FOR GEO-DISTRIBUTED CLOUD DATA CENTER MANAGEMENT[3]

Cloud workloads today are typically managed in a distributed environment and processed across geographically distributed data centers. Cloud service providers have been distributing data centers globally to reduce operating costs while also improving quality of service by using intelligent workload and resource management strategies. Such large scale and complex orchestration of software workload and hardware resources remains a difficult problem to solve efficiently. Researchers and practitioners have been trying to address this problem by proposing a variety of cloud management techniques. Mathematical optimization techniques have historically been used to address cloud management issues. But these techniques are difficult to scale to geo-distributed problem sizes and have limited applicability in dynamic heterogeneous system environments, forcing cloud service providers to explore intelligent data-driven and Machine Learning (ML) based alternatives. The characterization, prediction, control, and optimization of complex, heterogeneous, and ever-changing distributed cloud resources and workloads employing ML methodologies have received much attention in recent years. In this chapter, we review the state-of-the-art ML techniques for the cloud data center management problem. We examine the challenges and the issues in current research focused on ML for cloud management and explore strategies for addressing these issues. We also discuss advantages and disadvantages of ML techniques presented in the recent literature and make recommendations for future research directions.

[3] The research presented in this chapter is published in [170].

4.1. MOTIVATION AND CONTRIBUTION

In October 2021, 4.88 billion people across the globe were connected to the internet. That is nearly 62% of the global population, and this number is growing at a rate of 4.8% per year [115]. Most people use smartphones to connect to the Internet, but the use of portable computers, wearable devices, smart home and security products, Internet-of-Things (IoT) devices, etc., is continuously increasing. To handle the resulting massive volumes of internet data, companies of all sizes are looking for ways to achieve reliable and secure storage and data processing, with easy administration of internal and external data. This can be achieved with the help of cloud computing. Cloud computing is the delivery of virtual resources (servers, storage, apps, online services, etc.) over the Internet by centralized systems situated far away from end users. For simplicity in this chapter, we refer to all hardware resources as "resources." The trend of moving computation, data storage, and applications into cloud data centers has facilitated ubiquitous access to shared computing and storage resources on demand. Among their many benefits, cloud computing services have helped improve IT workflows, reduce costs by removing on-premises servers, increase scalability, ease maintenance, improve deployment speed, lessen downtime, enhance security, and facilitate mobility and flexibility of work practices.

The ever-growing use of the internet, data processing, storage, and advancement in modern technology has led to an increase in the use of cloud computing over time. The worldwide cloud computing market is projected to grow from $445.3 billion in 2021 to $947.3 billion by 2026 [116]. Because of the COVID-19 pandemic, digital business transformation has become more difficult and urgent. As a result of the pandemic's economic impact, major companies are providing clients with cost-effective digital solutions. The need for cloud services has surged as a result of the sudden closures of offices, schools, and businesses due to the pandemic. By 2025, Gartner [117]

predicts that over 85% of businesses will have adopted the cloud-first strategy, with over 95% of new workloads being deployed on cloud-native platforms (up from 30% in 2021). Cloud revenue will overtake non-cloud income in IT markets during the next few years.

4.1.1. NEED FOR GEO-DISTRIBUTED CLOUD DATA CENTERS

Cloud service providers are increasingly developing additional data centers to fuel the cloud computing boom, bolstering the capabilities of their cloud computing offerings. While centralized data centers were once commonplace, there has been a trend in recent years toward dispersing data centers throughout different geographical regions [118], [119]. There are many advantages to distribute data centers globally. It brings businesses closer to their clients by improving performance (e.g., latency) and lowering network expenses. Due to the redundancy that distributed data centers enable, it also provides superior resilience to unanticipated failures (e.g., environmental disasters). It enables businesses to handle data with various criteria as required by a specific country/region, industry, or workflow. It also aids in the compliance of regional data storage and privacy laws because some data or information is not allowed to be disseminated or processed outside of a specific geographic region. Corporations can utilize the distributed cloud to enforce restrictions while still adhering to data privacy regulations. Another compelling reason to geographically distribute data centers is to reduce energy costs by taking advantage of electricity pricing and renewable energy differences across different regions. Because of the aforementioned reasons and the increasing trend of distributing data centers globally, in this chapter, we review research efforts and recent developments that focus on “geo-distributed” cloud data centers.

4.1.2. CHALLENGES WITH GEO-DISTRIBUTED CLOUD SERVICES

Key cloud service providers such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud, and Alibaba Cloud run massive data centers around the world. All of these data centers are packed with high-core-count CPUs, terabytes of RAM, and petabytes of storage. Bloomberg's latest report [120] shows that data centers currently account for roughly 1% of worldwide electricity usage, and that percentage is expected to climb to 3 - 8% in the next decade. The amount of energy used by data centers around the world has doubled in the last decade, and some studies predict that it may triple or even quadruple in the next decade [83]. The annual electricity expenditure for powering data centers is continually increasing, so minimizing such expenditures is critical. As an example, by 2023, China's data centers are expected to consume more energy than the entire country of Australia [121]. Such annual energy expenditures sometimes exceed the cost of purchasing data center equipment. Thus, determining how to manage cloud computing components in a cost-effective, carbon/energy-saving, and balanced manner has become a significant focus of researchers working in the cloud computing area.

Another challenge is in providing timely access to cloud services, without incurring SLA and QoS violations. A Service-level Agreement (SLA) is a contract between a service provider and a client that defines the scope of the service, e.g., service quality, availability, and vendor responsibilities. From defining services to the end of the agreement, SLAs normally have a lot of components and are also defined at many levels. Quality of service (QoS) is the description of a service’s overall performance, such as a computer network or a cloud computing service, as observed by its users. Many network service components such as goodput, availability, latency, transmission delay, out-of-order delivery, errors, and packet loss, are frequently used to assess QoS for cloud services. High-volume data collection and processing in cloud data centers often

leads to high latencies and network bandwidth use. According to an Amazon analysis, a 100ms increase in web page loading time cost the company 1% of its sales. Google discovered that adding 0.5 seconds to the time it takes to generate a search query result reduced traffic to its site by 20% [122]. For interactive and time-critical applications, lower latencies can greatly improve the quality of the experience. This highlights the importance of preventing SLA and QoS violations.

Cloud management approaches are often employed to reduce the energy consumption, carbon emissions, network latency, query completion time, data center operating costs, and QoS and SLA violations. However, orchestrating the sophisticated, heterogeneous, large-scale software and hardware systems in complex network settings, and automating computing platforms involved in geo-distributed cloud computing systems, is a very difficult problem. Researchers in this field have been employing heuristics and algorithms for sub-problems such as low overhead virtual machine (VM) migration, queuing minimization, latency-aware service/workload migration, and performance/energy optimal resource allocation. Current mathematical methods to resolve such problems employ many strategies, such as heuristic methods, meta-heuristic algorithms, probability algorithms, hybrid algorithms, and dynamic programming methods. These mathematical methods have limited relevance in large scale dynamic HPC systems and are difficult to scale to geo-distributed levels. As a result, cloud service providers have been obliged to investigate intelligent data-driven and Machine Learning based alternatives.

4.2. SURVEY ORGANIZATION

The purpose of this article is to discuss the applications of ML in geo-distributed cloud data center management, as well as the challenges that remain unsolved, to highlight opportunities for future work in this area. We do not limit our scope to a specific context, scenario, or eco-system

in this regard, but rather focus on a variety of techniques proposed to address geo-distributed data center management; and discuss how these techniques are influencing the future generation of cloud computing architectures. Such architectures are highly heterogeneous and complex, to the point that a completely engineered approach to modeling and prediction could be needlessly complicated, if not outright futile. In such environments, cloud optimization plays a very important role. The process of correctly identifying and assigning the right resources to workloads (tasks or applications) or vice versa is known as cloud optimization. We specifically focus on techniques that use ML for prediction, classification, profiling of resources and workloads, application/task placement, migration, cloud optimization, and hybrid ML methods across the spectrum of geo-distributed computing scenarios. Accordingly, this article deconstructs and analyzes the challenge of reliable geo-distributed cloud data center management before surveying ML based approaches that solve this problem (or parts of it). The contributions of this article can be summarized as follows:

- we decompose the geo-distributed cloud data center management problem into various sub-problems;
- our study comprehensively reviews state-of-the-art ML-based techniques for workload and resource profiling, parameter prediction, and cloud optimization;
- we discuss challenges and future directions for ML in geo-distributed cloud data center management.

4.3. OVERVIEW OF MACHINE LEARNING METHODS

Machine Learning (ML) refers to a machine's ability to learn from data. Without being explicitly programmed, ML allows computers to learn how to perform tasks, such as predicting outcomes and classifying objects. One of its key assumptions is that by utilizing training data and statistical

approaches, it is possible to develop algorithms that can anticipate future or previously unseen values. ML has slowly established itself as the de-facto solution in a variety of enterprise applications, including speech recognition, self-driving cars, business intelligence, web search, fraud detection, purchase recommendations, and customer service, to name only a few examples. The availability of enormous datasets and continued advances in both ML theory and the computational capabilities of servers are largely responsible for this accomplishment. In recent years, big tech companies have made considerable investments in ML and Artificial Intelligence (AI) by introducing new services and undertaking major reorganizations to strategically place AI in their organizational structures [123].

There are three broad ML paradigms in widespread use today: supervised learning, unsupervised learning, and Reinforcement Learning (RL). The purpose of supervised learning is to find a function that best approximates the relationship between inputs and outputs, given a sample of input data and desired outputs. Unsupervised learning, on the other hand, has no labeled outputs and instead seeks to infer the inherent structure existing in a set of input data points. RL is an ML algorithm that rewards desirable behaviors while penalizing undesirable ones. We briefly introduce the ML algorithms included in this survey in the following sub-sections.

4.3.1. CLUSTERING

Clustering is the process of dividing a set of data points into groups so that data points in the same group are more like each other and different than the data points in other groups. Clustering methods are unsupervised ML algorithms. The goal of *k*-means clustering, a widely used clustering algorithm, is to divide *n* data points into *k* clusters, with each data point belonging to the cluster with the closest mean (cluster centroid or cluster center), which serves as the cluster's prototype. Spectral clustering is another method that has its origins in graph theory and is used to discover

clusters of nodes in a graph based on the edges that connect them. Figure 26 shows an example of clustering process in ML.

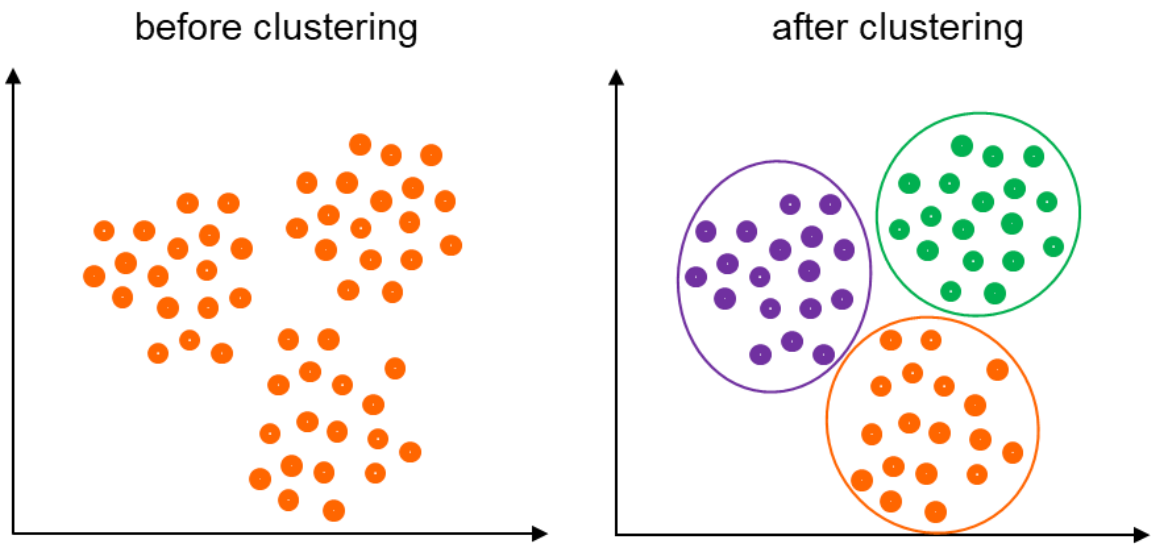


Figure 26: Clustering process in ML

4.3.2. REGRESSION

A problem is called regression when the purpose is to forecast a continuous or quantitative output value. Regression is a subfield of supervised ML. A linear model, such as one that predicts a linear relationship between the input variables *x* and a single output variable *y*, is known as *Linear Regression (LR)*. The link between the input variable *x* and the output variable *y* is treated as an *n*th degree polynomial in *x* in *Polynomial LR (Poly LR)*. The *Least Absolute Shrinkage and Selection Operator (LASSO)* is a regression analysis approach that does both variable selection and regularization to improve the statistical model's prediction accuracy and interpretability. In *Decision Trees (DTs)*, the goal is to learn simple decision rules from data attributes to develop a model that predicts the value of a target variable. *Random forest (RF)* algorithms use ensemble learning method for regression. The ensemble learning method combines predictions from several ML algorithms to get a more accurate forecast than a single model. RF regression works by training

a large number of DTs. Gradient boosting is a regression technique that generates a prediction model from an ensemble of weak prediction models, most commonly DTs. The resulting technique is called *Gradient-Boosted Regression Trees (GBRTs)* when a DT is the weak learner. *eXtreme Gradient Boosting (XGBoost)* is an efficient and popular open-source implementation of the gradient boosted trees. The engineering goal of XGBoost is to push the computational resources' limits for boosted tree algorithms. The *k-Nearest Neighbors (k-NN)* regression technique approximates the relationship between independent variables and continuous outcomes by averaging observations from the same neighborhood. *Support Vector Machine (SVM)* regression is a supervised learning method for predicting discrete values. SVM regression's key concept is to identify the best-fit line. The hyperplane with the most values is the best fit line in this case. *Auto-Regressive Integrated Moving Average (ARIMA)* is a generalization of the simpler Auto-Regressive Moving Average (ARMA) and adds the notion of integration. Both models are used to fit time series data to better understand or anticipate future points in the series. *Seasonal ARIMA (SARIMA)* is an ARIMA variant that supports univariate time series data with an explicit seasonal component. Figure 27 shows an example of regression process in ML.

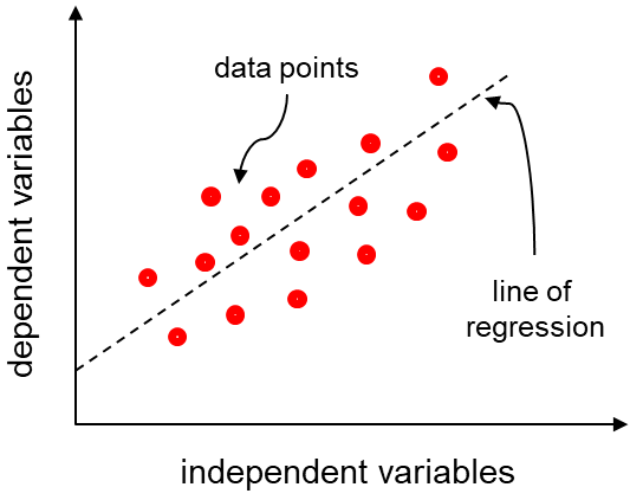


Figure 27: Regression process in ML

4.3.3. NEURAL NETWORKS

Neural networks (NNs), also known as artificial NNs (ANNs), are a class of ML algorithms whose name and structure are derived from the human brain. These networks resemble networks of biological neurons and mimic the manner in which they communicate with one another. NNs are made up of node layers, consisting of an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, is connected to the others and has a weight and threshold associated with it. The node is activated if its output, which is frequently generated using a mix of input data and its associated weight, exceeds a particular threshold value. This "activation" refers to the method by which the node propagates the information farther down the network, typically following the application of a specific activation function that adds non-linearity to the learning process. If the output does not exceed the threshold, no data is forwarded to the next layer of the network. To learn and increase their accuracy over time, NNs depend on training data. However, once these learning algorithms have been fine-tuned for accuracy, they become strong tools in area of computer science and AI, allowing us to rapidly cluster, classify, or predict data. When compared to manual identification by human analysts, tasks in the domains of voice or image recognition can take seconds with NNs rather than hours.

Stochastic configuration networks (SCNs) use a supervised learning mechanism to automatically and rapidly construct universal approximators that achieve promising performance for solving regression problems. A *Multi-Layer Perceptron (MLP)* is a type of feedforward NN. A perceptron is a model of a single neuron that served as a predecessor to larger NNs. *Deep Learning (DL)* is a subset of ML, which is essentially a NN that has three or more layers. While a single-layer NN may produce approximate predictions, extra hidden layers can help to optimize and enhance accuracy. A *Convolutional Neural Network (ConvNet/CNN)* is a DL model that can

take an input image, assign relevance (learnable weights and biases) to numerous objects in the image, and distinguish one from the other. A *Recurrent Neural Network (RNN)* is an NN that works with time series or sequential data. RNNs have hidden states and allow previous outputs to be utilized as inputs. Language translation, image captioning, natural language processing (NLP), and speech recognition are just a few of the challenges that these RNNs are widely utilized for. *Long short-term memory (LSTM)* is a type of RNN with a more complex neuron structure. LSTMs can handle predictions with individual data inputs (such as images) as well as long data sequences (such as speech or video). LSTM cells uses a gating mechanism with three gates: input, output, and forget. *Gated recurrent units (GRUs)* are similar to LSTMs but have only two gates: reset and update. Because GRUs have fewer gates than LSTMs, they are less complicated. If the dataset is small, GRUs are generally preferable; otherwise, for larger datasets, LSTMs often demonstrate better performance. Figure 28 shows examples of neural networks in ML.

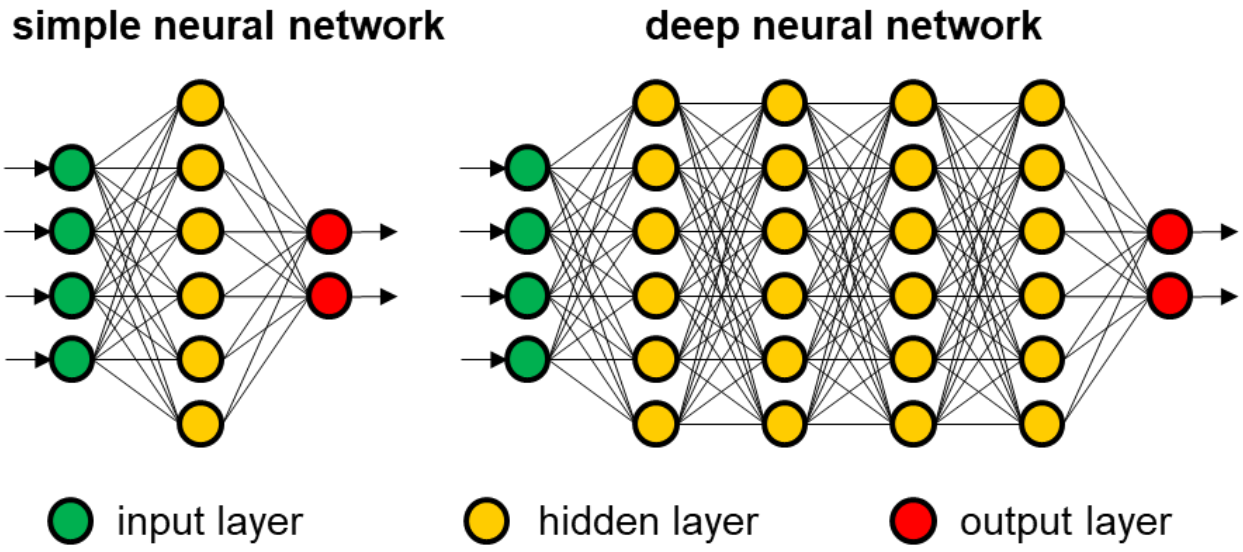


Figure 28: Neural networks in ML

4.3.4. REINFORCEMENT LEARNING

Reinforcement learning (RL) is an ML technique that allows an agent to learn by trial and error in an interactive environment using feedback from its own actions and experiences. The *agent* represents the RL algorithm, while the *environment* refers to the item on which the agent is acting. The environment sends a state to the agent, which then takes *action* in response to that condition based on its knowledge. The environment then sends the agent a pair of next state and reward information. To evaluate its latest action, the agent will update its knowledge using the reward returned by the environment. The cycle continues until the environment sends a terminal state, bringing the episode to an end. Below are the few important terms used in RL.

- *Action:* All the moves available to the agent.
- *State:* Current state returned by the environment.
- *Reward:* An immediate return (feedback) from the environment to assess the quality of the previous action.
- *Policy:* The agent's strategy for determining the next course of action based on the current state.
- *Value:* The predicted long-term return with discount, in contrast to the short-term reward.

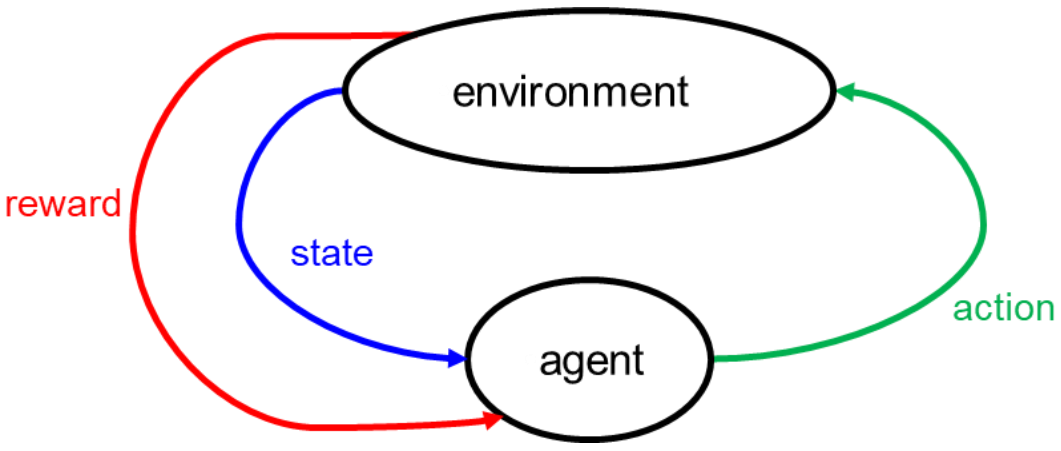


Figure 29: Reinforcement learning

- *Q-value* or *action-value:* The long-term return of the current state, taking an action, under a policy.
- *Model:* The agent's perspective on the environment, which converts state-action pairings into probability distributions over states.

It is worth noting that not every RL agent makes use of a model of its surroundings. *Model-based* RL tries to model the environment and then determine the best policy based on the model it has learned. On the other hand, in *model-free RL*, the agent uses trial and error experience to output the optimal policy. Policy optimization and Q-learning are the two basic methodologies for representing agents using model-free reinforcement learning. In *policy optimization* or policy-iteration methods, the policy function that maps states to actions is directly learned by the agent. A value function is not used to decide the policy. In *Policy Gradient (PG)*, gradient descent is used to optimize parametrized policies with respect to the expected return (long-term cumulative reward). *Actor-critic* is another policy optimization method where the critic estimates the action-value (Q-value) or state-value function and the actor updates the policy distribution in the direction suggested by the critic. The *Q-learning* algorithm attempts to determine the best action given the current situation using a lookup table called Q-table, where the maximum expected reward for actions at each state are stored. *Deep Q Neural Network (DQN)* combines Q-learning with NNs. Instead of building Q-table, NNs approximate Q-values for each action based on the state. *Deep Deterministic Policy Gradient (DDPG)* is an algorithm that learns both a Q-function and a policy at the same time. It learns the Q-function with off-policy data and then utilizes the Q-function to learn the policy. *Deep reinforcement learning (DRL)* is a combination of reinforcement learning and deep learning. Deep learning is included into DRL, allowing agents to make decisions based on unstructured input data without having to manually build the state space. DRL algorithms can

process enormous amounts of input and determine what actions to take to optimize an objective. The use of RL in a multi-agent system is known as *Multi-Agent Reinforcement Learning (MARL)*. Typically, agents develop their decisions as a result of their previous experiences. Here, an agent, in particular, must understand how to coordinate with other agents similar to game theory. Figure 29 shows an example of reinforcement learning.

4.4. RELATED SURVEYS

4.4.1. CLOUD MANAGEMENT

There are many previous review articles that have explored resource management, workload management, security, VM allocation, energy efficiency, and load balancing for cloud computing. Some recent reviews [124], [125], [126] presented a multi-dimensional classification of existing cloud resource management solutions based on their scheduling architectures, objectives, and methods. In [127], various clustering, optimization, and ML methods used in cloud resource management to improve energy efficiency and performance are compared, classified, and analyzed. In [128], [129], and [130] an overview is provided for existing cloud computing load balancing algorithms and their performance metrics. These works also discussed the advantages and disadvantages of the chosen load balancing algorithms. In [131] a study of various available resource allocation methods, load balancing techniques, scheduling techniques and admission control techniques for cloud computing is presented along with an analysis of the advantages and disadvantages of each method. Based on our analysis of these surveys, we observed that some reviews considered a few ML based studies, but they do not comprehensively review ML based cloud management that presents the importance of the ML techniques in cloud computing.

4.4.2. MACHINE LEARNING BASED CLOUD MANAGEMENT

Several recent surveys have considered ML based cloud management. Among these, [132] divided the cloud resource scheduling problem into various objectives, such as time optimization, energy consumption optimization, and load balancing optimization. The authors discussed and researched the architecture of RL and Deep Reinforcement Learning (DRL). The review progressively evaluated various architectures of RL and DRL, offering a unified view for the subsequent RL or DRL in cloud computing resource scheduling. In [133], the authors looked at how the topic of reliable resource provisioning in joint edge-cloud systems has been addressed in the scientific literature, and what strategies have been utilized to increase the reliability of distributed applications in diverse and heterogeneous network environments. The survey discussed how the characterization, management, and control of complex distributed systems utilizing ML methodologies are given significant attention due to the problem's complexity. The survey is organized around a three-part deconstruction of the resource provisioning problem: workload characterization and prediction, component placement, system consolidation, and application elasticity and remediation. The authors in [134] divided cloud resource management into four major categories: resource provisioning, resource scheduling, resource allocation, and energy efficiency. After presenting a comparison between works in these categories, they suggested the most suitable ML model for each category. In [135], a broad survey of works that leverage ML techniques for resource management in cloud computing is presented. Unlike other works, this review categorized and analyzed the included studies on the basis of three types of ML models: supervised learning, unsupervised learning, and RL. This review further evaluated the ML techniques by the type of features they use, the outputs they produce, and the objectives they try to optimize. In [136] a detailed review of challenges in ML-based cloud computing resource

management is presented. The survey also discussed state-of-the-art approaches to resolve these challenges, along with their pros and cons. Finally, the authors proposed possible future research directions based on identified limitations.

4.4.3. MACHINE LEARNING BASED GEO-DISTRIBUTED CLOUD DATA CENTER MANAGEMENT

While it is very useful to employ ML in distributed cloud computing systems and to assess the progress of the recent research and engineering of these systems, none of the recent articles go into detail about ML-based cloud management for geo-distributed data centers, nor do they go into the specifics of the challenges and issues that exist in the current state-of-the-art and future research directions on this theme. As discussed in our previous work [137], geo-distributed cloud data centers can exploit differences between electricity prices, net metering, peak demand pricing distribution, resource availability, data transfer/network prices, co-location in servers, and renewable energy at various data center locations. Additionally, geo-distributed data centers give higher resilience to unplanned failures due to their redundancy. As a result, we believe that it is the right time to now present a comprehensive survey that discusses various ML algorithms and their applications to different problem contexts within cloud management scenarios for geo-distributed data centers, as well as their shortcomings, challenges, and future directions, in accordance with our vision. Thus, before moving further with their new ideas in this regard, researchers can use this article to assess the current ML scenarios in geo-distributed cloud management and their limitations.

4.5. COMPONENTS OF CLOUD MANAGEMENT

Reliable cloud management for data centers is a complex problem, especially when it is examined in a geo-distributed cloud environment where distributed infrastructure is utilized to host numerous heterogeneous workloads coming from different geographical regions. Each workload typically has its own set of requirements, and there is a high possibility that controlling operations or tuning the performance of one workload that is using one or multiple resources will have some effect on the others e.g., workload co-location interference effects [107]. Additionally, the problem involves deployment and operation with workloads in geographically dispersed heterogeneous resource environments. In such complex environments, solving the problem completely and optimally is a difficult task, and viable solutions must involve a combination of various methods to achieve the predictability and controllability of both the (software) workload and (hardware) resource performance. This necessitates a thorough investigation of all parts of the problem to develop a holistic understanding of the problem and the challenges while developing a solution. For this purpose, as shown in Figure 30, we divide and discuss aspects of geo-distributed cloud management across three categories: profiling/clustering, parameter prediction, and cloud optimization.

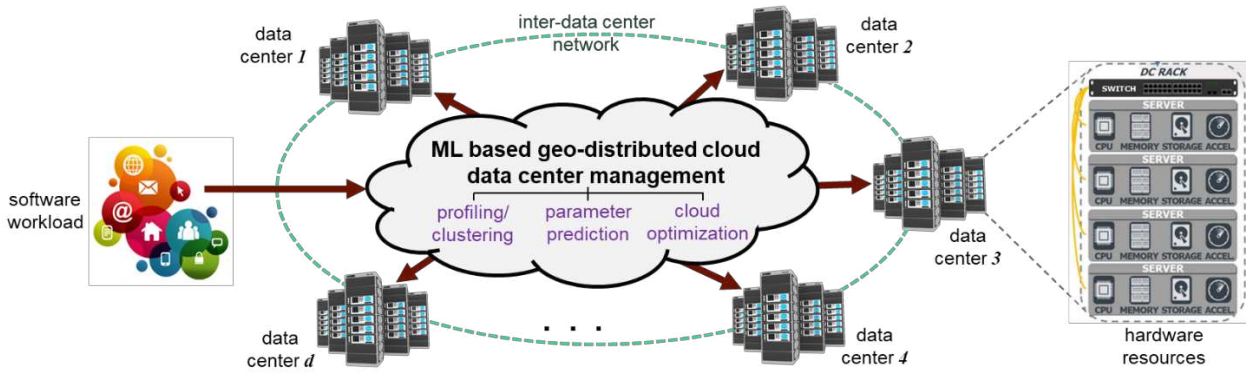


Figure 30: Machine Learning (ML) based geo-distributed cloud data center management

4.5.1. WORKLOAD AND RESOURCE PROFILING

Cloud workloads can be classified into different types based on various characteristics and perspectives, such as transactional or non-transactional, sequential or non-sequential/random, and memory intensity. In terms of both resource type and volume, the need for resources varies depending on the workload. A cloud application workload's components are designed to carry out certain activities that are likely to use various types of resources (e.g., CPUs or GPUs). Moreover, because of the wide geographic distribution of data centers over networks, workloads can be distributed in different ways across different data center locations. These characteristics lead to high complexity and challenges in workload and resource profiling. The ability to understand workload and resource behavior using ML appears to help improve an application's performance and reliability, by allowing cloud providers to ensure the availability of sufficient resources to support any future workloads. Therefore, many cloud management techniques require workload and resources to be profiled or clustered into a specific category, before the techniques can be effectively applied. Figure 31 shows an example of workload and resource profiling.

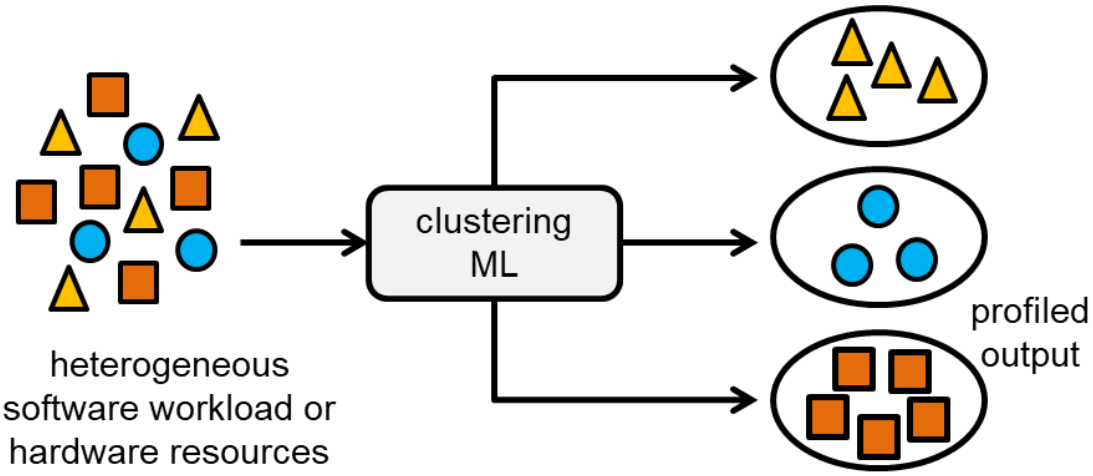


Figure 31: Workload and resource profiling

4.5.2. PARAMETER PREDICTION

In the age of pervasive edge and IoT computing, the workloads running across cloud environments are steadily increasing in both complexity and volume. The heterogeneity of workloads and the number of users using a given cloud application service at various times introduces changes in cloud workloads and the corresponding resources that the workloads need to use across data center locations. In other words, cloud applications experience variations of workload arrival and resource usage in a manner that can be difficult to predict. The ability to predict the spatio-temporal distribution of future workloads or their parameters in advance with ML brings benefits for geo-distributed cloud management solutions, i.e., the system gains the ability to scale proactively to satisfy real-time resource requirements. Or, in some cases, the workload can be allocated to resources effectively if future utilization states of the resources are known. In this way, it is possible to improve workload allocation and optimize resource utilization while guaranteeing QoS requirements and SLAs, which in turn enhances the efficacy and performance of geo-distributed cloud management. Figure 32 shows an example of parameter prediction.

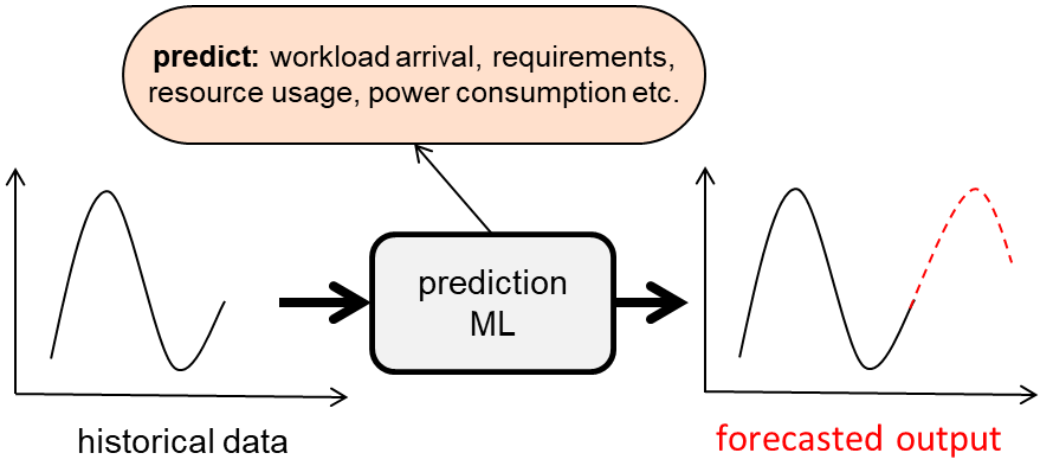


Figure 32: Parameter prediction

4.5.3. CLOUD OPTIMIZATION

Cloud optimization techniques are typically implemented using either static or dynamic algorithms. While the workload placement or the resource allocation rules are predetermined in a static scheme, in a dynamic scheme the workload is dynamically processed and allocated to the resources. Even though static techniques are simple and usually more stable, they do not achieve optimal results with heterogeneous workloads and distributed resources subject to erratic workload distribution and unpredictable resource behavior. The cloud optimization techniques allocate or release resources on the fly according to the demand of the workload, thus reducing the operating costs while maintaining QoS requirements and SLAs. Ideally, cloud resources and workloads need to be dynamically configurable, programmable, and optimizable with the help of ML, without any human input. An important component of cloud resource management involves Virtualized Network Functions (VNFs) that are created on open computing platforms and run virtual network services on common servers in cloud data centers. Virtualized routers, firewalls, WAN optimization, and network address translation (NAT) services are all examples of VNFs. Virtual machines (VMs) running on common virtualization infrastructure software such as VMWare or Kernel-based Virtual Machine (KVM) run the majority of VNFs. Network service providers must

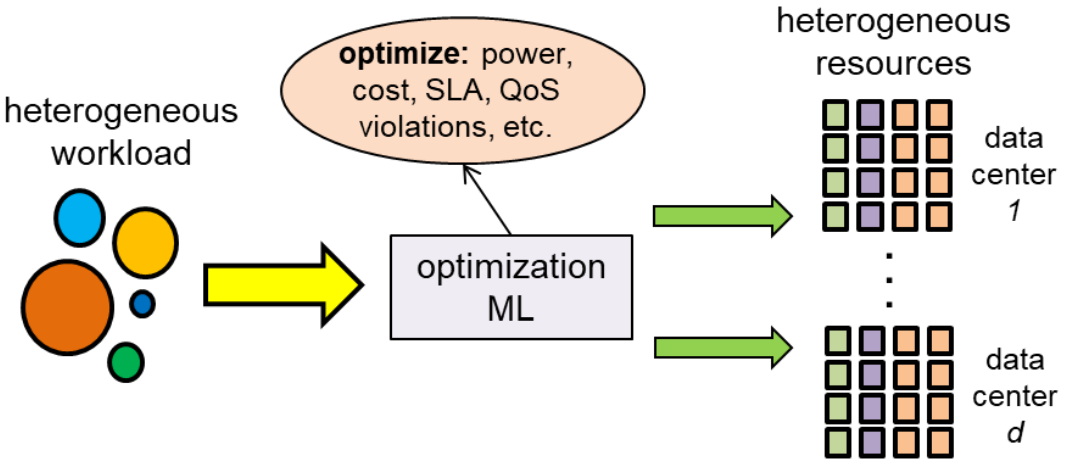


Figure 33: Cloud optimization

configure network traffic according to flow-level granularity, lowering network costs and ensuring a good user experience. As workload and resource demands vary over time in cloud environments, it is also important to dynamically scale the VNF instances, with the help of ML. Figure 33 shows an example of cloud optimization.

4.6. SURVEY OF RELEVANT WORKS

ML and cloud computing have become popular and widely adopted in industry. A recent trend has been to explore how ML may help with cloud management for geo-distributed data centers. In the past few years there has been a continuous increase in the numbers of studies in this area. For this survey, we selected 22 articles related to the ML based geo-distributed cloud data

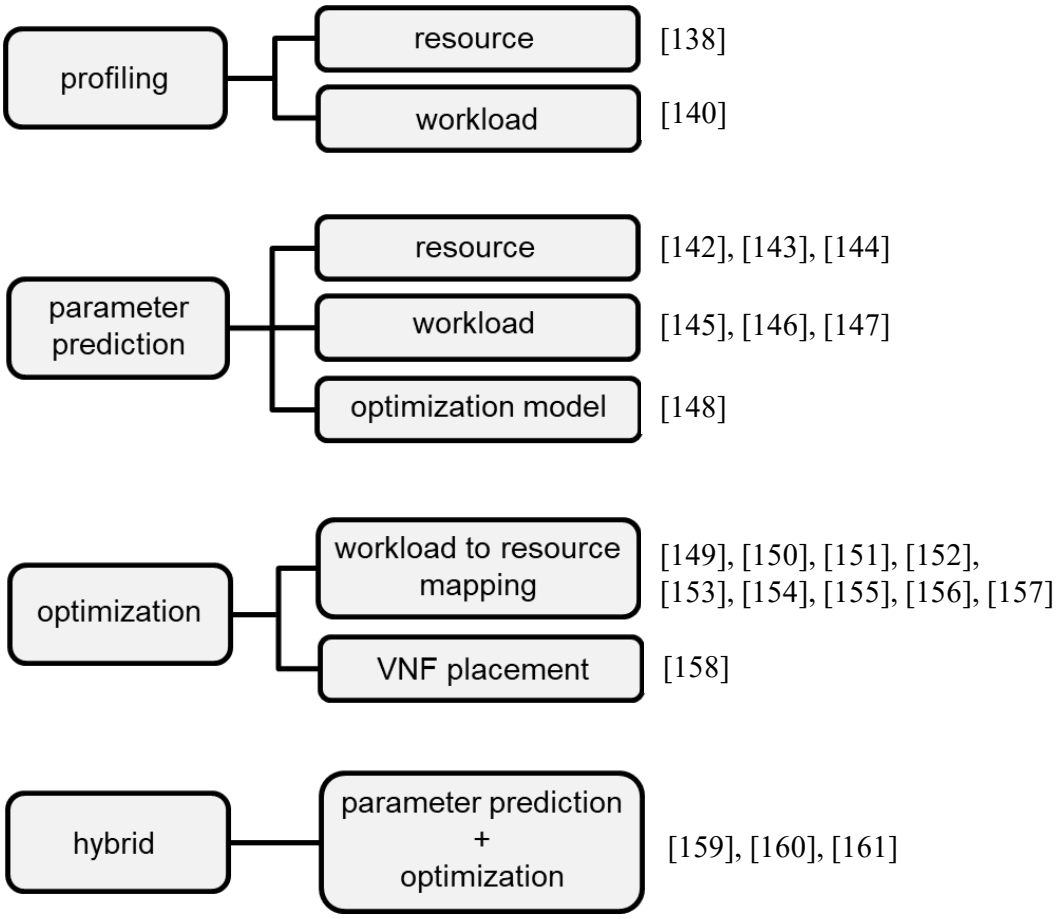


Figure 34: Taxonomic classification of the literature surveyed

center management. The articles have been classified into four main groups. A taxonomic classification of the literature surveyed in this review is shown in Figure 34. In the figure and the rest of the article, "resource" refers to hardware systems, whereas "workload" refers to software applications or tasks. The following subsections discuss the prior works in these four main groups in more detail.

In Table 9, we provide a comparative summary of the investigated papers based on the various ML approaches utilized in the geo-distributed cloud management environment. The 1st column describes the ML subcategory of geo-distributed cloud management problem that we discussed in Section 4.5. The 3rd column demonstrates why different ML models are used in various research articles and their main purpose. Column number 4 and 5 contains the information about the ML model and the dataset used in each paper. The 6th column lists the comparison and the proposed ML methods used in each study. The 7th column shows the quantitative/qualitative improvements for the proposed method over the comparison methods listed in column 6.

4.6.1. PROFILING

4.6.1.1. RESOURCE PROFILING

The goal of [138] is to select the optimal data center (resources) among a set of geo-distributed data centers based on SLA standards and the needs of the user. The SLA here is comprised of a set of service-level objects (SLOs), but in this study, the four most essential ones are cost, response time, availability, and reliability. After receiving the user's request in a geo-distributed cloud environment, firstly, the data are normalized using the Z score normalization method to remove abnormal distribution of various SLOs. Then the number of clusters are determined according to four SLA criteria, where 3844 data centers are clustered using the *k*-means algorithm. In the next step, based on the longitude and latitude of the user's location, the nearest

geographical cluster is selected. Furthermore, accessibility and reliability are maximized, while cost and response time are minimized, and the multi-objective NSGA-II [139] algorithm is used to select the best data center based on these SLOs.

4.6.1.2. WORKLOAD PROFILING

Classical approaches for uniformly partitioning data across distributed nodes are the norm in many major distributed data storage solutions, such as Cassandra and Hadoop Distributed File System (HDFS). These methods can generate network congestion for data-intensive services, lowering system bandwidth. Data-intensive services, in contrast to MapReduce workloads, require access to many datasets throughout each session. To address these issues, the authors in [140] proposed a scalable method for distributing data-intensive services across geographically distributed cloud data centers. The authors first constructed a hypergraph using a given set of data items, and the set of user check-ins, each comprising a data request pattern being obtained from a data center location. Hypergraphs allow for the capturing of multi-way relationships, making it easier to model data item to data item, and data item to data center location relationships. The dataset used in the experiments is a trace of a location-based online social network – Gowalla, available publicly from the Stanford Network Analysis Platform (SNAP) repository [141]. Using spectral clustering on hypergraphs, the proposed approach divided a set of data items into geo-distributed cloud data center locations to minimize the weighted average of the per unit cost of outgoing traffic from each data center, the inter data center latency for each data center pair, the per unit storage cost of each data center, and the average number of data centers accessed by the data items requested in each request pattern. Furthermore, the spectral clustering algorithm employed randomized techniques to create low-rank approximations of the hypergraph matrix, allowing for greater scalability in the computation of the hypergraph Laplacian spectra.

Table 9: Summary of ML Techniques Used in Geo-distributed Cloud Data Center Management

ML Type	Ref	ML Functionality	ML Model	Dataset	Comparison Methods and Proposed Method	Main Results
resource profiling	[138]	cluster data centers according to four SLAs - cost, response time, availability, and reliability	*k*-means clustering	1998 FIFA World Cup Website traffic	Random, Greedy, and Multi-Objective Particle Swarm Optimization (MOPSO), and NSGAII_Cluster	selected the best data center according to SLA rules and the user's request
workload profiling	[140]	partition a set of data-items into geo-distributed clouds	spectral clustering	Gowalla dataset	Random, Nearest and hypergraph-based partitioning (Hyper), and Spectral Clustering (Spectral)	performed up to 10 times (3–4 times on average) faster
resource prediction	[142]	predict data center incoming workload locally and power consumption globally	SARIMA & CNN	Google cluster data	Capacity aware, Cost-aware, Power-aware, Power-Aware based RS participation (PARS), and Cost-Aware based RS participation (CARS)	reduced the energy cost by 22% on average
	[143]	predict the power pattern	LR & NN	Google cluster data	actual test data set	power prediction model reached the accuracy of 87.2%
	[144]	predict the approximate transcoding resources to be reserved	GRU, LSTM, MLP, CNN, and XGBoost	Facebook live videos dataset	Greedy Minimal Cost (GMC), Greedy Cheapest Algorithm (GCA), and Greedy Nearest Cheapest Algorithm (GNCA)	achieved more than 20% gain in terms of system cost
workload prediction	[145]	accurately estimate the time cost of query execution plans	LASSO Regression and GBRT	TPC-H benchmark	default Spark SQL optimizer, Clarinet, Turbo - Shortest Completion Time First (SCTF), Turbo - Maximum Data Reduction First (MDRF), Turbo - Maximum Data Reduction Rate First (MDRRF)	achieved cost estimation accuracy of over 95% and reduce query completion times by 41%
	[146]	predict the workload in future time series	SGW-SCN	Google cluster data	ARIMA, BPNN, SCN and their variants with either Savitzky-Golay filter (SG), wavelet decomposition (W), or both-SGW, and SGW-SCN	produced more accurate results and with faster learning speed
	[147]	predict delay, jitter, and throughput of each network link	LR, Poly LR, DT, RF, SVM, MLP	Synthetic	no comparison	chose the best data center to restore the services of a faulty one
model parameter prediction	[148]	predict optimization model weights	*k*-NN	Planetlab traces, Synthetic	Simple fit, first fit, best fit, and best fit - SLA versions of Power and Cost-aware Virtual Machine Placement (PCVM) models, and PCVM Normalized Weight Prediction (PCVM-NWP)	improved cloud provider net profit by reducing DCs power consumption
workload to resource mapping optimization	[149]	balance QoS revenue and power consumption	RL (Q-learning)	CoMon project	Dynamic Voltage and Frequency Scaling (DVFS), static THReshold (THR), Median Absolute Deviation (MAD), InterQuartile range Regression (IQR) and Local Regression (LR) algorithms, and RL based adaptive algorithm	power consumption is up to 13.3% better
	[150]	search for the shortest path while minimizing network costs	RL (Q-learning)	Synthetic	no comparison	solved the problem achieving low network and compute resource costs
	[151]	allocate resources to workloads while considering the green power availability and QoS requirements	RL (Q-learning)	Synthetic, Facebook-Derived (FaceD)	heterogeneity-oblivious approach GLB, and holistic workload and resource manager (sCloud)	improved 37% system goodput and saved 33% QoS violations

ML Type	Ref	ML Functionality	ML Model	Dataset	Comparison Methods and Proposed Method	Main Results
workload to resource mapping optimization	[152]	choose the right data center to allocate resources and serve potential viewers (for live streaming), while minimizing delays and operating costs	RL (DQN)	Facebook live videos dataset	Random Forest (RF), Decision Tree (DT) and MultiLayer Perceptron (MLP), Greedy Minimal Cost algorithm (GMC), and RL for Online and Proactive Resource Allocation (RL-OPRA)	outperformed comparison heuristics through continuous learning
	[153]	maximize operating profits	RL (Q-learning)	Google cluster data	Round-robin (RR), MinBrown (MB), and proposed RL based method	saved up to 50% more energy costs
	[154]	schedule HVAC control and workload for minimizing the total energy cost while meeting desired temperature and meeting workload deadlines	RL (DRL)	National Solar Radiation Database	Rule based baseline, Non distributed DRL, and heuristic DRL	achieved up to 26.9% energy cost reduction for a single location, and an additional 5.5% for all data centers combined
	[155]	minimize costs of big data analytics on data centers connected to unpredictable renewable energy sources	RL (DQN)	Google cluster data	Round robin (RR), MinBrown (MB), and Temporal Difference (TD), and Advanced Strategy Learning Algorithm (ASLA)	reduced the data centers' energy and data migration cost significantly
	[156]	control number of container replicas based on the application response time, and allocate containers as per network-aware placement policy	RL (model-based)	Synthetic	optimal ILP formulation (referred as OPT), Network-Aware greedy heuristic (NetAware), the greedy first-fit heuristic, round-robin heuristic, default policy included in Kubernetes (kube-scheduler); and Geo-distributed and Elastic deployment of containers in Kubernetes (ge-kube)	processed 2.91 times more operations per second
	[157]	joint energy consumption optimization of servers, and the network	RL (DQN)	Synthetic	Joint Optimization Control Algorithm (JOCA), Carbon-Aware Control Algorithm (COCA), CPHA (previous work), and Load-Balanced Control Algorithm (LBCA), Cost-Aware Capacity Provisioning (CACP), deep Reinforcement Learning (DRL)-based Control Algorithm (RLCA)	has high efficiency in cost-saving and performance-enhancing
VNF optimization	[158]	learn the policy of VNF deployment	DRL (DDPG)	Huawei web traffic	Pure DDPG algorithm (P-DDPG), DRL, Customized DDPG algorithm (C-DDPG), and DRL + Game Based Module (GBM) - hybrid DRL	average network savings are up to 29.8%
workload prediction & resource optimization	[159]	predict future flow rates; make VNF chain placement decisions	LSTM; DRL (actor-critic)	Huawei web traffic	ILP based Online Algorithm (OA), Greedy Algorithm (GA), Deep Q-Learning (DQN), and LSTM+DRL - actor-critic	reduced total system cost by up to 42%
	[160]	predict request latency and split across data centers; then use storage planning to reduce monetary cost	SARIMA & RL; RL	Wikipedia traces	CosTLO, TAILCUTTER (TC), Prop, SPANStore (SPS), SEDPT, and Geo-distributed cloud storage system with low Cost and latency using RL (GeoCol)	achieved 32% monetary cost reduction and 51% data object request latency reduction
resource prediction & workload optimization	[161]	predict the amount renewable energy generation and its demand at each data center; then determine how much renewable energy to request from each generator	SARIMA ; MARL	Wikipedia traces	Green Scheduling (GS), Renewable Energy Management (REM), Renewable Energy-Aware reinforcement learning (REA), Single Reinforcement Learning (SRL), Multi-Agent Reinforcement Learning (MARL) without Deadline-guaranteed job postponement (MARLw/oD), MARL with Deadline-guaranteed job postponement (MARL)	increased up to 35% SLO satisfaction ratio, and reduced up to 19% (0.33 billion dollars in 90 days) total monetary cost and 33% total carbon emission

4.6.2. PREDICTION

4.6.2.1. RESOURCE PREDICTION

In a Geographical Load Balancing (GLB) policy, the amount of load delivered to each local data center is determined by the overall system's conditions rather than the local situation. Regulation Service (RS) is a type of service provided by the power market to continuously balance supply and demand at a very granular level while also offering economic benefits to consumers and the network/utility. Optimizing RS for local data centers in a geo-distributed cloud is a difficult problem to solve. Motivated by this phenomenon, [142] used SARIMA to predict the incoming workload at a global level. Then the authors used a CNN-based ML model to forecast/output the power consumption of local data centers. Inputs to the models are the data center's outdoor temperature, solar power, electricity price, and workload arrival rate. The model used Rectified Linear Unit (ReLU) as the activation function, Adam optimizer, and Root Mean Squared Error (RMSE) as an evaluation metric. To develop a prediction model, this research used two main GLB policies, namely Power-aware and Cost-aware. The goals of these policies are to minimize the total power consumption and the total cloud electricity cost, respectively, summed across all geo-distributed data centers. Following that, this study used prediction results to offer geo-distributed data centers the possibility to participate in RS.

It is very difficult to predict power consumption of a data center that is part of a group of geo-distributed cloud data centers, especially with the intermittent availability of renewable energy and the use of free cooling devices like air economizers in data centers. The relationship between geo-distributed cloud data center power patterns and the weather characteristics (based on various conditions and infrastructure) is investigated in [143], and a set of influential features such as humidity, pressure, temperature, cloudiness, and wind speed, are extracted using linear regression

(LR). The acquired features are then used to create an NN-based power consumption prediction model that predicts the power consumption pattern of each participating data center in a cloud.

The work in [144] presented an ML-based predictive resource allocation methodology for geo-distributed cloud data centers, which considered latency and quality constraints to ensure the best possible QoS for viewers and the lowest possible cost for live streaming content providers. To begin, this work proposed an offline optimization that determined the required transcoding resources in distributed locations near the viewers, balancing QoS and total cost. Next, ML is used to create forecasting models that anticipate the approximate transcoding resources that will be reserved at each cloud data center in advance. There are five different algorithms used namely, LSTM, GRU, CNN, MLP and XGBoost. The mean absolute error (MAE) metric is used to assess these regression models. Because there is no way to predict which combination of hyperparameters, such as the number of hidden layers and neurons for LSTM, GRU, CNN, and MLP models, or the number of estimators for XGBoost models, is optimal, the authors generated numerous models for each ML approach. The best performing models were chosen based on the best determination coefficient value, which is used to evaluate the regression models' goodness of fit. Finally, the authors created a Greedy Nearest and Cheapest Algorithm (GNCA) for allocating live incoming videos to the forecasted transcoding resources. The authors used Facebook video metadata as an input and considered 10 AWS cloud instances as a geo-distributed cloud platform for their analysis.

4.6.2.2. WORKLOAD PREDICTION

Because it is built for a single data center, the existing data analytics software stack is insensitive to on-the-fly resource variations over inter-data center networks, which can have a severe impact on query performance. To address this issue, [145] proposed Turbo, a light and non-intrusive data-driven

framework that dynamically adapts query execution plans for geo-distributed data analytics in response to runtime resource fluctuations across data centers. Turbo uses two ML regression models, GBRT and LASSO, to precisely predict the time cost of query execution plans. The authors created a new dataset with 15K samples, each recording the time it took to run a query from the TPC-H benchmark [162], the size of the output, and a number of other parameters relevant to query execution. They set up a 33-instance cluster on the Google Cloud Compute Engine in eight different regions. Turbo is non-intrusive which means it does not necessitate any changes to the existing data analytics software stack.

Accurate workload forecasting is critical for dynamic resource scaling in geo-distributed cloud data centers. In [146], the authors proposed Savitzky-Golay and Wavelet-supported Stochastic Configuration Networks (SGW-SCN), an integrated forecasting approach with noise filtering and data frequency representation for predicting the workload in future time slots. The workload traces in Google production compute clusters from May 2011 were studied in this work. Over the course of 29 days, traces from a 12.5k-machine cell were collected, yielding a total of 25,462,157 tasks. Google clusters organize all tasks into multiple tiers, and each task has a characteristic reflecting its relevance. A higher level indicates that the task is more critical. Twelve attributes were investigated, which were divided into three types: gratis (0–1), other (2–8), and production (9–11). After that, the authors divided 29 days into 8352 five-minute time slots and counted the aggregated arrival rates of three different types of tasks. They used the data from the first 25 days for training and the last 4 days for testing. The workload time series is smoothed with a Savitzky-Golay filter before being split into several components using wavelet decomposition in this method. An integrated model is developed using SCN to characterize statistical properties of trends for different workload time series.

Data center availability is a crucial requirement in geo-distributed cloud environments. In this context, preventative efforts that reduce the time it takes to fix a data center's service in the event of a

breakdown are critical. The authors in [147] proposed a method for determining the influence of network performance on service availability. This study considered six data centers around the world and calculated delay and jitter using Round Trip Time (RTT) and the download transfer rate in Mbps. Using ML algorithms capable of forecasting the time to transfer a large amount of data based on delay and jitter, this work developed smart agents capable of selecting the optimal data center to restore the services of a malfunctioning one. The authors selected LR, Poly LR, DT, RF, SVM, and MLP regression algorithms. They developed ten distinct train/test splits, assessed their accuracy with the Mean Absolute Error (MAE) metric, and then averaged the results.

4.6.2.3. OPTIMIZATION MODEL PARAMETER PREDICTION

The authors of [148] investigated the topic of VM placement that reduces energy usage, CO_2 emissions, and access latency to reduce operating costs for geo-distributed cloud data center providers. This challenge is defined as a multi-objective function, with an intelligent ML model built to improve the presented Power and Cost-aware VM placement (PCVM) model's performance. The PCVM aims to reduce total costs by minimizing the weighted sum of two key sub-objectives: carbon emission and network communication costs. The PCVM optimization equation uses two constant normalized model parameter weights for weighting the two sub-objectives such that the sum of the weights is 1. While deciding among the two normalized model parameter weights associated with the two objectives, each one can be in conflict with one another. The authors applied a k-NN regression method to determine the optimal values for the model parameter weights in the equation so that they can help with finding an optimal solution to the VM placement problem.

4.6.3. OPTIMIZATION

We discuss prior work in the optimization category across two sub-categories: ‘workload to resource mapping’ and ‘VNF placement’. Workload to resource mapping optimization refers to assigning incoming workloads (tasks or applications) to the best hardware resources across data centers. VNF placement optimization refers to dynamically placing VNFs on appropriate hardware to achieve an objective.

4.6.3.1. WORKLOAD TO RESOURCE MAPPING OPTIMIZATION

The authors of [149] proposed an adaptive resource management technique based on RL, with the goal of achieving a balance between QoS revenue and power usage across a set of geo-distributed cloud data centers. The proposed Q-learning based RL technique with custom random action selection is robust in allocating workload and does not require prior distribution of resource requirements. Instead of employing SLA violations, this work explicitly modeled QoS revenue using differential revenue of separate jobs. The authors considered the time spent migrating VMs across data centers and the cost of network communications. This work achieved quick decision-making by fine-tuning the RL algorithms' information storage and random action selection. The proposed RL based technique aims to balance QoS revenue and power consumption by performing inter data center VM migration and dynamic allocation of server resources. The technique tries to migrate the VMs from the data centers where the estimated migration revenue is high to the data centers where the estimated migration revenue is low, until there are no VMs left to migrate or the migration revenue cannot be minimized further.

In a set of geo-distributed cloud data center environments, the cost of network resources is complicated to calculate because it is dependent on complex and heterogeneous inter data center WAN links. [150] abstracted geo-distributed data centers into an incomplete undirected graph. The

problem of cloud data center selection is developed as the shortest-path problem from the client/origin vertex to all cloud data center vertices. The edge lengths in the graph were weighted, where the weights represent the distance between data centers or between the data center and a user. The authors considered the weights of the edge (network resource) and the vertex (computation resource) while selecting a data center. For mapping incoming workloads to an appropriate data center, they proposed a Q-learning based data center selection technique to achieve the optimal networking and compute costs. The state space comprised of a set of vertices of the graph and the action space is made up of the graph edges. After the workload moves from one vertex (state) along the edge (action) to another vertex (state), the length of the edge was used as the positive reward value. Through the Q-table, the path with the smallest Q value, i.e., the shortest path between each user and data center, was obtained.

In [151], the authors presented sCloud, a holistic heterogeneity-aware cloud resource management strategy, with the goal of maximizing system throughput in geo-distributed self-sustainable data centers. While taking into consideration renewable power availability and QoS requirements, sCloud adaptively distributed workloads to geo-distributed data centers. The proposed method assigned available resources to heterogeneous workloads in each data center and migrated batch jobs between data centers. The workload placement problem is formulated as a constrained optimization problem that can be solved using nonlinear programming. Moreover, when the green power supply fluctuated greatly at different places, the authors suggested a batch job migration strategy to boost system throughput even further. Finally, the authors enhanced sCloud by adding a Q-learning based RL technique. The job progress for batch jobs at separate data centers are represented by the elements in the RL state space. The action space is a collection of the progress control factors. The progress control factors determined how long the batch job can be delayed

or accelerated compared to the job execution progress achieved in the previous control interval. The reward is the ratio of current goodput to a reference goodput plus penalties incurred when job level SLOs are violated. When tuning resource allocations, a low reward implies that the job execution may miss the soft or hard deadline, both of which should be avoided. The proposed RL based configurable batch job manager allowed dynamically regulation of the job execution process while staying on schedule.

Due to the dynamism of live broadcasting and the broad geo-distribution of viewers and broadcasters, satisfying all requests with adequate resources from geo-distributed data centers remains a challenge. To address this issue, [152] presented a prediction-based approach that assessed the potential number of viewers at distinct cloud sites at the time of broadcasting. This work developed an Integer-Linear Program (ILP) based on the derived predictions to proactively and dynamically identify the best data centers to allocate exact resources and serve potential viewers while reducing perceived delays. Because ILP-based optimization is time consuming and thus inefficient for online serving, the authors presented RL-OPRA, a real-time DQN based RL technique. The authors defined a state as the set of data center index, predicted viewers, cost incurred from previous time-steps, and incremental average delay. An action indicated the index of the site that will serve the viewers. The total reward was determined by adhering to the average delay and reducing the cost of various actions. The proposed RL based solution adaptively learned to optimize allocations and service decisions while interacting with the network environment.

In [153], the problem of energy-aware workload management in geo-distributed cloud data centers considering their green energy generation with variable capacity was investigated. This work introduced a blockchain-based decentralized cloud management architecture to reduce the overall energy cost, workload scheduling cost, and workload migration in data centers. In this

decentralized cloud management, the workload is scheduled by data centers themselves, removing the dependency on a central schedular. In addition, the authors proposed a Q-learning based RL technique incorporated in a smart contract to reduce energy cost even further. The state vector is comprised of data centers' execution load, the reward vector is made up of energy costs, and actions contain all workload migration possibilities. The proposed RL approach migrates workloads among geo-distributed data centers using the information about historical migration decisions and tries to minimize the total energy cost.

The data centers located in buildings that also provide significant space for office rooms are said to inhabit mixed-use buildings (MUBs). It is beneficial to use scheduling flexibility in both the heating, ventilation, and air conditioning (HVAC) system and the data center workload to successfully lower the total energy cost of MUBs. In [154], the authors proposed two DRL frameworks for lowering overall energy cost while maintaining target room temperature and satisfying data center workload deadline constraints. In case of the DRL based algorithm for office HVAC control, airflow rate controls are the actions, temperatures in the various office zones are the states, and the cooling energy cost is the reward. In case of the DRL based data center workload allocation, a server's activation (on/off) control is the action, execution rate of a server is the state, and data center cooling and compute energy cost is the reward. Then, the authors presented a joint office HVAC and data center workload control algorithm in a single MUB. They also created a heuristic DRL-based method to enable interactive workload distribution among geo-distributed MUBs, resulting in even more energy savings. Here the number of servers preferred for use in each MUB is the action, execution rate of an MUB is the state, and total energy cost for all MUBs is the reward.

In [155], the authors discussed the problem of big data analytics cost reduction in geo-distributed data centers powered by renewable energy sources with intermittent capacity. To address this problem, a DQN based RL technique for workload scheduling was proposed. The state vector was composed of current information about the CPU usage, free RAM size, free I/O bandwidth, weather, and electricity price. The action vector consists of the job locations (information about where to migrate the jobs). The reward is comprised of migration costs and energy costs. In addition, two strategies are created to improve the framework's performance. Random Pool Sampling (RPS) is proposed to retrain the NN using collected training data, and a unique Unidirectional Bridge Network (UBN) structure is devised to improve the training time even further by leveraging the historical information stored in the trained NN.

VMs virtualize hardware to allow several OS instances to operate simultaneously. On the other hand, Containers allow users to virtualize an OS so that many workloads can operate on the same machine. Containers also make it possible to bundle a program with all of its dependencies (such as code and libraries), allowing for faster startup, shutdown, and migration times than VMs. The default container placement technique in Kubernetes is unsuitable for a geographically distributed computing environment or dealing with the volatility of computing resources and workload. [156] proposed ge-kube (Geo-distributed and Elastic deployment of containers in Kubernetes), an orchestrated solution that leverages Kubernetes and adds self-adaptation and network-aware placement features. This study provided a two-step control loop in which the number of replicas of specific containers are dynamically controlled based on the application response time using a model-based RL technique, and containers are assigned to geo-distributed computing resources using a network-aware placement mechanism. For the RL technique, the state is made up of number of application instances, their CPU utilization, and their CPU resource limit.

The action has vertical (add or remove CPU share) and horizontal (add or remove containers) scaling control. The reward is total cost, which is defined as the weighted sum of the performance penalty (paid after a deadline miss), resource utilization cost, and the adaptation cost. The study proposed an optimization problem formulation and a network-aware heuristic to address the placement issue, which explicitly took into consideration non-negligible network delays across geo-distributed computing resources to satisfy latency-sensitive workload's QoS needs. This work conducted an extensive series of experiments using a surrogate CPU-intensive application and a real application (i.e., Redis), demonstrating the benefits of combining elasticity and placement strategies, as well as the advantages of adopting network-aware placement solutions.

The authors of [157] investigated intra- and inter-data center energy optimization with both homogeneous and heterogeneous servers and workloads. They initially proposed an optimization model to reduce the combined server and network energy costs. The authors used the Lyapunov optimization model to turn the original problem into a well-studied queue stability problem to address the time-coupling restriction of carbon emission. Then, using the generalized Benders decomposition method, the authors developed a centralized solution. They also enhanced the model to handle heterogeneous workloads and data centers. At the start of every time slot, this technique found a feasible solution for energy optimization. But it followed this method for the entire time slot/epoch, despite changes in the network environment (e.g., occurrence of faults). To address this issue, the authors developed a DQN based RL approach. The state is the status of a data center which includes number of active servers and the frequency of CPU. The action considers a set of data centers and controls which data center to choose. The reward is the total long term energy cost. The proposed DQN based RL method was able to explore and learn the dynamic nature of the network and make run-time decisions based on limited knowledge.

4.6.3.2. VNF PLACEMENT OPTIMIZATION

Service function chaining (SFC) aims to establish VNFs in geo-distributed data centers and facilitates interconnect routing between them. DRL has recently been shown to be beneficial in the field of SFC. But current DRL-based algorithms, possess a large action space which causes poor convergence and scalability. Some studies have found a solution to this challenge by reformulating the SFC problem, which usually leads to low utilization and steep costs. To address this problem, [158] created a hybrid DRL-based system that separates VNF deployment and flow routing into separate modules. A DRL agent's sole responsibility is to learn the VNF deployment policy. Adaptive parameter noise, Wolpertinger policy, and prioritized experience replay are used to adjust the structure of the agent based on DDPG and to increase learning efficiency. A game-based module (GBM) is used to route the flow. A decentralized routing algorithm for the GBM is designed to address the scalability. The DRL agent altered the deployment policy based on the reward generated by the GBM during the learning process. The proposed method outperformed traditional DRL-based algorithms in terms of learning efficiency for two reasons. First, the suggested technique drastically shrank the DRL agent's action space. Second, traditional DRL-based algorithms are almost entirely model-free. The GBM, on the other hand, used model-based algorithms to optimize flow routing in the proposed method. The model-based information, such as the gradient, led to notable improvements in the algorithm's efficiency.

4.6.4. HYBRID ML MANAGEMENT

Most cloud management solutions use one problem-solving technique in their operational workflow. But some are required to achieve multiple goals at the same time, such as predicting the incoming workload and then allocating it to resources, or clustering a set of resources based on the current execution state and then provisioning them dynamically. These workflows can benefit from

separate ML techniques for each objective. Hybrid ML methods have been developed to overcome drawbacks of standalone techniques by combining one or more ML techniques together. A few workflows that we consider in this survey are required to predict some resource or workload parameters before starting the dynamic optimization process. Accordingly, the hybrid ML methods that we discus here combine parameter prediction and cloud optimization.

Designing efficient scaling algorithms is difficult, particularly for geo-distributed VNF chains, where WAN traffic bandwidth costs and latencies are essential, yet difficult to account for when scaling decisions are made. To resolve this problem, [159] offered a deep learning-based methodology, analyzing intrinsic patterns of traffic volatility and efficient deployment techniques over time, with the goal of improved judgments enabled by in-depth learning from experiences. An LSTM is used for predicting future flow rates, and a DRL agent is used to make dynamic VNF chain placement decisions. To improve results, this study used an experience replay technique based on the actor–critic DRL algorithm. In comparison to previous representative algorithms, trace-driven simulation showed that the newly designed learning framework reacted swiftly to traffic dynamics online and achieved lower system costs with minimum offline training.

Web app developers previously had to store more data object copies in many geo-distributed data centers or send duplicate requests to multiple (e.g., closest) data centers to provide low request latency to users, both of which increased monetary cost. To address this issue, [160] proposed an RL based geo-distributed cloud storage system named GeoCol, with the goal of achieving low cost and latency. First, this new system included a request split mechanism and a storage planning method to achieve the best tradeoff between monetary cost and request latency. To enable parallel transmissions for a data object, the request split approach used the SARIMA technique to anticipate request latency as an input to an RL model to calculate the number of sub-requests and

the data center assigned to each sub-request. Second, GeoCol employed another RL method to decide if it needs to store each data object and the storage type of each stored data object to help reduce the storage cost.

In [161], the authors studied ways to connect different renewable energy generators to geo-distributed data centers from various cloud providers in order to reduce carbon emissions, monetary costs, and service level objective (SLO) violations caused by renewable energy shortages. The authors tested multiple ML approaches for long-term forecast accuracy on renewable energy generation and demand, and chose SARIMA for the prediction. Based on the projected findings, the study presented a multi-agent RL (MARL) approach for each data center to determine how much renewable energy to request from each generator. This research also presented a Deadline Guaranteed Job Postponement approach (DGJP) for deferring the execution of non-essential jobs when renewable energy supplies are insufficient.

4.7. DISCUSSIONS

4.7.1. ML TECHNIQUE CATEGORIZATION

To handle the problem of reliable cloud management for geographically distributed data centers, several sub-problems must be addressed. We looked at a variety of methods and strategies presented for such reasons in the previous section. Most use one, but a few of them use multiple ML techniques in their operational flow. According to the information presented in Section 4.1.3, ML approaches contain any models that can be trained using data to perform supervised, unsupervised, or RL based tasks. In Table 10, we categorize the ML models used in all papers discussed in this survey into four different types. Figure 35 shows the percentage of ML model types considered by various articles.

Table 10: ML methods used in the survey

ML Type	ML Algorithms
Clustering	*k*-means, Spectral
Regression	LR, Poly LR, LASSO, DT, RF, GBRT, XGBoost, *k*-NN, SVM, SARIMA
Neural Networks	SCN, MLP, CNN, LSTM, GRU
Reinforcement Learning	Model-based, Actor-critic, Q-learning, DQN, DDPG, DRL, MARL

From Figure 35, by looking at the limited usage of clustering-based learning techniques (*k*-means and spectral), we can say that these are the least suitable for cloud management because they group workloads or resources into clusters, but at a scale and granularity that is not particularly beneficial for cloud management frameworks. The only time when unsupervised clustering can be advantageous is when the workload or resources change very frequently.

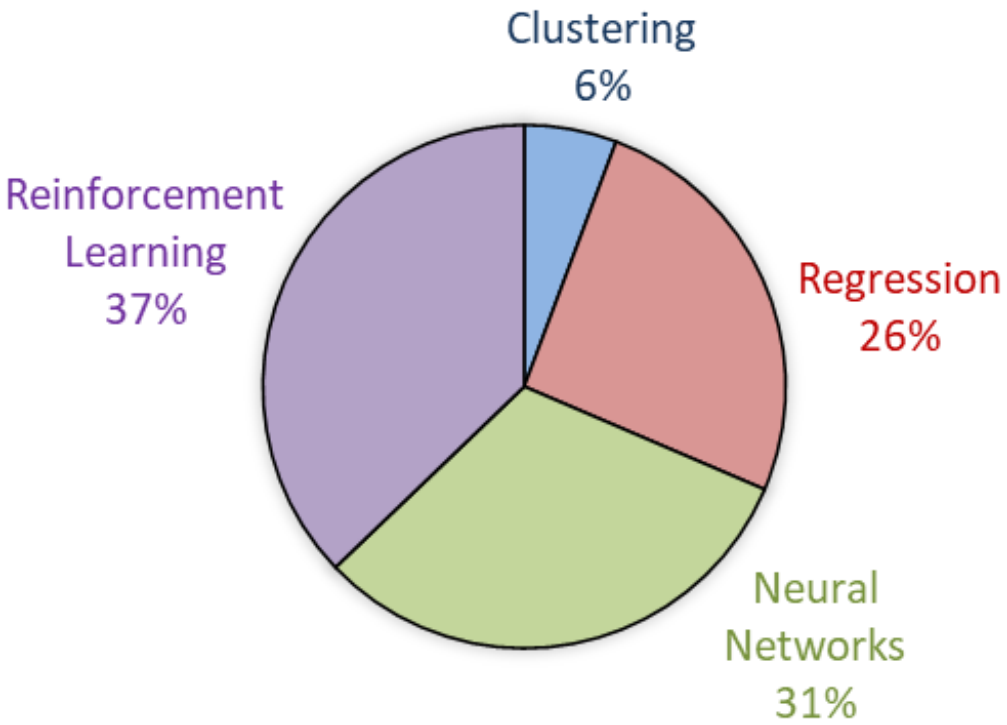


Figure 35: Percentage of ML model types considered in prior work

As per the findings of this review, workload and resource parameter prediction is a critical task in large-scale geo-distributed cloud computing infrastructures. High-accuracy predictions enable reliable management of incoming workload and available resources, ensuring QoS and SLAs while lowering cloud operating costs. However, predicting workload and resource parameters is a difficult problem in general, especially when they are dependent on unpredictable activities (e.g., web servers, IoT devices, biological sensors, smartphone /PC workloads, resources failures). Regressive models, such as LR, Lasso, or SARIMA, are often used to make such predictions. A new generation of data analysis models based on NN, on the other hand, has just been embraced, and extensive evaluations of them have shown encouraging results. On average, RNN-based models, e.g., LSTMs, seem to produce better predictions quickly compared to traditional regression based models [144].

In practice, various challenging characteristics of geo-distributed cloud systems, e.g., massive scale and heterogeneity of systems, randomness of workload arrival parameters and resource availability, and timeliness of scheduling decisions, make it very difficult to design accurate models, which necessitates the use of a high-speed scheduling algorithm. It is difficult for a single Heuristic or Metaheuristic algorithm to adequately adapt to the real-world dynamics of geo-distributed cloud computing systems. In such scenarios, the profound use of RL algorithms in reviewed articles shows that it can be a very promising for cloud computing management. In the area of ML based optimization, RL is in fact the most prevalent method explored.

4.7.2. CHALLENGES AND FUTURE OPPORTUNITIES

Having discussed the current state-of-the-art in geo distributed cloud management, we now discuss several research challenges and opportunities in this section that address the open issues

highlighted in current literature and hence represent viable future research directions in the area of ML for geo-distributed cloud data center management.

Advanced ML algorithms: Current studies use traditional ML algorithms such as regression, *k*-means clustering, Q-learning, etc. More advanced NN and Deep Learning (DL) techniques that have been shown to successfully and more accurately solve large scale problems in other application domains, could be adapted for the geo-distributed cloud management domain. For example, LSTMs can predict future values based on previous sequential data. They can be used where lightweight and fast time series forecasting models are needed. If a little more computational overhead is acceptable, Transformer ML [163] models can be used instead of LSTMs. They can process much larger amounts of data in the same amount of time because of the parallelizability of the transformer mechanism. DL is especially useful when dealing with problems that have a high-dimensional state space. In DRL, because of DL's ability to learn different levels of abstraction from data, RL can perform increasingly sophisticated tasks with less prior information. Emerging Soft Actor-Critic (SAC) [164] RL techniques can optimize a stochastic policy in an off-policy manner, bridging the gap between stochastic policy optimization and DDPG-style techniques. It employs entropy regularization, in which the policy is trained to maximize the trade-off between expected return and entropy (randomness in the policy). Due to entropy factor regularization, SAC can be highly efficient for energy-based optimization approaches.

Efficient ML training and deployment: One component of the design of parallel ML algorithms is to leverage several threads/processes to speed up the ML model's learning speed. In some cases, NNs require a large amount of training data, a significant amount of processing time, and even specific hardware to run parallel ML algorithms, such as servers with GPUs. Instead of using such

computationally heavy ML algorithms (e.g., NN, Deep Learning, RL, etc.), researchers need to propose fast and lightweight ML solutions that can be deployed on commodity hardware. Model compression techniques (pruning, quantization, Low-rank factorization, Knowledge distillation, etc.) are particularly promising to reduce model inference overheads., particularly if the ML models are executed frequently. Model compression is a strategy for deploying cutting-edge ML models with lower memory usage and reduced latency, without sacrificing accuracy. Many cloud management studies have developed efficient ML-based profiling, prediction, or optimization algorithms. But most of these ML techniques are not designed to work online (for inference) and make real-time decisions after processing large volumes of real-time data. These existing techniques could be converted to online ones (e.g., RL or DRL based algorithms) to work with profiling and prediction methods, with model compression to improve efficiency. Another element to investigate is using ensemble learning and RL together for quick deployment.

Benchmarks/Workloads: Most proposed ML models have not been trained and evaluated using big, high-quality datasets obtained by strong industrial cloud operators in production scenarios. The majority of prior work uses synthetic, small datasets, or datasets that are not reflective of real-world geo-distributed events. Because there is a scarcity of current and relevant data (for workloads, resource usage, and trends) from geo-distributed cloud environments, it is difficult to accurately assess the quality of published results and, more importantly, to compare the results of competing studies. A large scale, realistic, publicly available geo-distributed cloud data center dataset from a major cloud vendor would serve as a standard, allowing researchers to analyze and compare alternative ideas and approaches on a much larger scale.

State-of-the-art Distributed Computing Systems: A logical step would be to explore the design and construction of a distributed version of the proposed ML algorithms and build the complete management system using cutting-edge big data technologies, e.g., Apache Spark, that can be deployed in geo-distributed cloud data centers.

Industry Involvement for Validation: It would be ideal to have more industrial participants involved in the studies, not only by supplying appropriate requirements, but also by contributing to the evaluation of the results, and especially the deployment of small-scale pilots in production settings. Although small testbeds that are used in some studies allow for the testing of many ideas and configurations, it is impossible to draw definitive or full findings without a thorough examination in a production setting.

Heterogeneous Resources/Hardware: Most studies consider traditional hardware resources e.g., server nodes, CPUs, etc. The proposed approaches in these studies could be extended to consider more complicated user requests with multiple types of resources other than CPUs, e.g., disaggregated memories, GPUs, accelerators, network switches, emerging cooling systems, and considering heterogeneity between them.

Networks: Inter and intra-data center networks play a very important role during cloud management (reducing operating costs, QoS, SLA, etc.). With ever increasing data volumes in new cloud workloads, it is very important to factor in inter as well as intra data center networks, and the network costs related to the data transfers across geo-distributed data centers [107]. Studies must include network related parameters in the optimization function to resemble state-of-the-art cloud management systems.

Energy Modeling: Most if not all studies included in this survey consider simple data center energy models that are based on CPU/node utilization. In the data center energy model, for improved accuracy, researchers could consider various factors such as different performance states (P states) of cores, thermal aware cooling power, net metering and peak shaving, i.e., peak power reduction, etc. [137].

Multi-objective Algorithms: Unlike many existing studies, real-world cloud management systems must address more than one or two objectives. Objectives of future studies can be modified to include more optimization metrics. For example, data center computing/network performance (latency, SLAs, QoS, etc.) and data center operating/network costs could be optimized jointly; data center brown energy consumption, renewable energy generation, and carbon emission could be considered; and data center performance, uptime/availability, and fault tolerance could also be optimized jointly.

4.8. CONCLUSIONS

In this survey, we reviewed how the challenges of reliable geo-distributed cloud management have been addressed using ML techniques in the scientific literature. For simplification, we dissected this problem into three sub-categories: profiling, parameter prediction, and cloud optimization. Moreover, we discussed how ML methods have been used to improve the reliability of cloud management algorithms in diverse and heterogeneous environments. The number of studies that use ML-based approaches in this area has increased dramatically in recent years. Several researchers have employed a variety of methodologies, ranging from traditional statistics and regression to advanced ML algorithms. As per the results compiled in last two columns of Table 9, we conclude that, on average, ML methods outperform traditional

mathematical techniques, particularly when dealing with large and complicated environments. The progression of ML methodologies in current research is illustrated in this chapter, which we hope will help readers comprehend the research gaps in this topic. Finally, based on the challenges discovered in the surveyed studies, several prospective future research directions and opportunities are identified to strengthen the current ML approaches. These challenges and future opportunities are discussed in detail in Section 4.7.2.

5. GAME-THEORETIC DEEP REINFORCEMENT LEARNING TO MINIMIZE CARBON EMISSIONS AND ENERGY COSTS FOR AI WORKLOADS IN GEO-DISTRIBUTED DATA CENTERS[4]

Data centers are increasingly using more energy due to the rise in Artificial Intelligence (AI) applications, which negatively impacts the environment and raises operational costs. Reducing operating expenses and carbon emissions while maintaining performance in data centers is a challenging problem. This chapter introduces a unique approach combining Game Theory and Deep Reinforcement Learning (DRL) for optimizing the distribution of AI workloads in geo-distributed data centers to reduce carbon emissions and cloud operating (energy + data transfer) costs. The proposed technique integrates the principles of non-cooperative Game Theory into a DRL framework, enabling data centers to make intelligent decisions regarding workload allocation while considering the heterogeneity of hardware resources, the dynamic nature of electricity prices, the inter-data center data transfer costs, and carbon footprints. We conducted extensive simulation experiments comparing our game-theoretic DRL (GT-DRL) approach with current DRL-based and other optimization techniques. The results demonstrate that our strategy outperforms the state-of-the-art in reducing carbon emissions and minimizing cloud operating costs without compromising computational performance. This research has significant implications for achieving sustainability and cost-efficiency in data centers handling AI workloads across diverse geographic locations.

[4] The research presented in this chapter is submitted to the *IEEE Transactions on Sustainable Computing* journal in June 2023 and is under review.

5.1. MOTIVATION AND CONTRIBUTION

The internet has evolved into a crucial aspect of modern life, with billions of users worldwide and the ability to support various services and apps. Approximately 5.25 billion people have access to and utilize the internet globally as of 2023, making up 66.2% of the expected 7.9 billion people on the planet [1]. The use of data centers to support Internet applications is growing rapidly, with experts predicting increased reliance on data centers fueled by Artificial Intelligence (AI)-driven workloads, new cloud services, demand for edge applications, expansion of the Internet of Things (IoT) devices, and the proliferation of technologies such as 5G/6G.

The rise of AI workloads in data centers has been a significant trend in recent years. AI applications require immense computing power, especially for training complex deep learning models. They also need vast amounts of data for training and improving models, leading to increased data transfer and storage requirements. As a result, AI-driven computing is fueling demand for more power-hungry data centers with expanded storage capabilities [165]. In addition to storage and computing power, the widespread use of AI applications has led to a growing need for real-time processing and decision-making. These workloads have gained prominence in data centers due to several factors, including the growth in smart edge computing (e.g., with increasingly autonomous vehicles) and the proliferation of IoT applications (e.g., those used in smart home devices) [166].

Cloud service providers are investing significantly in extending their data center infrastructure to accommodate the rising demand for cloud services and AI workloads. In particular, there has been a shift towards building and adding data centers across multiple geographic locations. There are several reasons to favor geographically distributed data centers as they provide advantages in load balancing, reliability and redundancy, latency reduction, and

compliance with data sovereignty rules [167]. Another compelling reason for geographically spreading data centers is to reduce electricity costs by taking advantage of Time-Of-Use (TOU) electricity pricing. According to the TOU electricity pricing model in [26], electricity prices fluctuate depending on the time of day. When demand for the entire electrical grid is high, electricity costs rise, whereas when the demand is low, they decline [27]. Utilities also impose an additional flat-rate (peak demand) cost on non-residential or commercial customers depending on the maximum (peak) power utilized at any given moment within a monthly billing period [28]. These peak demand fees sometimes exceed the amount of the electricity bill devoted to energy use. One efficient way to lower electricity costs is to move workloads between geo-distributed data centers. With the help of this strategy, cloud service providers can distribute workloads to data centers with lower energy prices. This can reduce the cost of cloud computing for clients and operating costs for cloud service providers.

Today, major cloud service providers run multiple massive power-hungry data centers around multiple geographic locations. These data centers consume about 3% of global electricity and contribute approximately 2% of all GreenHouse Gas emissions (GHG) emissions worldwide [168]. During the COVID pandemic, the increased reliance on web and cloud services increased internet traffic and data center utilization drastically. If this trend continues, it has been predicted that data center GHG emissions may increase to 5–7% of global emissions, which is a major concern [168]. Thus, making data centers carbon-conscious and energy efficient has become a top priority for operators and cloud service providers.

When making decisions concerning AI workload management in geographically dispersed data centers, it is crucial to consider all the factors mentioned above. It is a challenging (NP-hard) problem to orchestrate complex, heterogeneous, large-scale software and hardware systems and

automate computing platforms used in geo-distributed cloud computing systems. Researchers in this area have traditionally solved such problems using several mathematical methods, such as linear, non-linear, convex, integer, stochastic, combinational, dynamic, heuristic, probability, and hybrid optimization approaches. As discussed in our previous work [107] and [137], some algorithms are very good (e.g., genetic algorithms) at finding solutions but take too long to execute. Some algorithms are fast but susceptible to getting stuck in local minima (e.g., greedy and game-theoretic algorithms) [42], [137]. Owing to the computational intensity and intricate characteristics of the data, coupled with the continuously changing problem space, such mathematical methods often exhibit limited applicability in large-scale dynamic distributed systems and encounter challenges in scalability concerning geographically distributed architectures. As a result, researchers have been investigating intelligent data-driven and Reinforcement Learning (RL) based optimization alternatives.

In RL, an agent learns to solve a problem by trial and error through interacting with the environment. It uses a feedback loop that includes rewards, which act as prompt assessments of the efficacy of the prior action. A reward is an immediate return (feedback) from the environment to assess the quality of the previous action. The agent represents the RL algorithm, while the environment refers to the item on which the agent is acting. The environment sends a current state to the agent to start the RL process. The agent chooses an action based upon prior knowledge or a predetermined policy, where the policy is the agent's strategy for determining the next course of action based on the current state. The environment gives the agent the new state and related reward after completing an action. The agent then updates its knowledge or policy using this information. The cycle continues until the environment sends a terminal state, ending the episode. By combining Deep Learning (DL) with RL, Deep Reinforcement Learning (DRL) has been

demonstrated to work better than traditional RL algorithms [169]. As discussed in our literature review on machine learning approaches for geo-distributed data center management [170], many works have recently explored using DRL, showing the feasibility of such solutions in real cloud environments. In DRL, Deep Neural Networks (DNNs) approximate the policy. To maximize its cumulative reward over time, the agent takes a sequence of actions and learns how to make better decisions by updating its DNN (policy).

However, DRL techniques have disadvantages, including high data and computational intensity, sample inefficiency, exploration and exploitation dilemma, struggle with dynamic environments, and difficulty defining rewards [170]. Therefore, to address the intrinsic shortcomings of using DRL alone, we combine DRL with a Nash equilibrium-based fast non-cooperative game-theoretic technique [137] to propose a novel game-theoretic DRL (GT-DRL). The proposed approach can result in a more adaptive, scalable, and efficient methodology for workload management in geo-distributed data centers.

This chapter aims to create and assess a workload management strategy for geographically distributed data centers that aims to reduce cloud carbon emissions and operational costs across all data centers for processing incoming workloads. We define cloud carbon emissions as the total carbon emissions of all geo-distributed data centers used by a cloud service provider. Similarly, we define cloud operating costs as the total operating (energy and data transfer) costs of all combined geo-distributed data centers in the cloud infrastructure. We use real workloads and arrival patterns and compare our approach to state-of-the-art mathematical techniques and other DRL methods to evaluate it. Our research makes the following innovative contributions, in brief:

- we formulate the cloud workload distribution problem as a Nash equilibrium-based non-cooperative game;

- we then combine the game theory with deep reinforcement learning (DRL) and design a new intelligent game-theoretic DRL (GT-DRL) workload distribution framework to minimize the overall cloud carbon emissions and operating costs for geographically distributed heterogeneous data centers;
- the workload scheduling decisions leverage detailed models for a comprehensive set of characteristics that impact cloud carbon emissions and workload performance, including data center compute and cooling power, co-location performance interference, and variable renewable power available across different data center locations.

5.2. RELATED WORK

We analyzed the recent literature on RL-based workload management for geo-distributed data centers. There are two primary types of RL algorithms: model-based and model-free RL. In model-based RL, the agent tries to create a model of the environment and optimizes its policy based on this model. In [156], the authors use a model-based RL as part of a two-step control loop in which a network-aware placement policy allows containers on geographically distributed computing resources, and a model-based RL technique dynamically adjusts the number of copies of individual containers based on the application response time. Model-based RL techniques work well for a small problem where all environment variables are known. However, the solution space grows when the problem becomes large, such as in the case of the geo-distributed heterogeneous workload to data center resource mapping problem. In these cases, such algorithms become impractical.

On the other hand, model-free RL is independent of an environmental model. Instead, the agent uses a trial-and-error strategy to determine the best action. Model-free RL can be divided

into two methodologies: Q-learning (value iteration) and policy optimization (policy iteration). Q-learning learns the action-value function. In other words, it aims to identify the best action for a particular state using a Q-table, which is a lookup table that stores the highest possible predicted rewards for actions at each state. The authors from [149], [153], [150], and [151] use Q-learning for their RL-based geo-distributed data center workload management. Deep Q-Network (DQN) is a variation of Q-learning that uses DNNs. DQN has been used for large problems with large state-action spaces where creating a Q-table would be extremely difficult, time-consuming, and complex. In DQN, DNNs approximate Q-values for each action based on the state. The approaches presented in [154], [152], [155], [157], [160], and [171] use DQN for run-time workload to resource mapping for geo-distributed data centers.

In policy optimization methods, the agent directly learns the policy function that maps the state to action. A value function or Q-table is not used to decide the policy. Actor-Critic methods combine both Q-learning and policy optimization. The critic component estimates the action-value (Q-value) or state-value function, while the actor adjusts the policy distribution according to the critic's suggestions. Some complex optimization problems need to have a continuous action space. Deep Deterministic Policy Gradient (DDPG) is a type of actor-critic RL algorithm that extends DQN to the continuous action space. It uses an actor-critic architecture; and employs a deterministic policy that influences the loss function used for updates. DDPG is an off-policy method where the agent learns from data collected by a separate policy. In this algorithm, the Policy Gradient (PG) part uses gradient descent to optimize parametrized policies concerning the expected return (long-term cumulative reward). An important component of cloud resource management involves Virtualized Network Functions (VNFs) that are created on open computing platforms and run virtual network services on common servers in cloud data centers. Virtualized

routers, firewalls, network optimization, and Network Address Translation (NAT) services are all examples of VNFs. The authors in [158] use DDPG for efficient VNF deployment. Proximal Policy Optimization (PPO) is another type of actor-critic RL algorithm that updates its decision-making policy by collecting a small batch of experiences interacting with the environment. It is an on-policy learning method where the acquired experience samples are only used once to update the current policy. PPO ensures the new policy updates do not deviate too far from the prior ones. The methodology presented in [172] uses information about electricity pricing and renewable energy generation and develops a PPO-based workload management technique to minimize cloud operating costs. The strategies proposed in [158] and [172] minimize the cloud operating costs in a geo-distributed environment but do not consider carbon emissions. In this chapter, we propose a new game-theoretic version of DRL for geo-distributed workload management and study the effects of cloud carbon emissions and operating costs. Additionally, we compare our proposed strategy against state-of-the-art RL methods, including [158] and [172].

Our previous work [137] uses a Nash Equilibrium-based game-theoretic technique for geo-distributed workload management. However, the Nash equilibrium-based game-theoretic technique does not consistently achieve global optima solutions. The scope of this research goes beyond [137] by developing a new carbon emission model that considers the effects of renewable energy generation at different data center locations while making workload distribution decisions. Moreover, we propose a new game-theoretic DRL-based workload management technique that takes a holistic approach to the cloud carbon emissions and operating costs minimization problem, which is shown to be more effective than the multiple resource management strategies presented in [137], [158], and [172]. More details about the comparison strategies are provided in Section 5.6, and a thorough analysis to evaluate the improvement is presented in Section 5.7.

5.3. SYSTEM MODEL

5.3.1. OVERVIEW

Our proposed framework comprises a geo-distributed-level Cloud Workload Manager (CWM) that distributes incoming workload requests to geographically distributed data centers. Each data center has its own local Data center Workload Manager (DWM) that takes the workload assigned to it by the CWM and maps requests to compute nodes within the data center. We first describe the system model at the geo-distributed level and then provide further details into the models of components at the data center level.

5.3.2. CLOUD LEVEL MODEL

We consider a rate-based workload management scheme. In our work, an epoch length T^e is one hour, and a 24-epoch period represents an entire day. Within the short duration of an epoch, the workload arrival rates can be reasonably approximated as constant. This presumption is based on the fact that the rate of requests or workload coming to a data center frequently appears fairly constant over short time intervals. This is especially true when considering the collective behavior of numerous users or services, where little deviations can average out [173], [174]. CWM makes workload allocation decisions in advance, i.e., before the start of an epoch. As the epoch length is one hour, the workload to resource mapping optimization technique can take up to one hour to calculate the workload distribution strategy.

Let D be a set of $|D|$ data centers, and let d represent an individual data center. We assume that a cloud infrastructure (Figure 36) comprises $|D|$ data centers, that is, $d = 1,2, \dots, |D|$ and $d \in D$. Let I be a set of $|I|$ task types, and i represents an individual task type. We consider $|I|$ task (i.e., workload) types, that is, $i = 1,2, \dots, |I|$ and $i \in I$. A task type $i \in I$ is characterized by its

arrival rate and execution rate, i.e., reciprocal of the estimated time required to complete a task of task type i on each of the heterogeneous compute nodes, in each performance state (P-state). We assume that the beginning of each epoch τ represents a steady-state scheduling problem where the CWM splits the cloud level arrival rate $CAR_i(\tau)$ for each task type i into the local data center level arrival rate $AR_{i,d}(\tau)$ and assigns it to each data center $d \in D$. That is,

$$\sum_{d=1}^{|D|} AR_{i,d}(\tau) = CAR_i(\tau), \quad \forall i \in I. \tag{52}$$

The CWM performs this assignment such that the cloud carbon emissions and operating (energy and network) costs across all data centers are minimized, with the constraint that the execution rates of all task types meet or exceed their arrival rates at each data center, i.e., all tasks complete without being dropped or unexecuted. At the start of each epoch, the CWM calculates (explained in Section 5.3.3.2) a co-location aware data center maximum execution rate $ER_{i,d}$ for each task type i at each data center d. Later, it splits cloud level arrival rate $CAR_i(\tau)$ for each task type i into local data center level arrival rates $AR_{i,d}(\tau)$ such that the data center's maximum

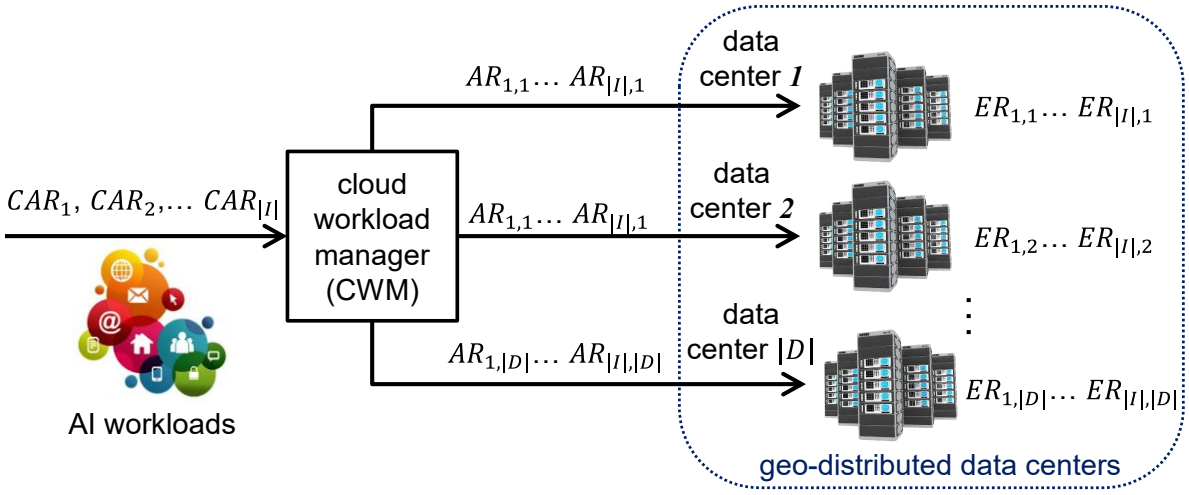


Figure 36: Cloud workload manager (CWM) performing geo-distributed level task (workload) assignment to data centers.

execution rate $ER_{i,d}(\tau)$ meets or exceeds the corresponding arrival rate $AR_{i,d}(\tau)$, thus ensuring the workload is completed. That is,

$$ER_{i,d}(\tau) \geq AR_{i,d}(\tau), \quad \forall i \in I, \forall d \in D. \tag{53}$$

5.3.3. DATA CENTER LEVEL MODEL

5.3.3.1. ORGANIZATION OF EACH DATA CENTER

Each data center d houses NN_d number of nodes and a cooling system comprised of computer room air conditioning (CRAC) units. Let NCR_d be the number of CRAC units. Heterogeneity exists across compute nodes, where nodes vary in their execution speeds, power consumption characteristics, and the number of cores. The number of cores in node n is NCN_n, and NT_k is the node type to which core k belongs.

5.3.3.2. CO-LOCATION-AWARE EXECUTION RATES

Tasks competing for shared memory in multicore processors can cause severe performance degradation, especially when competing tasks are memory intensive. The memory intensity of a task refers to the ratio of last-level cache misses to the total number of instructions executed. We use a linear regression model that combines disparate features based on the current tasks assigned to a multicore processor to predict the execution time of a target task i on core k in the presence of performance degradation due to interference from task co-location [65].

We classify the task types into memory-intensity classes on each of the node types and calculate the coefficients for each memory-intensity class using the linear regression model to determine a co-located execution rate for task type i on core k, $CoER_{i,k}^{core}(\tau)$ [65]. When

considering co-location at a data center d, the co-location aware data center execution rate for task type i is given by:

$$ER_{i,d}(\tau) = \sum_{n=1}^{NN_d} \sum_{k=1}^{NCN_n} CoER_{i,k}^{core}(\tau). \quad (54)$$

The linear regression model was trained using execution time data collected by executing benchmarks from the AIBench [175] benchmark suite on a set of server-class multicore processors that define the nodes used in our study (see Section 5.6 for node type details). This model for execution time prediction under co-location interference is derived from real workloads and real server machines and has a mean prediction error of approximately 7% [65].

5.3.3.3. DATA CENTER POWER MODEL

We use the detailed data center power model from our prior work [107]. Let $CRP_{d,c}(\tau)$ be the CRAC power consumption of CRAC unit c in data center d. $NP_n(\tau)$ is node power consumption of node n. Each data center is partially powered by either solar power, wind power, or some combination of both. The total renewable power, $RP_d(\tau)$, available at data center d is the sum of the wind and solar power available in epoch τ. $RP_d(\tau)$ can be zero if no renewable power is available. Let Eff_d be an approximation of the power overhead coefficient in data center d due to the inefficiencies of power supply units. Eff_d is always greater than or equal to 1. The total net data center power consumption, $DP_d(\tau)$, at data center d, is calculated as:

$$DP_d(\tau) = \left(\sum_{c=1}^{NCR_d} CRP_{d,c}(\tau) + \sum_{n=1}^{NN_d} NP_n(\tau) \right) \cdot Eff_d - RP_d(\tau). \quad (55)$$

For epoch τ, $DP_d(\tau)$ can be negative if the renewable power available at the data center, $RP_d(\tau)$, is greater than the cooling and computing power used in the data center.

5.3.3.4. DATA CENTER CARBON EMISSION MODEL

The unique combination of energy sources shapes carbon emissions from electricity consumption in each geographic location. Other factors influencing carbon emissions include the efficiency of power generation and electricity use and the extent of renewable energy generation at a location. As a result, the emissions can fluctuate hourly, daily, monthly, and annually. The U.S. Energy Information Administration (EIA) provides estimates of these emissions on a monthly and annual basis. In 2021, power plants burning coal, natural gas, and petroleum fuels were responsible for around 61% of the total yearly U.S. utility-scale electricity net generation. However, they accounted for 99% of U.S. carbon emissions associated with utility-scale electric power generation. EIA publishes annual carbon emissions and average annual carbon emissions factors related to total electricity generation by the electric power industry in the United States and each state [176]. These carbon factors, C_d^{factor}, are in kilograms of carbon emitted per kWh. And the value differs for every data center location. The data center carbon emissions $DE_d(\tau)$ at data center d can be defined as:

$$DE_d(\tau) = C_d^{factor} \cdot DP_d(\tau). \tag{56}$$

5.3.3.5. NET METERING MODEL

Net metering allows data center operators to sell back the excess renewable power generated on-site to the utility company. When the extra power is added to the grid, utility companies pay a

fraction of the retail price. The value of this factor varies by data center location. This fraction is called the net metering factor, α_d.

5.3.3.6. PEAK DEMAND MODEL

Most utility providers charge a flat-rate (peak demand) fee based on the highest (peak) power consumed at any instant during a given billing period, e.g., a month. The peak demand price per kW at data center d is denoted as P_d^{price}. We define $Pc_d^{peak}(\tau)$ as the highest grid power consumed since the beginning of the current month, including the current epoch τ. We define $Pp_d^{peak}(\tau)$ as the highest grid power consumption since the beginning of the current month until the start of the current epoch τ. The peak power cost increase at data center d, $\Delta_d^{peak}(\tau)$, is then defined as:

$$\Delta_d^{peak}(\tau) = P_d^{price} \cdot \left(Pc_d^{peak}(\tau) - Pp_d^{peak}(\tau)\right) \tag{57}$$

if $Pc_d^{peak}(\tau) \geq Pp_d^{peak}(\tau)$, else it is equal to 0.

The peak power cost increase $\Delta_d^{peak}(\tau)$ is calculated in each epoch τ and summed over all epochs in a billing period to calculate the total peak power cost.

5.3.3.7. NETWORK COST MODEL

Tasks and their associated data that are migrated across data centers incur a network cost. Two principal components of the network cost are the network price per data traffic unit ($/GB), N^{price}, and the amount of data volume (GB) for the number of tasks migrated (outward). We assume that N^{price} is the same across all data centers. Let S_i be the size (in GB) of a task type i. The network cost at data center d, $NC_d(\tau)$, is calculated as

$$NC_d(\tau) = \sum_{i=1}^{|I|} \sum_{n=1}^{NN_d} \left(N^{price} \cdot S_i\right). \tag{58}$$

5.3.3.8. DATA CENTER COST MODEL

The electricity price per kWh at data center d is defined as $E_d^{price}(\tau)$. Data center operators can use net metering if the total net power consumed throughout the data center, $DP_d(\tau)$, is negative. For such conditions, the total data center cost $DC_d(\tau)$ for data center, d can be defined as

$$DC_d(\tau) = E_d^{price}(\tau) \cdot \alpha_d \cdot DP_d(\tau) + \Delta_d^{peak}(\tau) + NC_d(\tau) \tag{59}$$

where α_d= 1 if $DP_d(\tau)$ is positive and $0 \leq \alpha_d \leq 1$ otherwise.

The first term in (59) represents the TOU electricity cost, the second term represents the peak demand cost, and the third represents the network cost.

5.4. PROBLEM FORMULATION

5.4.1. OVERVIEW

We consider a scenario with multiple data centers maintained by a cloud service provider. The system is assumed to be under-subscribed in the sense that the system is expected to have enough computation resources to complete the workload without requiring that any tasks be dropped or terminated before completion. The tasks originate off-site from the data centers, and we do not consider the transmission time and cost from a task origin to a data center. If the CWM migrates a task from one location to another, there is a data transfer cost associated with it (as discussed in Section 5.3.3.7). The main objective of a CWM is to allocate the workload across

geo-distributed data centers to minimize cloud carbon emissions (the sum of (56) across all data centers). We also consider the related problem of minimizing the monetary cloud operating (energy and network) costs of the system (the sum of (59) across all data centers). Our proposed technique ensures that all workloads are completed according to the constraint defined by (53).

The cloud workload allocation problem's complexity makes it NP-hard, so we propose a game-theoretic DRL-based workload management framework. We model the geo-distributed level workload distribution problem as a non-cooperative game. In a non-cooperative game, there could be a finite (or infinite) number of players who aim to maximize/minimize their objective independently but ultimately reach an equilibrium. For a limited number of players, this equilibrium is called the Nash equilibrium [137].

5.4.2. OBJECTIVE FUNCTIONS

The main goal of our proposed framework is to minimize aggregate cloud data center carbon emissions. Alternatively, our framework also allows updating the objective function to optimize cloud operating (energy and network) costs. In this research, we consider single objective optimization and do not optimize carbon emissions and operating costs together.

Let J_d be a set of node types in a data center d, and let j represent an individual node type. In a data center d, there are a total of $NN_{d.j}$ nodes of node type j. Let P_j^D be the average peak dynamic power for node type j. It is calculated by averaging (for all task types) the peak power for each task type i executing on node type j. Let DP_d^{max} be the maximum data center power consumption possible at data center d, which can be calculated as:

$$DP_d^{max} = \left(NCR_d \cdot CRP_{d,c}^{max} + \sum_{j \in J_d} NN_{d,j} \cdot P_j^D \right) \cdot Eff_d. \quad (60)$$

Recall that $RP_d(\tau)$ is the total renewable power available at the data center d. Let $DP_{i,d}^{est}(AR,\tau)$ be the estimated data center power consumption possible for each task type i at data center d, which can be calculated as:

$$DP_{i,d}^{est}(AR,\tau) = \left(\frac{DP_d^{max} \cdot AR_{i,d}(\tau)}{ER_{i,d}(\tau)} - RP_d(\tau)\right). \tag{61}$$

Observe that, at data center d with zero $RP_d(\tau)$ for each task type i, $DP_{i,d}^{est}(AR,\tau)$ in (61) will increase to DP_d^{max} as the data center level arrival rate $AR_{i,d}(\tau)$ approaches to its maximum execution rate $ER_{i,d}(\tau)$. Thus, we can say that $DP_{i,d}^{est}(AR,\tau)$ is function of AR and τ.

5.4.3. Estimated Cloud Carbon Emissions

The estimated data center carbon emissions $DE_{i,d}^{est}(AR,\tau)$ incurred by the task type i at data center d with that data center maximum execution rate $ER_{i,d}(\tau)$, can be calculated by modifying (56) as:

$$DE_{i,d}^{est}(AR,\tau) = C_d^{factor} \cdot DP_{i,d}^{est}(AR,\tau). \tag{62}$$

The estimated cloud carbon emissions incurred for the task type i, $CET_i^{est}(AR,\tau)$, can be calculated as:

$$CET_i^{est}(AR,\tau) = \sum_{d=1}^{|D|} DE_{i,d}^{est}(AR,\tau) \tag{63}$$

Our optimization goal is to minimize the estimated cloud carbon emissions incurred for all task types, $CE^{est}(AR,\tau)$, which can be calculated as:

$$CE^{est}(AR,\tau) = \sum_{i=1}^{|I|} CET_i^{est}(AR,\tau) \tag{64}$$

subject to the constraints defined by (1) and (53).

5.4.4. ESTIMATED CLOUD OPERATING COST

Let $NC_{i,d}^{max}$ be the maximum network cost possible for each task type i at data center d, which can be calculated as:

$$NC_{i,d}^{max} = N^{price} \cdot NN_d \cdot S_i. \tag{65}$$

Let $NC_{i,d}^{E}(AR, \tau)$ be the estimated network cost possible for each task type i at data center d, which can be calculated as:

$$NC_{i,d}^{est}(AR, \tau) = \frac{NC_{i,d}^{max} \cdot AR_{i,d}(\tau)}{ER_{i,d}(\tau)}. \tag{66}$$

Observe that, at data center d for each task type i, $NC_{i,d}^{est}(AR, \tau)$ in (66) will increase to $NC_{i,d}^{max}$ as the data center level arrival rate $AR_{i,d}(\tau)$ approaches to its maximum execution rate $ER_{i,d}(\tau)$. Thus, we can say that $NC_{i,d}^{est}(AR, \tau)$ is a function of AR and τ.

Then the estimated data center cost $DC_{i,d}^{est}(AR, \tau)$ incurred by the task type i with a data center maximum execution rate $ER_{i,d}(\tau)$ at data center d, can be calculated by modifying (59) as:

$$DC_{i,d}^{est}(AR, \tau) = E_d^{price}(\tau) \cdot \alpha_d \cdot DP_{i,d}^{est}(AR, \tau) + \Delta_d^{peak}(\tau) + NC_{i,d}^{est}(AR, \tau) \tag{67}$$

where α_d= 1 if $PD_{i,d}^{est}(AR, \tau)$ is positive and $0 \leq \alpha_d \leq 1$ otherwise.

The estimated cloud operating costs incurred for the task type i, $CCT_i^{est}(AR, \tau)$, can be calculated as:

$$CCT_i^{est}(AR, \tau) = \sum_{d=1}^{|D|} DC_{i,d}^{est}(AR, \tau) \tag{68}$$

Our (second) optimization goal is to minimize the estimated cloud operating costs incurred for all task types, $CC^{est}(AR, \tau)$, can be calculated as:

$$CC^{est}(AR,\tau) = \sum_{i=1}^{|I|} CCT_i^{est}(AR,\tau) \tag{69}$$

subject to the constraints defined by (1) and (53).

5.5. OUR APPROACH

In this section, we first formulate the cloud workload distribution problem as the Nash equilibrium-based non-cooperative game and then combine it with DRL to design a new intelligent game-theoretic DRL (GT-DRL). In our workload management problem, the goal is to split the cloud level arrival rate $CAR_i(\tau)$ for each task type i into local data center level arrival rates $AR_{i,d}(\tau)$ and assign them to each data center $d \in D$. The following subsections describe the components of our technique and its architecture in more detail.

5.5.1. GAME THEORETIC COMPONENT

We model our cloud workload management problem as a non-cooperative game played among a set of players. Figure 37 shows the architecture of the game-theoretic component, where the dotted yellow boxes show the elements of our non-cooperative game. These elements are:

- **Players:** a finite set of players, where each task type i is a player; $i = 1,2,\dots,|I|$ (as per Section 5.3.2)
- **Strategy sets:** load distribution strategy of a player $AR_i = \{AR_{i,1}, AR_{i,2}, \dots, AR_{i,|D|}\}$
- **Rewards:** the goal of each player i is to minimize rewards associated with its strategy, i.e., the cloud carbon emissions incurred $CET_i^{est}(AR,\tau)$ calculated using (63), or cloud operating costs incurred $CCT_i^{est}(AR,\tau)$ calculated using (68) associated with its strategy, subject to the constraints defined by (1) and (53).

The game described here can be solved using a Nash equilibrium to obtain the workload distribution strategy for the cloud data centers. For minimizing carbon emissions, the Nash equilibrium of our non-cooperative game is a load distribution strategy of the entire cloud $AR = \{AR_1, AR_2, \dots, AR_{|I|}\}$ such that for each player i:

$$AR_i \in \underset{AR_i}{argmin}\, CET_i^{est}(AR, \tau). \tag{70}$$

subject to the constraints defined by (1) and (53).

For minimizing cloud operating costs, the Nash equilibrium of our non-cooperative game is a load distribution strategy of the entire cloud $AR = \{AR_1, AR_2, \dots, AR_{|I|}\}$ such that for each player i:

$$AR_i \in \underset{AR_i}{argmin}\, CCT_i^{est}(AR, \tau). \tag{71}$$

subject to the constraints defined by (1) and (53).

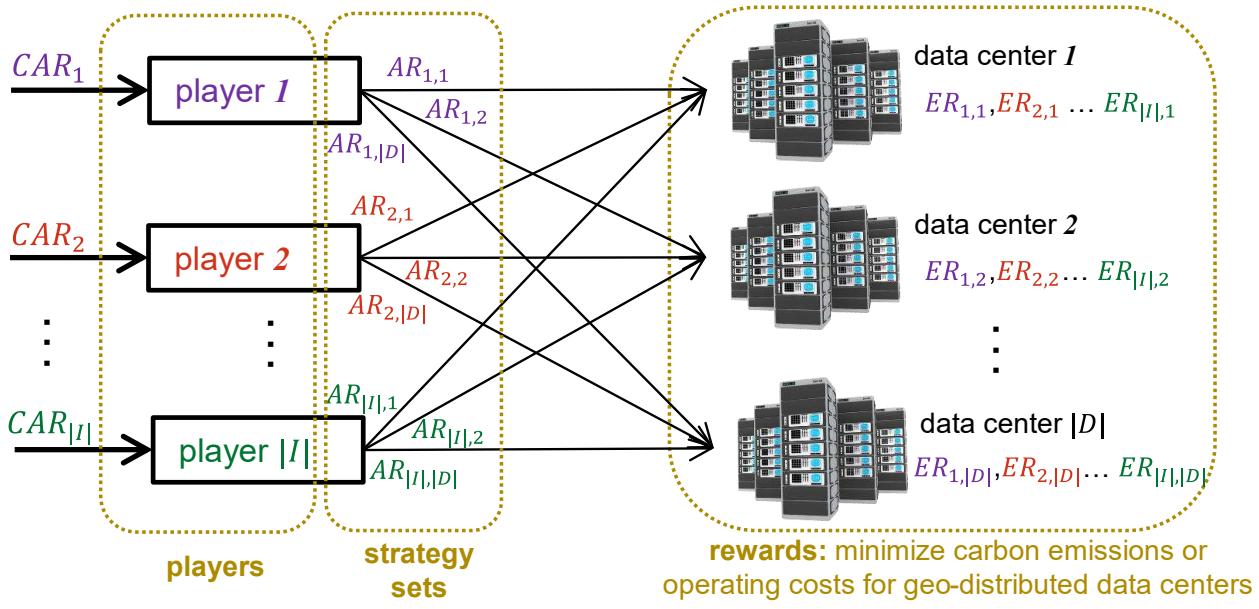


Figure 37: Game-theoretic component of our framework

If no player can further minimize the reward (emissions or costs) incurred by changing its current strategy to another one, the strategy AR is a Nash equilibrium. In this equilibrium, a player i cannot decrease the overall reward incurred by choosing a different workload distribution strategy AR_i given the other players' workload distribution strategies. Note that the goal of our approach is to find a workload distribution strategy for the entire cloud $AR = \{AR_1, AR_2, \dots, AR_{|I|}\}$ where each player i determines its own workload distribution strategy $AR_i = \{AR_{i,1}, AR_{i,2}, \dots, AR_{i,|D|}\}$ that satisfies equations (70) or (71). To find these strategies, our previous implementation of the Nash equilibrium-based game-theoretic technique used the "Best-Reply" algorithm [137], which is a mathematical algorithm that often fails to achieve global optimum solutions. To overcome this issue and improve the quality of our optimization approach, we use DRL for each player i to determine its workload distribution strategy AR_i.

5.5.2. DRL COMPONENT

DRL uses DNNs with several layers to explore a problem space. DNNs are highly suited for analyzing complex problems and high-dimensional information because they can autonomously learn hierarchical features. Our cloud workload scheduling problem for heterogeneous geo-distributed data centers is complex. Therefore, we consider a DRL-based method to discover an optimal workload distribution strategy automatically. Since DRL agents can sample from an environment or simulator, they need a reward function as input. As a result, they can interact with the environment, receive rewards or penalties for their actions, and learn optimal policy through trial and error.

Figure 38 displays the structure of the DRL component of our framework. The game-theoretic component of our framework uses DRL to find workload distribution strategy $AR_i =$

$\{AR_{i,1}, AR_{i,2}, \ldots, AR_{i,|D|}\}$ for each player i (as explained in Section 5.5.1). Below are the major elements of the DRL component.

- **Agent:** decision maker and learner, i.e., cloud workload manager who makes task placement decisions for a player or a task type i.
- **Environment:** information about geo-distributed data centers such as cloud level arrival rates for each task type, data center maximum execution rates for each task type at all data centers, maximum data center power consumptions, renewable power availability, carbon factors, electricity prices, net-metering factors, peak prices for all data center locations, and network costs for each task type at all data centers.
- **State:** state of the agent in the environment, i.e., current arrival rate distribution of a player/task type (i.e., the strategy set as described in Section 5.5.1)
- **Reward:** feedback value from the environment to assess the quality of the prior action, i.e., optimization objective of a player (i.e., reward as described in Section 5.5.1). The reward function calculates these values using information available in the environment. Typically, an agent tries to maximize the rewards; but our optimization goal is to minimize the rewards. Thus, we consider negative values.
- **Actions:** a set of $|D|$ desired fraction values. Each data center d gets a desired fraction $DF_{i,d}(\tau)$ of cloud level arrival rate $CAR_i(\tau)$ for each task type i. Thus, the local data center level arrival rate $AR_{i,d}(\tau)$ for a task type i at a data center d, can be calculated as,

$$AR_{i,d}(\tau) = DF_{i,d}(\tau) \cdot CAR_i(\tau)$$
$$\text{where } \sum_{d=1}^{|D|} DF_{i,d} = 1, \quad \forall i \in I \tag{72}$$

subject to the constraints defined by (1) and (53).

- **Policy:** The agent's strategy for determining the next course of action based on the current state.
- **Value function:** The predicted long-term reward with discount, in contrast to the short-term reward.
- **Action-value:** The long-term reward of the current state, taking action under a policy.

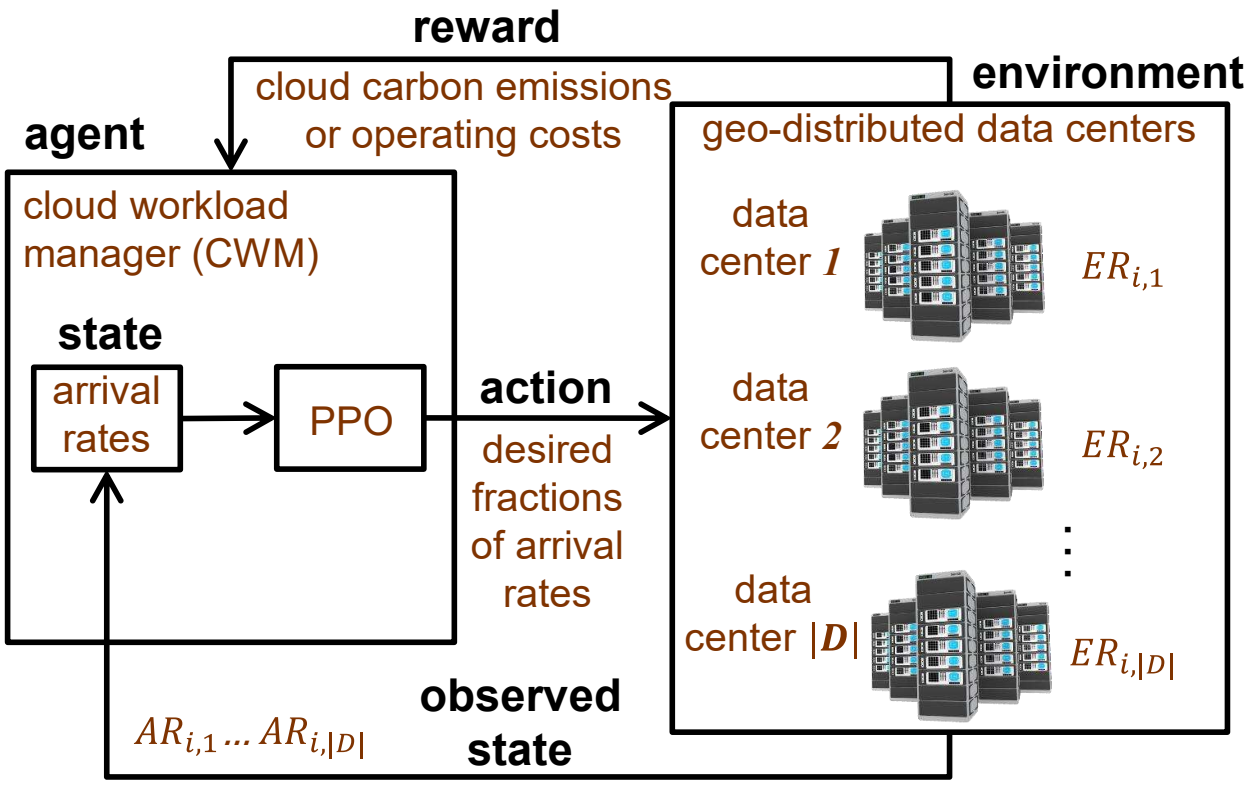


Figure 38: DRL component of our framework

5.5.3. GT-DRL FRAMEWORK

We adapt the Proximal Policy Optimization (PPO) DRL algorithm in our framework. It is an on-policy method that is designed to address the challenges of other policy optimization methods, such as the difficulty of choosing a good step size and poor sample efficiency. It is a type of actor-critic method where the critic's learning is used to benefit the actor. A value function that

the critic learns is used to calculate the expected return (sum of future rewards) from a given state. These estimates are used by the actor to modify its policy. In our proposed GT-DRL, a PPO-based DRL is combined with non-cooperative game theory to form a game-theoretic DRL, as shown in Figure 39.

Figure 39 shows the flow of our GT-DRL framework. Different colors represent different components. All yellow boxes are part of the PPO algorithm used by the agent. The agent interacts with the environment and receives the state and reward. It is an iterative optimization process that allows an agent to learn and improve its strategy for decision-making in an environment. It uses

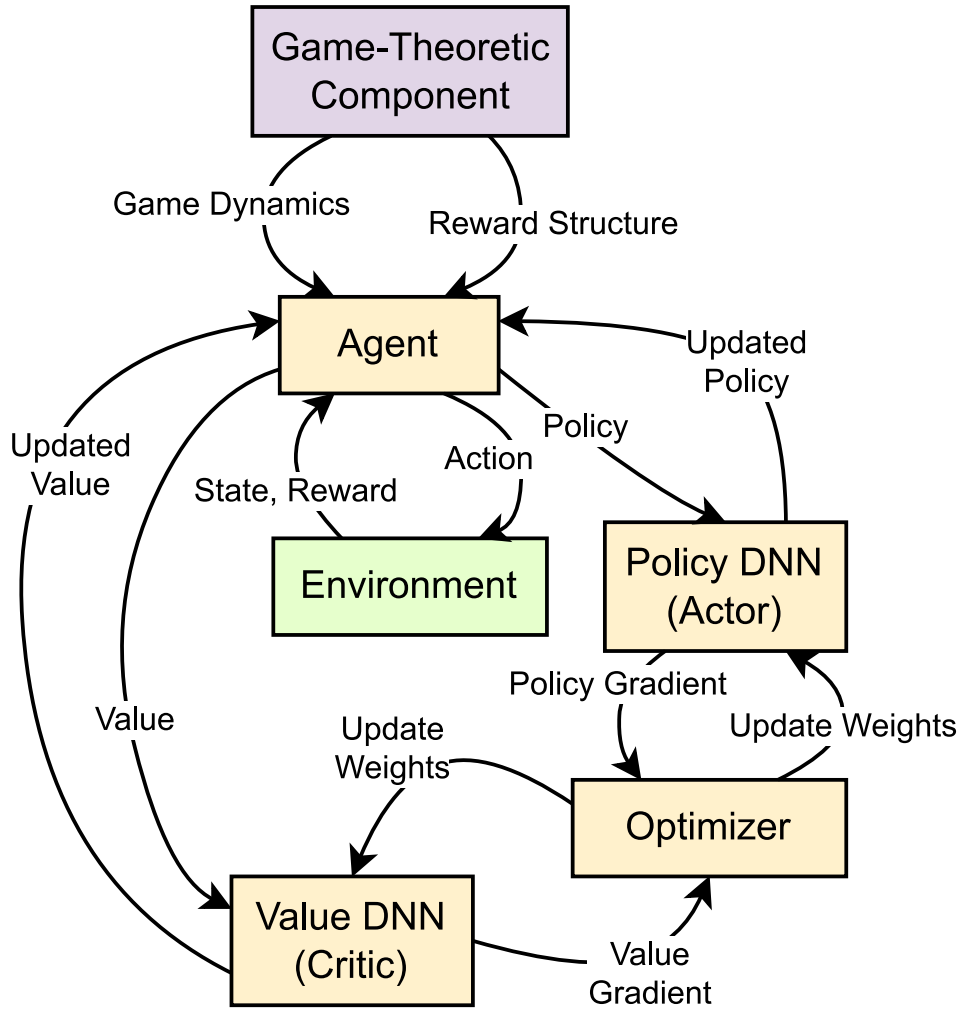


Figure 39: GT-DRL Framework

the policy DNN (actor) to decide on what action to take in the environment. It also utilizes the value DNN (critic) to estimate the value function of the current state. The policy gradient from the actor and the value gradient from the critic is evaluated and passed to the optimizer. The PPO algorithm refines and updates both the actor and the critic DNNs using the optimizer. The policy and the value function are improved based on the observed rewards and the next state from the environment. Then the updated/improved policy is then employed by the agent for the next interaction with the environment. The game-theoretic component (as described in Section 5.5.1) is used to apply game dynamics and to provide the reward structure to the agent so that the agent determines the optimal workload distribution strategy. The game-theory guides the exploration and improves the effectiveness of the policy. Breaking the complex problem of geo-distributed workload management into smaller chunks also makes PPO more efficient.

The DRL component of our GT-DRL finds the workload distribution strategy $AR_i = \{AR_{i,1}, AR_{i,2}, \dots, AR_{i,|D|}\}$ for each player i first and combines them to form a workload distribution strategy of the entire cloud $AR = \{AR_1, AR_2, \dots, AR_{|I|}\}$. On the other hand, if deployed directly in our geo-distributed environment, a conventional (non-game-theoretic) DRL would find the optimal workload distribution strategy of the entire cloud $AR = \{AR_{1,1}, \dots, AR_{i,d}, \dots, AR_{|i|,|D|}\}$ such that the cloud carbon emissions $CE^{est}(AR, \tau)$ calculated using (64) or cloud operating costs $CC^{est}(AR, \tau)$ calculated using (69) are minimized. In the conventional DRL, the action vector, i.e., the decision strategy AR involved in the optimization, has $|I|$ x $|D|$ variables. In our GT-DRL each player i minimizes the reward for its own strategy. Therefore, the action vector, i.e., the decision strategy AR_i involved in this optimization, has only $|D|$ variables. The non-cooperative game-theoretic nature of GT-DRL reduces the state and action vector variables from $|I|$ x $|D|$ values to

$|D|$ values. This greatly reduces the state-action space and makes the underlying DNN fast and lightweight.

5.6. SIMULATION ENVIRONMENT

We compare our proposed GT-DRL approach to mathematical (e.g., greedy, genetic), game-theoretic, and other DRL-based techniques. We implemented the DRL-based techniques using Python 3.8, OpenAI Gym [177], and PyTorch [178]. In addition, we used Stable Baseline 3 (SB3) [179], a set of reliable implementations of RL algorithms in PyTorch. It is a widely used, open-source, and well-tested RL library. We run the algorithm training in an offline environment because training in a real setup is too slow and often infeasible for large complex environment. This is a common practice in DRL-based optimization research [180], [172]. We train our DRL models with randomly sampled values of uniformly distributed set of arrival rates. All other environment variable values that we use are detailed in the subsequent portion of this section. We monitor the rewards and train the DRL models until either the rewards converge, or no further improvements are observed. The comparison techniques for geo-distributed data center management considered in our experimental analysis are summarized below.

(a) Force-Directed (FD): This is a greedy approach proposed in [42]. It is a variation of force-directed scheduling, frequently used for optimizing semiconductor logic synthesis. We adapt this approach to our environment and implement FD. It is an iterative method that selectively performs operations to minimize system forces until all constraints are met.

(b) Genetic Algorithm (GA): We adapt the GA from [107]. The Genitor style [69] GA has two parts: a genetic algorithm-based CWM and a local data center level greedy heuristic that is used to calculate the fitness value of the genetic algorithm. The local greedy heuristic has information

about task-node power dynamic voltage and frequency scaling (DVFS) models described in [107].

(c) Nash equilibrium (NASH): This technique, which was proposed in [137], integrates detailed models for data center power consumption, cost, and inter-data center network. This paper uses Nash equilibrium-based game-theoretic workload distribution techniques for geo-distributed data centers.

All the above methods (a), (b), and (c) are mathematical and need to be used with some workload prediction systems to avoid delays with workload allocation. This limitation does not apply to the DRL-based techniques because of their dynamic nature.

(d) Deep Deterministic Policy Gradient (DDPG): We do not compare our proposed technique with DQN-based prior works [154], [152], [155], [157], [160], [171] because DQN-based algorithms use discrete action space while our complex optimization problem uses continuous action space. Instead, we compare it against DDPG [158], which extends DQN to the continuous action space. We adapt the framework developed in [158] to our environment and build a DDPG-based DRL.

(e) Proximal Policy Optimization (PPO): PPO adjusts its policies less frequently and more cautiously. It adds a "clip" function to the policy objective function to avoid the policy altering too much with one update. We modify the PPO-based DRL proposed in [172] to our problem and environment.

Experiments were conducted for three geo-distributed data center configurations containing four, eight, and sixteen data centers. As shown in Figure 40, the locations of the data centers in the three configurations were selected from major cities around the continental United States to provide a variety of wind and solar conditions among sites and at various times of the day. The locations of each configuration were selected so that each configuration would have an even east-

coast to west-coast distribution to exploit better TOU pricing, peak demand pricing, net metering, renewable power, and carbon factor. Each data center comprises 4,320 nodes arranged in four aisles and is heterogeneous, with nodes from two or three node types, with most locations having three node types. We use Intel Xeon E3-1225v3 (4 core), Xeon E5649 (6 core), and Xeon E5-2697v2 (12 core) processor-based nodes.

Figure 40: Location of simulated data centers

The time of each epoch τ was set to be one hour. We limit our mathematical techniques (FD, GA, and NASH) to run for a maximum of one hour (epoch length). The DRL-based techniques run in a few seconds and can be deployed at run-time because of their dynamic nature. Carbon factor values used in the experiments are taken from EIA [176]. The network prices used in the experiments were taken from Amazon Web Services [112]. The real-world TOU electricity prices and the peak demand prices were taken directly from the utility companies. We calculated the renewable power at each data center location using solar and wind data from the National Solar

Radiation Database [63]. For all experiments, we used the renewable power calculated for the month of June. Figure 41 depicts the renewable power available at four locations with the yellow bar plots. The left y-axis is normalized to the highest renewable power available. The net metering factor α_d is 1 at most locations; it is less than 1 in rare cases; and it is 0 in some cases, i.e., net metering is not available at that location [37].

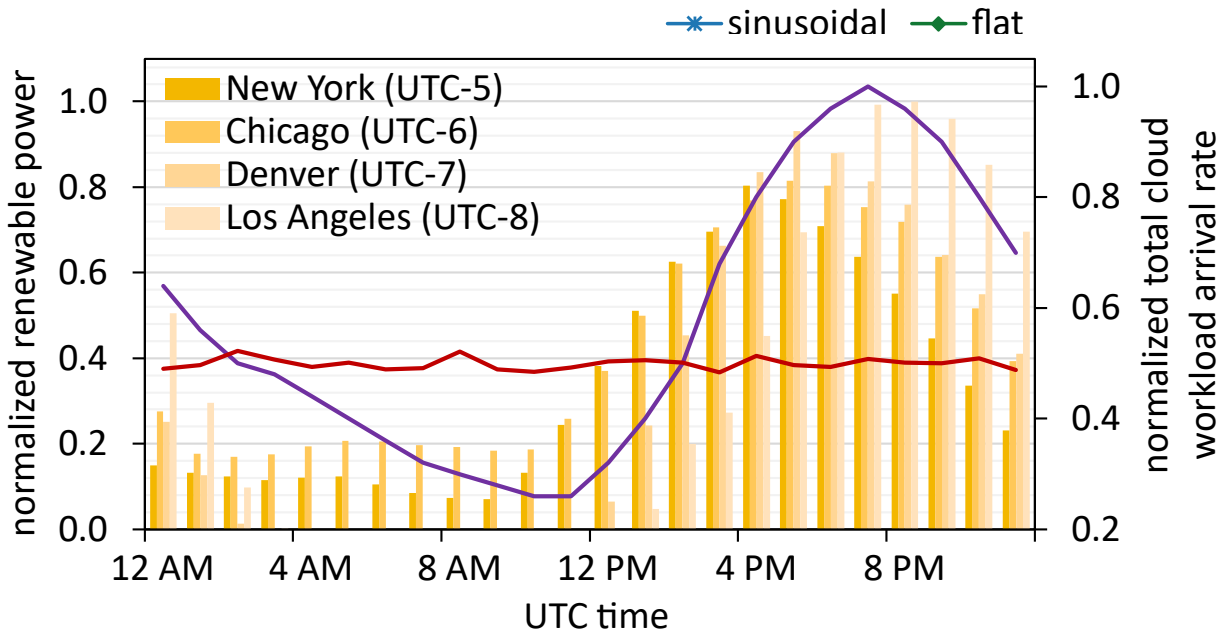


Figure 41: Renewable power and workload arrival rate at four data centers

Organizations are heavily using AI workloads [181]. Due to the popularity of such workloads among cloud providers, we use data-intensive AI workloads from the AIBench [175] benchmark suite. Table 11 summarizes the task types, ten in total, from this suite that we consider in our work. Task execution times and co-located performance data for tasks of the different memory intensity classes were obtained from running the benchmark applications on the three heterogeneous nodes. We considered two different workload arrival rate patterns: sinusoidal and flat. Figure 41 depicts a sinusoidal task arrival rate pattern and a flat task arrival rate pattern with

line plots. The right y-axis is normalized to the highest total cloud workload arrival rate. A sinusoidal task arrival rate pattern exists in environments where the workload traffic depends on consumer interaction and follows their demand during the day. However, for the environments where continuous computation is needed, and the workload pattern is non-user/consumer interaction specific, the task arrival rate pattern is usually flat (nearly constant). We conducted five simulation runs for every experiment with different workload arrival patterns. For each run, the arrival rate values were randomly sampled (with a normal distribution). We used the original arrival rate values, shown in Figure 41, as the mean and employed 20% of the mean as the standard deviation.

Table 11: Task Types Used in Experiments

Benchmark	**Algorithm**	**Dataset**
Image Classification	ResNet50	ImageNet
Image Generation	WassersteinGAN	LSUN
Image-to-Text	Neural Image Caption Model	Microsoft COCO
Image-to-Image	CycleGAN	Cityscapes
Speech Recognition	DeepSpeech2	Librispeech
Face Embedding	Facenet	VGGFace2, LFW
3D Face Recognition	3D Face Model	Intellifusion
Video Prediction	Motion-Focused Predictive Model	Robot Pushing
Image Compression	Recurrent Neural Network	ImageNet
3D Object Reconstruction	Convolutional Encoder-Decoder Network	ShapeNetCore

5.7. EXPERIMENTS

5.7.1. CLOUD CARBON EMISSIONS COMPARISON

Our first set of experiments analyzes the cloud carbon emissions, the total amount of carbon emissions across all data centers in the cloud service provider's infrastructure, for each technique

described in Section 5.6. For this experiment, the optimization objective of all methods is to minimize cloud carbon emissions. This experiment used a data center configuration with four locations running a sinusoidal workload pattern. We conducted five simulation runs and analyzed each technique's cloud carbon emissions variation over a day by plotting the average with the standard error bars, as shown in Figure 42. The y-axis is normalized to the highest cloud operating cost.

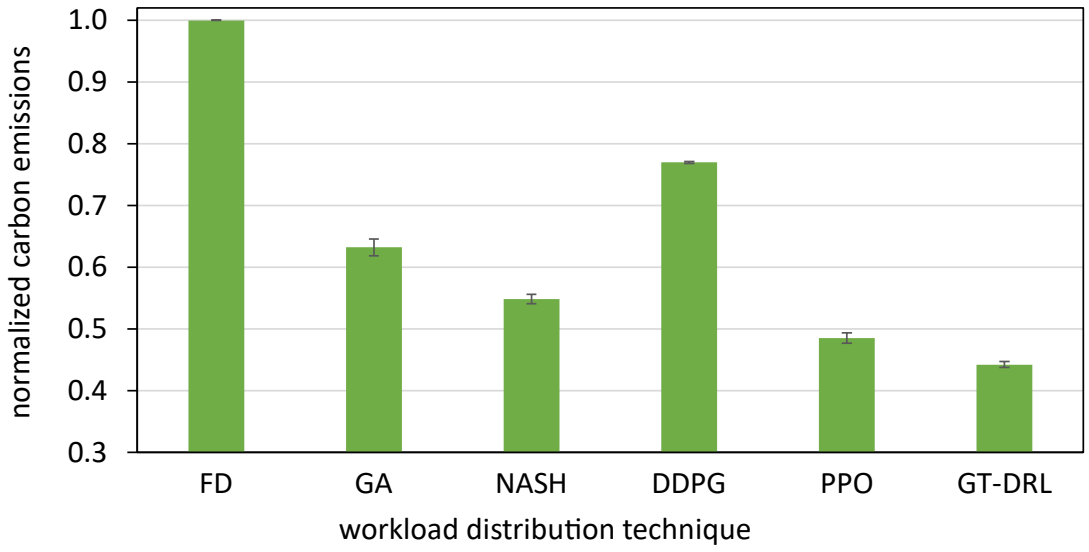


Figure 42: Cloud carbon emissions for each technique over a day for a configuration with four data centers

From Figure 42, it can be observed that the FD method performed the worst, severely over-provisioning nodes. Over-provisioning caused nodes to consume high power, increasing data center carbon emissions. The GA method has information about task-node power (DVFS) models [107], allowing it to make better task placement decisions. NASH performed the best among mathematical methods (FD, GA, and NASH). Because of the non-cooperative nature of this method [137], each player/task type determined the lowest possible carbon emissions with its workload allocation strategy.

It can be observed that the DDPG technique did not perform well among DRL methods (DDPG, PPO, and GT-DRL). DDPG depends on exploration to find the optimal policies. Our complex objective function optimization frequently demands a thorough search through the solution space, which is unsuitable for DDPG's exploration strategy. On the other hand, PPO urges the acting policy to adhere to the prior policy as closely as possible while permitting exploration within a specific range. Due to its improved sample efficiency, PPO is better suited to optimization problems like ours, where costly exploration is involved. Finally, GT-DRL performs the best because it allows multiple players to quickly optimize policy in a non-cooperative way, covering a broader solution space.

5.7.2. DATA CENTER SCALABILITY ANALYSIS

In this experiment, we analyze the impact of larger problem sizes. Five simulation runs were conducted with the sinusoidal workload patterns for 8 and 16 data center configurations and the previously discussed 4 data center configuration. Figure 43 shows the % cloud carbon emissions reduction of GT-DRL over each technique from each configuration, where the y-axis is the % cloud carbon emissions reduction.

As the number of data centers in the group grows, the problem size increases and becomes more complex. Therefore, the results show that cloud carbon emissions reduction decrease with the increasing number of data centers. The large error bars for the GA indicate a higher degree of variability/sensitivity where GA did not produce consistent results and was unstable within the given time constraint (one hour). For DRL-based methods, the state and action space increase for large problems, increasing the problem complexity and making the DRL-based methods difficult to converge. For our GT-DRL, the rise in problem space makes it difficult for players to find

equilibrium points or optimal strategies that balance the actions of different players to achieve the common objective, i.e., to minimize the overall cloud carbon emissions. These experiments confirm that our technique, GT-DRL, consistently performed the best for all problem sizes. The results from Figure 43 shows that, on average, GT-DRL achieved up to 55%, 32%, 21%, 43%, and 5% better cloud carbon emissions savings than FD [42], GA [107], NASH [137], DDPG [158], and PPO [172] respectively.

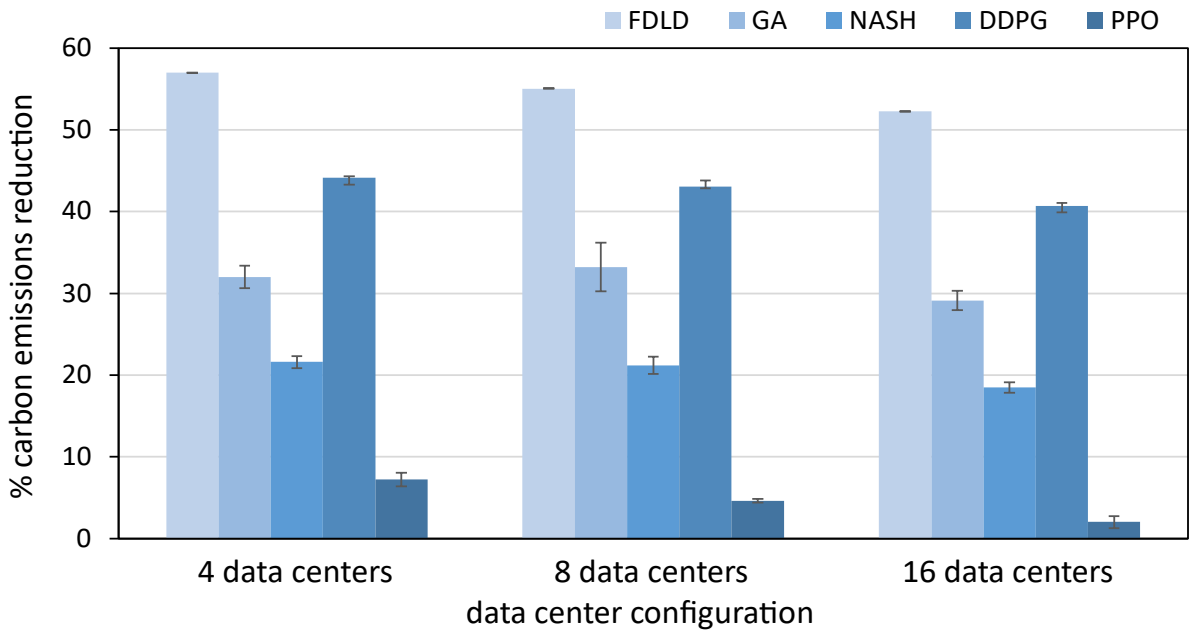
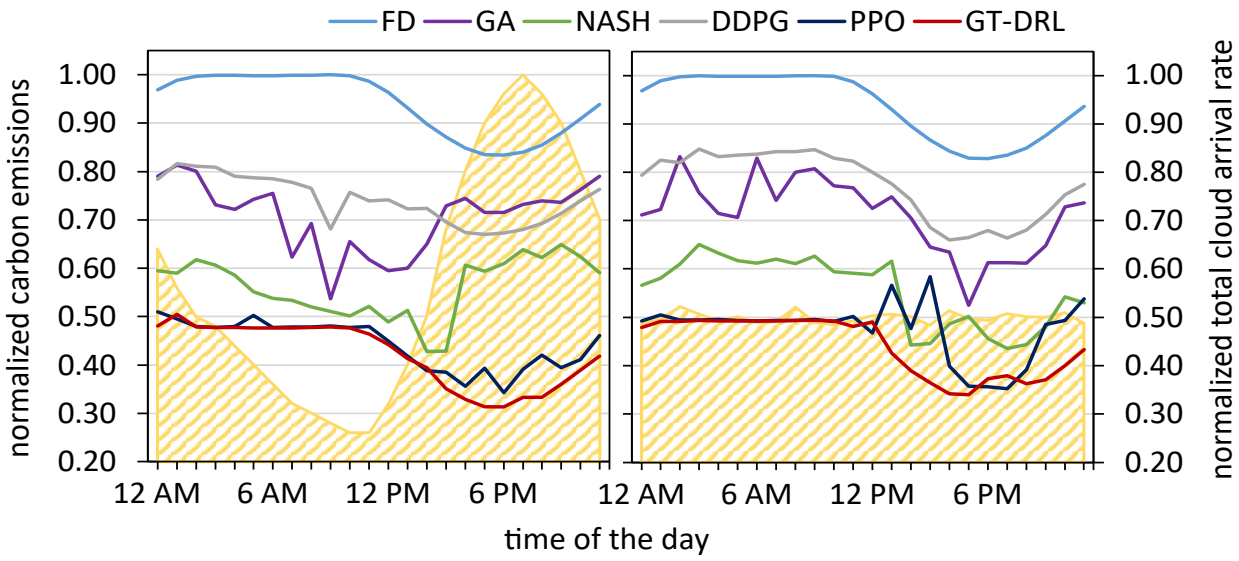


Figure 43: % cloud carbon emissions reduction of GT-DRL over each technique for a configuration with 4, 8, and 16 data centers

5.7.3. WORKLOAD ARRIVAL RATE PATTERN ANALYSIS

For this experiment, we considered a configuration with eight data centers executing both sinusoidal and flat workload arrival patterns. Figure 44(a) and Figure 44(b) show both arrival rate patterns (shaded yellow color). We conducted five simulation runs for each workload pattern. The experiment analyzes the cloud carbon emissions at each epoch (one-hour interval) over a day. The left y-axis is normalized to the highest carbon emitted at any epoch, and the right y-axis is

normalized to the highest total cloud workload arrival rate. Recall that each task type is characterized by its arrival rate and the estimated time required to complete the task on each heterogeneous compute node. The workload management techniques distribute the workload to minimize cloud carbon emissions across all data centers with the constraint that the execution rates of all task types meet their arrival rates (53). Therefore, the workload assignment was altered by changing the incoming arrival rate pattern, which further affected cloud carbon emissions. The results shown in Figure 44(a) and Figure 44(b) indicate that the geo-distributed system responded differently to sinusoidal and flat arrival rate patterns.



(a) sinusoidal workload arrival pattern (b) flat workload arrival pattern

Figure 44: Comparison of cloud carbon emissions among techniques over a day for (a) sinusoidal and (b) flat workload arrival rate patterns for eight data centers

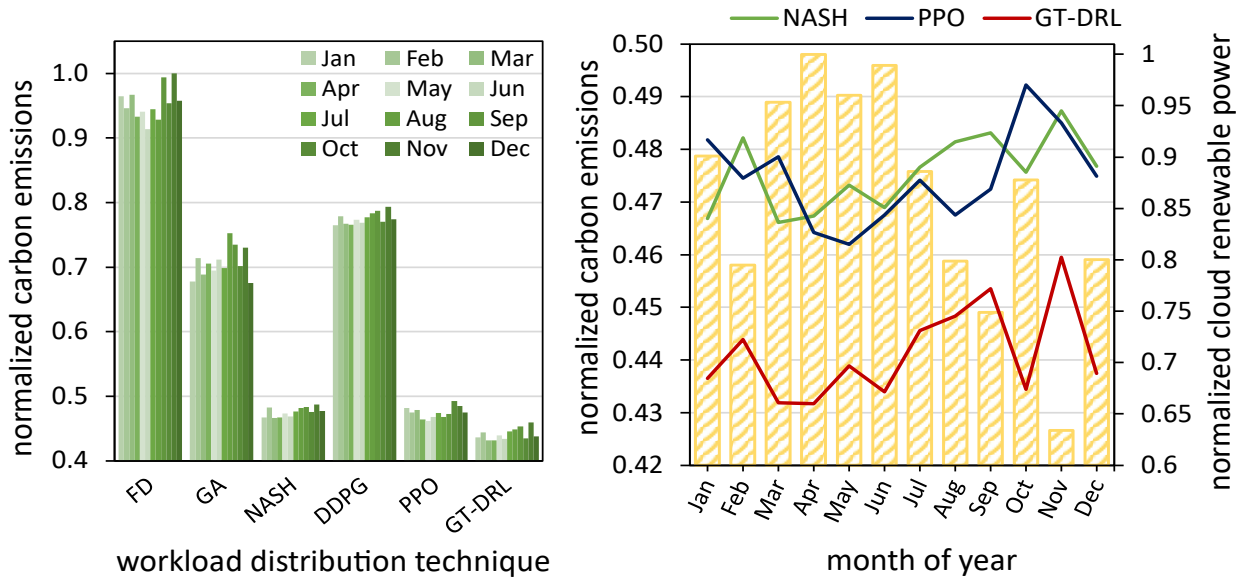
Overall, cloud carbon emissions were higher for the sinusoidal workload arrival pattern, where the elevated workload arrival rates in the second half of the day produced higher resource demand. This high resource demand resulted in higher power consumption causing more carbon emissions than the flat workload arrival pattern. For sinusoidal workload arrival in Figure 44(a),

the PPO-based DRL techniques performed better than all others during the high workload traffic period. The GT-DRL consistently performed the best for both arrival rate patterns during the day. This shows that our proposed technique allocated resources most effectively to reduce power consumption that reduced carbon emissions. For both arrival rate patterns, we see a dip in carbon emissions around 6 PM UTC because of the increased availability of renewable power (as shown in Figure 41) around that time. We analyze the effects of renewable power in the next section.

5.7.4. RENEWABLE POWER SENSITIVITY ANALYSIS

We performed another experiment to analyze the impact of the amount of renewable power on cloud carbon emissions. As mentioned in Section 5.3.3.4, carbon emissions from electricity consumption at each data center depend on renewable power generation. As a result, the emissions can fluctuate hourly, daily, monthly, and annually. This experiment considered all techniques for eight data centers executing a sinusoidal workload. We simulated 12 runs, where each run used different renewable power available at each month of the year. Results from Figure 45(a) show cloud carbon emissions for all techniques over each month of the year. The y-axis is normalized to the highest carbon emitted by any technique at any month of the year. Each column shows the amount of carbon emitted for a different month. Here, each technique followed the performance trend we discussed in our first experiment, where FD performed the worst, followed by DDPG and GA. GT-DRL performed the best, followed by PPO and NASH. However, NASH, PPO, and GT-DRL came close to each other. To closely analyze the performance difference and impact of those techniques for each month, we plotted them differently on a finer scale, as shown in Figure 45(b). The left y-axis is normalized to the highest carbon emitted at any month, and the right y-axis is normalized to the highest total renewable power available (shaded yellow bars) at all data

centers combined. The results from Figure 45(b) show that the cloud carbon emissions decreased with the increase in renewable power available. The plot showed that our proposed GT-DRL utilized the available renewable power most effectively and consistently performed the best for all months of the year.



(a) emissions for all techniques (b) emissions for NASH, PPO, and GT-DRL

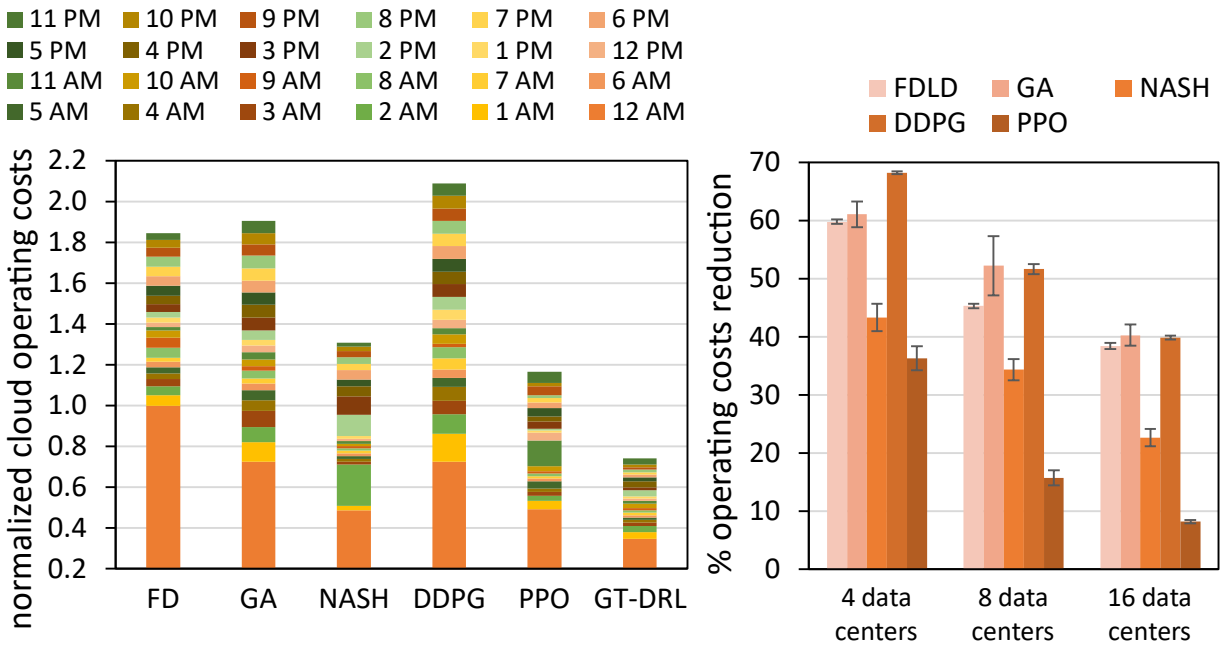
Figure 45: Impact of renewable power on cloud carbon emissions among techniques over a year for (a) all techniques and (b) NASH, PPO, and GT-DRL for eight data centers

5.7.5. CLOUD OPERATING COSTS COMPARISON

For each technique, this experiment analyzes the cloud operating costs, comprised of energy and network costs associated with inter-data center data transfers. The optimization objective for all methods in this experiment is to minimize cloud operating costs. This experiment used a data center configuration with four locations running a sinusoidal workload pattern. We conducted five simulation runs with different sinusoidal workload arrival patterns and analyzed the cloud operating costs that vary for each technique over a day. Results are shown in Figure 46(a), where each stacked column/block represents the cloud operating costs for an epoch. The column values

are normalized to the highest cost value at any epoch in the day. The results show that the costs for each method are very high during the first epoch (12 AM – 1 PM) because the period for which the results are shown represents the first day of the month where the initial peak demand cost is added. This effect stays there for the first day and would not be present for other days of the month. Although the first epoch costs are significant for FD, the DDPG performed the worst. Because of its off-policy nature, DDPG learns from different experiences than those it uses to make decisions. Such highly unstable learning and non-reassuring convergence made it challenging to obtain good solutions, resulting in high costs for most of the epochs. GA performed the worst among mathematical methods (FD, GA, and NASH). Similar to DDPG, the GA took a long time to explore the solution space and could not perform an adequate number of GA generations that could take place within the time limit (one hour by default). FD performed better than GA and DDPG. Because of its greedy nature, it completed many more iterations to explore the solution space. But it could not perform better than NASH and PPO-based DRL methods because it tends to get stuck in local minima [137]. Using its game-theoretic decision-making, NASH produced better results than DDPG, GA, and FD. PPO is an on-policy algorithm that prevents significant policy updates, ensuring stability and continuous improvement in the objective function. Having a better convergence makes PPO suitable for our geo-distributed workload optimization problem. Finally, our proposed technique, i.e., GT-DRL, performed better than other algorithms. In our proposed game-theoretic approach, the Nash equilibrium concept brought stability to the system by reaching states where no single player could unilaterally improve its performance (after reaching equilibrium). This resulted in a more robust solution in fluctuating geo-distributed workload scheduling environments.

Figure 46(b) shows the scalability analysis in terms of the % cloud operating costs reduction of GT-DRL over each technique from each configuration, where the y-axis is the % cloud operating costs reduction. The results show that cloud operating cost reduction decrease with the increasing number of data centers because of the increasing complexity of the problem. The GA performed the worst for 8 and 16 data centers. As the problem complexity increases, the number of GA generations that can take place within the time limit (one hour by default) decreases, which decreases the performance of GA. GT-DRL performed the best overall. The results from Figure 46(b) show that, on average, GT-DRL achieved up to 48%, 51%, 34%, 53%, and 20% better cloud operating costs savings than FD [42], GA [107], NASH [137], DDPG [158], and PPO [172] respectively.



(a) comparison with prior work (b) scalability analysis

Figure 46: (a) Epoch-based cloud operating costs for each technique over a day for a configuration with four data centers and (b) % cloud operating costs reduction of GT-DRL over each technique for a configuration with 4, 8, and 16 data centers

5.8. CONCLUSIONS

This chapter tackles the critical challenge of carbon emissions and operating costs in cloud computing by proposing an innovative optimization method for geo-distributed cloud data centers. The significant contribution lies in developing a new game-theoretic DRL, GT-DRL. Our proposed approach incorporates game-theoretic principles to achieve optimal workload distribution and resource allocation in a non-cooperative manner. The mathematical system model presented in this chapter encompasses various aspects, including workload arrival rates, data center execution rates, carbon emissions, renewable energy sources, data center power consumption, electricity costs, and network costs. The model extensively represents real-world factors that influence cloud data centers' efficiency and environmental impact. Unlike mathematical optimization techniques [42], [107], and [137], our approach works without any workload prediction system because of its dynamic nature.

The experimental analysis provides a comprehensive evaluation of the proposed game-theoretic DRL method as compared to recent techniques like FD [42], GA [107], NASH [137], DDPG [158], and PPO [172]. Through elaborate simulations involving diverse scenarios, such as different numbers of data centers, workload patterns, and levels of renewable energy, the proposed method consistently exhibits superior efficiency. Notably, GT-DRL is adaptable and effective in variable workload characteristics. Unlike FD [42], GA [107], and NASH [137], one of the critical strengths of the proposed method is its adeptness in utilizing renewable energy sources to a significant effect. Consequently, it intelligently reduces carbon emissions, marking it an environmentally sustainable solution. By combining the rapidness of a non-cooperative optimization strategy with the extraordinary exploration abilities of DRL, our proposed method allows for searching through a broad solution space, which makes it sample efficient and reaches

the solution faster than other state-of-the-art DRL techniques. GT-DRL achieved 55%, 32%, 21%, 43%, and 5% cloud carbon emissions reductions and up to 48%, 51%, 34%, 53%, and 20% better cloud operating costs savings on average than FD [42], GA [107], NASH [137], DDPG [158], and PPO [172] respectively. In conclusion, our game-theoretic DRL algorithm, GT-DRL, significantly advances workload optimization for geo-distributed cloud data centers, allowing for faster adaption to dynamic environments, helping in avoiding local optima by considering the interplay between multiple players, and guiding exploration in DRL leading to intelligent exploration strategies. By effectively reducing carbon emissions and operating costs, it addresses economic concerns and contributes positively to environmental sustainability.

6. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

6.1. RESEARCH CONCLUSIONS

This dissertation provides a comprehensive assessment of the fundamental challenges involved in managing geographically distributed data centers, specifically energy efficiency, carbon emission reduction, and operating cost optimization. The proposed solutions encompass a wide range of techniques, combining mathematical optimization, game theory, and machine learning to pave the way for intelligent and effective workload management in geo-distributed cloud computing.

The first chapter of this dissertation addressed the critical topic of reducing energy costs in geo-distributed data centers. We were able to create efficient workload management techniques by proposing a mathematical optimization framework that harnessed a plethora of elements such as net metering, peak shaving, and the effects of task co-location interference. This investigation resulted in a significant reduction in energy costs, with our solution surpassing previous ones by up to a 61% reduction. This significant reduction demonstrates the efficacy of our method, demonstrating the validity of comprehensive workload management strategies that address various aspects of geo-distributed data center operations.

Following that, we decided to add network costs to our calculations. Given the increase in data volumes and the geographic location of data centers, the cost of transferring workloads has become an important factor in optimizing cloud operations. To address this, we proposed a unique game theory-based workload management system that takes into account the costs of data transfer, queuing delays, and energy consumption all at once. Our methodology managed to minimize cloud operational expenses more efficiently than previous approaches, generating an average cost reduction of up to 43%.

As the research developed, we expanded our focus to ML-based methodologies. Recognizing the potential for ML to manage sophisticated, large-scale geo-distributed cloud environments more efficiently than traditional mathematical methods, we conducted a comprehensive survey of ML-based techniques used in geo-distributed cloud management. The investigation indicated that ML approaches greatly outperform traditional techniques, particularly in large and complex settings, making them an attractive route for further research and use in geo-distributed cloud management.

The culmination of this research journey is the invention of Game-Theoretic Deep Reinforcement Learning (GT-DRL), a unique technique that blends game theory principles and DRL. Non-cooperative game theory dynamics are integrated into a DRL framework in this technique, allowing data centers to make intelligent decisions about workload allocation while taking into account the heterogeneity of hardware resources, the dynamic nature of electricity prices, inter-data center data transfer costs, and carbon footprints. Our simulations demonstrated GT-DRL's superior performance, not only in terms of cloud operational cost savings but also in decreasing carbon emissions, establishing it as a key step toward environmentally friendly cloud computing. Our GT-DRL approach consistently outperformed existing techniques, resulting in significant reductions in carbon emissions and operating costs across a range of scenarios involving varying numbers of data centers, workload patterns, and renewable energy levels. Notably, when compared to state-of-the-art techniques, our approach lowered carbon emissions by up to 55% and boosted cloud operating cost savings by up to 53%.

Taken together, the findings of this dissertation reflect important advances in the management of geo-distributed cloud data centers. The research presented in this dissertation

emphasizes the importance of employing sophisticated workload management techniques that are responsive to the diverse, dynamic, and complex nature of these geo-distributed cloud environments. Furthermore, the exploration has opened up new paths for future research, with opportunities for refining and improving the proposed methodologies, extending the findings to other essential cloud workloads, and examining other developing technologies and techniques. All of these efforts are geared toward a more sustainable and cost-effective cloud computing future.

6.2. FUTURE RESEARCH DIRECTIONS

The techniques and methodologies described in this dissertation represent significant advances in the optimization of workload management for geo-distributed cloud data centers, revealing potential strategies for minimizing carbon emissions, energy, and network costs. While these discoveries represent significant progress, further investigation and development of existing frameworks are required for ongoing improvement and adaptation to dynamic computing landscapes. The next sections go into detail about specific recommendations for future research targeted at overcoming current limitations and expanding the scope of our study.

6.2.1. DEPLOYMENT IN OPERATIONAL CLOUD DATA CENTERS: BRIDGING THE GAP BETWEEN THEORY AND PRACTICE

So far, our study has mostly concentrated on mathematical modeling and simulation-based examination of our proposed cloud workload allocation technique. The true test of this technology will be how well it performs in real-world scenarios, particularly in operational cloud data centers specializing in serverless computing. Because of its event-driven nature and auto-scaling

capabilities, serverless computing poses new problems that necessitate more dynamic workload management solutions. As a result, future research could try to apply, test, and develop our proposed technique in such operational settings. Furthermore, case studies across diverse data center environments would be valuable in examining the effects of various workload characteristics, data center capabilities, and energy policies on the success of our technique.

6.2.2. ADAPTATION FOR CLOUD-NATIVE APPLICATIONS AND COMPLEX WORKFLOWS: HANDLING MODERN COMPUTING PARADIGMS

Cloud-native applications and complex workflows are progressively being adopted by modern cloud computing ecosystems. These applications are designed and optimized for cloud environments, allowing them to make full use of the cloud's flexibility and scalability. This study, however, did not dive into this topic. Future research could look into how we can adapt or extend our proposed technique to these applications. The ever-changing nature of these workloads, combined with the need for rapid scaling and dependable performance, needs unique workload management solutions. Future research could look into how containerization, microservices, and serverless architectures can be included in our system.

6.2.3. PERFORMANCE EVALUATION WITH DIVERSE ACCELERATORS: NAVIGATING HARDWARE HETEROGENEITY

To improve the performance of specialized workloads, modern data centers are populated with a broad range of accelerators such as Field-Programmable Gate Arrays (FPGAs), Graphics Processing Units (GPUs), and Smart Network Interface Cards (SmartNICs). These accelerators have distinct resource allocation and workload distribution challenges. Future research could look

into how such accelerators affect the performance of the suggested workload management technique. Such a study can help to improve energy consumption modeling and forecast, task scheduling algorithms, and our understanding of the trade-offs between performance, energy efficiency, and hardware usage.

6.2.4. EXPLORATION OF ADVANCED DRL ALGORITHMS: EMBRACING EVOLVING MACHINE LEARNING TECHNIQUES

Deep Reinforcement Learning (DRL) algorithms are constantly changing, with newer algorithms frequently outperforming older algorithms in terms of performance and efficiency. As a result, it is critical to investigate and incorporate these improvements into our proposed game-theoretic DRL framework. Future research could look at the use of more powerful DRL algorithms in our system, e.g., Trust Region Policy Optimization (TRPO), SAC (Soft Actor-Critic), TD3 (Twin Delayed Deep Deterministic Policy Gradient). Furthermore, different components of DRL training, such as reward structuring, state representation, and the balance between exploration and exploitation, might be thoroughly investigated further in order to find possible areas for improvement.

6.2.5. INTEGRATION OF MULTI-OBJECTIVE OPTIMIZATION TECHNIQUES: ADDRESSING THE DICHOTOMY OF PROVIDER AND CUSTOMER OBJECTIVES

The current research has focused on reducing carbon emissions and operational costs, especially from the standpoint of the cloud service provider. However, workload or resource management in geo-distributed data centers is a multifaceted topic, with cloud providers and

customers frequently having competing objectives. Customers are concerned about performance, reliability, cost-effectiveness, scalability, and security, while providers are concerned about operational efficiency, operating expenses, sustainability, regulatory compliance, and maximizing uptime. Future studies could look into incorporating multi-objective optimization approaches inside our proposed framework, taking both provider and customer objectives into account. This strategy could include the use of techniques like Pareto-based optimization, goal programming, or even multi-objective evolutionary algorithms. A strategy like this could lead the way for a more balanced solution that maximizes both provider and customer satisfaction.

BIBLIOGRAPHY

[1] "Key Internet Statistics in 2023 (Including Mobile)," [Online]. Available: https://www.broadbandsearch.net/blog/internet-statistics. [Accessed 1 June 2023].

[2] OpenAI, "AI and compute," [Online]. Available: https://openai.com/research/ai-and-compute. [Accessed 1 June 2023].

[3] "Global Data Center Accelerator Market Forecast to 2026: Artificial Intelligence to Drive the Growth of Cloud Data Center Market - ResearchAndMarkets.com," [Online]. Available: https://www.businesswire.com/news/home/20210819005361/en/Global-Data-Center-Accelerator-Market-Forecast-to-2026-Artificial-Intelligence-to-Drive-the-Growth-of-Cloud-Data-Center-Market---ResearchAndMarkets.com. [Accessed 1 June 2023].

[4] "Data Center Market by Component (Solution and Services), Type (Colocation, Hyperscale, Edge, and Others), Enterprise Size (Large Enterprises and Small & Medium Enterprises (SMEs)), and End User (BFSI, IT & Telecom, Government, Energy & Utilities, and Othe," [Online]. Available: https://www.alliedmarketresearch.com/data-center-market-A13117. [Accessed 1 June 2023].

[5] "What impact will IoT edge computing have on the data center market?," [Online]. Available: https://www.techradar.com/news/what-impact-will-iot-edge-computing-have-on-the-data-center-market. [Accessed 1 June 2023].

[6] "Data Center Market by Component, End-user, and Geography - Forecast and Analysis 2023-2027," Technavio, [Online]. Available: https://www.technavio.com/report/data-center-market-industry-analysis. [Accessed 1 June 2023].

[7] "What Is a Data Center?," [Online]. Available: https://www.fiberopticshare.com/what-is-a-data-center.html. [Accessed 1 June 2023].

[8] "Different Types Of Data Centers with Examples," [Online]. Available: https://www.colocationamerica.com/blog/different-types-of-data-centers. [Accessed 1 June 2023].

[9] "Types of Data Centers: Enterprise, Colocation, Hyperscale," [Online]. Available: https://dgtlinfra.com/types-of-data-centers/. [Accessed 1 June 2023].

[10] Atomia, "Comparing the geographical coverage of AWS, Azure and Google Cloud," [Online]. Available: https://www.atomia.com/2016/11/24/comparing-the-geographical-coverage-of-aws-azure-and-google-cloud/. [Accessed 25 June 2023].

[11] Wolford, Ben, "What is GDPR, the EU's new data protection law?," [Online]. Available: https://gdpr.eu/what-is-gdpr/. [Accessed 25 June 2023].

[12] F. Farahnakian, P. Liljeberg and J. Plosila, "Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning," in *2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2014.

[13] R. Appuswamy, M. Karpathiotakis, D. Porobic, A. Ailamaki, É. Polytechnique and F. De Lausanne, "The Case For Heterogeneous HTAP," in *8th Biennial Conference on Innovative Data Systems Research (CIDR '17)*, 2017.

[14] Datacentremagazine.com, "Energy efficiency predictions for data centres in 2023," [Online]. Available: https://datacentremagazine.com/articles/efficiency-to-loom-large-for-data-centre-industry-in-2023. [Accessed 25 June 2023].

[15] "What Makes Hyperscale, Hyperscale?," [Online]. Available: https://www.aflhyperscale.com/articles/what-makes-hyperscale-hyperscale/. [Accessed 1 June 2023].

[16] "IEA (2022) Report, Data Centres and Data Transmission Networks," [Online]. Available: https://www.iea.org/reports/data-centres-and-data-transmission-networks. [Accessed 1 June 2023].

[17] "Silicon Heatwave: The Looming Change in Data Center Climates," [Online]. Available: https://uptimeinstitute.com/resources/research-and-reports/silicon-heatwave-the-looming-change-in-data-center-climates. [Accessed 1 June 2023].

[18] "Meta 2021 Sustainability Report," [Online]. Available: https://sustainability.fb.com/2021-sustainability-report/. [Accessed 1 June 2023].

[19] D. Patterson, J. Gonzalez, U. Hölzle, Q. H. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier and J. Dean, "The carbon footprint of machine Learning training will plateau, then shrink," arXiv, 2022.

[20] "Data centers keep energy use steady despite big growth," [Online]. Available: https://www.dw.com/en/data-centers-energy-consumption-steady-despite-big-growth-because-of-increasing-efficiency/a-60444548. [Accessed 1 June 2023].

[21] Codefari.com, "Microsoft Azure," [Online]. Available: https://azure.codefari.com/2018/12/microsoft-azure-regions-and-datacenters.html. [Accessed 25 June 2023].

[22] A. S. G. Andrae, "New perspectives on internet electricity use in 2030," *Engineering and Applied Science Letter,* vol. 3, p. 19–31, 2020.

[23] "Data Center Locations," [Online]. Available: http://www.google.com/about/datacenters/inside/locations/index.html. [Accessed 1 June 2017].

[24] "Global Infrastructure," [Online]. Available: http://aws.amazon.com/about-aws/global-infrastructure/. [Accessed 1 June 2017].

[25] Y. Li, H. Wang, J. Dong, J. Li and S. Cheng, "Operating Cost Reduction for Distributed Internet Data Centers," in *13th IEEE/ACM Int'l Symposium on Cluster, Cloud, and Grid Computing*, 2013.

[26] "What is time-of-use pricing and why is it important?," [Online]. Available: http://www.energy-exchange.net/time-of-use-pricing/. [Accessed 1 June 2017].

[27] "Dynamic pricing," [Online]. Available: http://whatis.techtarget.com/definition/dynamic-pricing. [Accessed 1 June 2017].

[28] "Demand Charges," [Online]. Available: http://www.stem.com/resources/learning/. [Accessed 1 June 2017].

[29] "Understanding Peak Demand Charges," [Online]. Available: https://energysmart.enernoc.com/understanding-peak-demand-charges. [Accessed 1 June 2017].

[30] C. Hsu, "Rack PDU for Green Data Centers," in *Data Center Handbook*, H. Geng, Ed., John Wiley and Sons, Inc., Hoboken, NJ, Nov. 2014.

[31] Pacific Gas and Electric Company, "ELECTRIC SCHEDULE E-19," [Online]. Available: http://www.pge.com/tariffs/tm2/pdf/ELEC_SCHEDS_E-19.pdf. [Accessed 1 June 2017].

[32] "Apple to build a 3rd massive solar panel farm in North Carolina," [Online]. Available: https://gigaom.com/2014/07/08/apple-to-build-a-3rd-massive-solar-panel-farm-in-north-carolina/. [Accessed 1 June 2017].

[33] "Solar energy project at McGraw-Hill site recently completed," [Online]. Available: http://www.nj.com/mercer/index.ssf/2012/01/solar_energy_project_at_mcgraw.html/. [Accessed 1 June 2017].

[34] "Green power: Accelerating the transition to a clean energy future," [Online]. Available: https://www.microsoft.com/about/csr/environment/renewable_energy/. [Accessed 1 June 2017].

[35] "Achieving Our 100% Renewable Energy Purchasing Goal and Going Beyond," [Online]. Available: https://static.googleusercontent.com/media/www.google.com/en//green/pdf/achieving-100-renewable-energy-purchasing-goal.pdf. [Accessed 1 June 2017].

[36] "Facebook's Altoona, Iowa data center to be completely wind-powered," [Online]. Available: https://www.slashgear.com/facebooks-altoona-iowa-data-center-to-be-completely-wind-powered-13305335/. [Accessed 1 June 2017].

[37] "Net Metering," [Online]. Available: http://freeingthegrid.org/. [Accessed 1 June 2017].

[38] I. Goiri, R. Beauchea, K. Le, T. D. Nguyen, M. E. Haque, J. Guitart, J. Torres and R. Bianchini, "GreenSlot: Scheduling energy consumption in green datacenters," in *Int'l Conf. for High Performance Computing, Networking, Storage and Analysis*, 2011.

[39] C. Ren, D. Wang, B. Urgaonkar and A. Sivasubramaniam, "Carbon-Aware Energy Capacity Planning for Datacenters," in *IEEE 20th Int'l Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2012.

[40] I. Goiri, K. Le, T. D. Nguyen, J. Guitart, J. Torres and R. Bianchini, "GreenHadoop: Leveraging Green Energy in Data-processing Frameworks," in *7th ACM European Conf. on Computer Systems (EuroSys '12)*, Bern, 2012.

[41] I. Goiri, W. Katsak, K. Le, T. D. Nguyen and R. Bianchini, "Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy," in *18th Int'l Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS '13)*, Houston, Mar. 2013.

[42] H. Goudarzi and M. Pedram, "Geographical Load Balancing for Online Service Applications in Distributed Datacenters," in *IEEE 6th Int'l Conf. on Cloud Computing (CLOUD '13)*, June 2013.

[43] E. Jonardi, M. A. Oxley, S. Pasricha, A. A. Maciejewski and H. J. Siegel, "Energy cost optimization for geographically distributed heterogeneous data centers," in *6th Int'l Green and Sustainable Computing Conf. (IGSC '15)*, 2015.

[44] A. Wierman, Z. Liu, I. Liu and H. Mohsenian-Rad, "Opportunities and challenges for data center demand response," in *Int'l Green Computing Conf.*, 2014.

[45] L. Gu, D. Zeng, S. Guo and B. Ye, "Joint optimization of VM placement and request distribution for electricity cost cut in geo-distributed data centers," in *Int'l Conf. on Computing, Networking and Communications (ICNC '15)*, Feb. 2015.

[46] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang and F. C. M. Lau, "Dynamic pricing and profit maximization for the cloud with geo-distributed data centers," in *IEEE Conf. on Computer Communications (INFOCOM '14)*, 2014.

[47] L. Gu, D. Zeng, A. Barnawi, S. Guo and I. Stojmenovic, "Optimal Task Placement with QoS Constraints in Geo-Distributed Data Centers Using DVFS," *IEEE Trans. on Computers,* vol. 64, pp. 2049-2059, July 2015.

[48] H. Xu, C. Feng and B. Li, "Temperature Aware Workload Managementin Geo-Distributed Data Centers," *IEEE Trans. on Parallel and Distributed Systems,* vol. 26, pp. 1743-1753, June 2015.

[49] M. Polverini, A. Cianfrani, S. Ren and A. V. Vasilakos, "Thermal-Aware Scheduling of Batch Jobs in Geographically Distributed Data Centers," *IEEE Trans. on Cloud Computing,* vol. 2, pp. 71-84, January 2014.

[50] D. Mehta, B. O'Sullivan and H. Simonis, "Energy Cost Management for Geographically Distributed Data Centres under Time-Variable Demands and Energy Prices," in *IEEE/ACM 6th Int'l Conf. on Utility and Cloud Computing*, 2013.

[51] Z. Abbasi, M. Pore and S. K. S. Gupta, "Impact of workload and renewable prediction on the value of geographical workload management," in *2nd Int'l Workshop on Energy Efficient Data Centers (E2DC '13)*, May 2013.

[52] C. Chen, B. He and X. Tang, "Green-aware workload scheduling in geographically distributed data centers," in *4th IEEE Int'l Conf. on Cloud Computing Technology and Science Proceedings*, 2012.

[53] P. Bodik, R. Griffith, C. Sutton, A. Fox, M. I. Jordan and D. A. Patterson, "Automatic Exploration of Datacenter Performance Regimes," in *1st Workshop on Automated Control for Datacenters and Clouds (ACDC '09)*, Barcelona, June 2009.

[54] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao and F. Zhao, "Energy-aware Server Provisioning and Load Dispatching for Connection-intensive Internet Services," in *5th USENIX Symposium on Networked Systems Design and Implementation (NSDI '08)*, San, Apr. 2008.

[55] D. G. Feitelson, D. Tsafrir and D. Krakov, "Experience with using the Parallel Workloads Archive," *J. of Parallel and Distributed Computing,* vol. 74, pp. 2967-2982, Oct. 2014.

[56] M. A. Oxley, E. Jonardi, S. Pasricha, A. A. Maciejewski, H. J. Siegel, P. J. Burns and G. A. Koenig, "Rate-based thermal, power, and co-location aware resource management for heterogeneous data centers," *J. of Parallel and Distributed Computing,* vol. 112, pp. 126-139, Feb 2018.

[57] V. Shestak, J. Smith, A. A. Maciejewski and H. J. Siegel, "Stochastic robustness metric and its use for static resource allocations," *J. of Parallel and Distributed Computing,* vol. 68, pp. 1157-1173, Aug. 2008.

[58] M. A. Iverson, F. Ozguner and L. Potter, "Statistical prediction of task execution times through analytic benchmarking for scheduling in a heterogeneous environment," *IEEE Trans. on Computers,* vol. 48, pp. 1374-1379, December 1999.

[59] H. Bhagwat, U. Singh, A. Deodhar, A. Singh and A. Sivasubramaniam, "Fast and accurate evaluation of cooling in data centers," *J. of Electronic Packaging,* vol. 137, Mar. 2015.

[60] "Thermal Guidelines for Data Processing Environments-Expanded Data Center Classes and Usage Guidance," *Technical report, American Society of Heating, Refrigerating, and Air-Conditioning Engineers, Inc.,* 2011.

[61] J. D. Moore, J. S. Chase, P. Ranganathan and R. K. Sharma, "Making Scheduling "Cool": Temperature-Aware Workload Placement in Data Centers," in *USENIX Annual Technical Conf. (ATEC '05)*, 2005.

[62] X. Deng, D. Wu, J. Shen and J. He, "Eco-Aware Online Power Management and Load Scheduling for Green Cloud Datacenters," *IEEE Systems Journal,* vol. 10, pp. 78-87, March 2016.

[63] NREL, "National Solar Radiation Database," [Online]. Available: https://mapsbeta.nrel.gov/nsrdb-viewer/. [Accessed 1 June 2017].

[64] S. Govindan, J. Liu, A. Kansal and A. Sivasubramaniam, "Cuanta: Quantifying Effects of Shared On-chip Resource Interference for Consolidated Virtual Machines," in *2nd ACM Symposium on Cloud Computing (SOCC '11)*, Cascais, Oct. 2011.

[65] D. Dauwe, E. Jonardi, R. D. Friese, S. Pasricha, A. A. Maciejewski, D. A. Bader and H. J. Siegel, "HPC node performance and energy modeling with the co-location of applications," *The J. of Supercomputing,* vol. 72, pp. 4771-4809, 01 December 2016.

[66] "PARSEC benchmark suite," [Online]. Available: http://parsec.cs.princeton.edu/index.htm/. [Accessed 1 June 2017].

[67] "NAS Parallel Benchmarks," [Online]. Available: https://www.nas.nasa.gov/publications/npb.html. [Accessed 1 June 2017].

[68] P. G. Paulin and J. P. Knight, "Force-directed scheduling for the behavioral synthesis of ASICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems,* vol. 8, pp. 661-679, June 1989.

[69] D. Whitley, "The GENITOR algorithm and selective pressure: Why rank-based allocation of reproductive trials is best," in *3rd lnternational Conf. on Genetic Algorithms*, June 1989.

[70] "Pyevolve: A Complete Genetic Algorithm Framework," [Online]. Available: http://pyevolve.sourceforge.net/0_6rc1/index.html. [Accessed 1 June 2017].

[71] T. D. D. Braun, H. J. J. Siegel, N. Beck, L. Bölöni, R. F. F. Freund, D. Hensgen, M. Maheswaran, A. I. I. Reuther, J. P. P. Robertson, M. D. D. Theys and B. Yao, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," *J. of Parallel and Distributed Computing,* vol. 61, pp. 810-837, June 2001.

[72] M. Maheswaran, S. Ali, H. J. J. Siegel, D. Hensgen and R. F. F. Freund, "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," *J. of Parallel and Distributed Computing,* vol. 59, pp. 107-131, Nov. 1999.

[73] C. Isci, S. McIntosh, J. Kephart, R. Das, J. Hanson, S. Piper, R. Wolford, T. Brey, R. Kantner, A. Ng, J. Norris, A. Traore and M. Frissora, "Agile, Efficient Virtualization Power Management with Low-latency Server Power States," in *40th Annual Int'l Symposium on Computer Architecture (ISCA '13)*, Tel-Aviv, Israel, June 2013.

[74] G. Cook, T. Dowdall, D. Pomerantz and Y. Wang, "Clicking clean: How companies are creating the green internet," *Greenpeace Inc., Washington, DC,* Apr. 2014.

[75] "Net Metering Policies," [Online]. Available: http://www.dsireusa.org/. [Accessed 1 June 2017].

[76] "Scryer: Netflix's Predictive Auto Scaling Engine," [Online]. Available: https://medium.com/netflix-techblog/scryer-netflixs-predictive-auto-scaling-engine-a3f8fc922270. [Accessed 1 June 2017].

[77] "Making Facebook's software infrastructure more energy efficient with Autoscale," [Online]. Available: https://code.facebook.com/posts/816473015039157/making-facebook-s-software-infrastructure-more-energy-efficient-with-autoscale. [Accessed 1 June 2017].

[78] R. N. Calheiros, E. Masoumi, R. Ranjan and R. Buyya, "Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS," *IEEE Trans. on Cloud Computing,* vol. 3, pp. 449-458, October 2015.

[79] J. Xue, F. Yan, R. Birke, L. Y. Chen, T. Scherer and E. Smirni, "PRACTISE: Robust prediction of data center time series," in *11th Int'l Conf. on Network and Service Management (CNSM)*, 2015.

[80] IEA (2017), "Digitalisation and Energy," 2017. [Online]. Available: https://www.iea.org/reports/digitalisation-and-energy. [Accessed 1 May 2020].

[81] W. B. Norton, "Internet Transit Prices - Historical and Projected," DrPeering white paper, [Online]. Available: http://drpeering.net/white-papers/Internet-Transit-Pricing-Historical-And-Projected.php.

[82] X. Yi, F. Liu, J. Liu and H. Jin, "Building a network highway for big data: architecture and challenges," *IEEE Network,* vol. 28, pp. 5-13, 2014.

[83] L. Belkhir and A. Elmeligi, "Assessing ICT global emissions footprint: Trends to 2040 & recommendations," *Journal of Cleaner Production,* vol. 177, p. 448–463, 2018.

[84] "Greenpeace: China's Data Centers on Track to Use More Energy than All of Australia," [Online]. Available: https://www.datacenterknowledge.com/asia-pacific/greenpeace-china-s-data-centers-track-use-more-energy-all-australia. [Accessed 1 May 2020].

[85] J. Kumar, D. Saxena, A. K. Singh and A. Mohan, "BiPhase adaptive learning-based neural network model for cloud datacenter workload forecasting," *Soft Computing,* p. 1–18, 2020.

[86] Z. Chen, J. Hu, G. Min, A. Y. Zomaya and T. El-Ghazawi, "Towards Accurate Prediction for High-Dimensional and Highly-Variable Cloud Workloads with Deep Learning," *IEEE Transactions on Parallel and Distributed Systems,* vol. 31, pp. 923-934, 2020.

[87] W. Wu, W. Wang, X. Fang, L. Junzhou and A. V. Vasilakos, "Electricity Price-aware Consolidation Algorithms for Time-sensitive VM Services in Cloud Systems," *IEEE Transactions on Services Computing,* pp. 1-1, 2019.

[88] M. Jawad, M. B. Qureshi, U. Khan, S. M. Ali, A. Mehmood, B. Khan, X. Wang and S. U. Khan, "A robust Optimization Technique for Energy Cost Minimization of Cloud Data Centers," *IEEE Transactions on Cloud Computing,* pp. 1-1, 2018.

[89] H. Yeganeh, A. Salahi and M. A. Pourmina, "A Novel Cost Optimization Method for Mobile Cloud Computing by Capacity Planning of Green Data Center With Dynamic Pricing," *Canadian Journal of Electrical and Computer Engineering,* vol. 42, pp. 41-51, 2019.

[90] M. Dabbagh, B. Hamdaoui and A. Rayes, "Peak Power Shaving for Reduced Electricity Costs in Cloud Data Centers: Opportunities and Challenges," *IEEE Network,* pp. 1-6, 2020.

[91] V. Papadopoulos, J. Knockaert, C. Develder and J. Desmet, "Peak Shaving through Battery Storage for Low-Voltage Enterprises with Peak Demand Pricing," *Energies,* vol. 13, p. 1183, 2020.

[92] Y. Peng, D.-K. Kang, F. Al-Hazemi and C.-H. Youn, "Energy and QoS aware resource allocation for heterogeneous sustainable cloud datacenters," *Optical Switching and Networking,* vol. 23, p. 225–240, 2017.

[93] J. Krzywda, V. Meyer, M. G. Xavier, A. Ali-Eldin, P.-O. Östberg, C. A. F. De Rose and E. Elmroth, "Modeling and Simulation of QoS-Aware Power Budgeting in Cloud Data Centers," 2019.

[94] Y. Mansouri and R. Buyya, "Dynamic replication and migration of data objects with hot-spot and cold-spot statuses across storage data centers," *Journal of Parallel and Distributed Computing,* vol. 126, p. 121–133, 2019.

[95] A. Hou, C. Q. Wu, R. Qiao, L. Zuo, M. M. Zhu, D. Fang, W. Nie and F. Chen, "QoS provisioning for various types of deadline-constrained bulk data transfers between data centers," *Future Generation Computer Systems,* vol. 105, p. 162–174, 2020.

[96] R. Prodan, E. Torre, J. J. Durillo, G. S. Aujla, N. Kummar, H. M. Fard and S. Benedikt, "Dynamic Multi-objective Virtual Machine Placement in Cloud Data Centers," in *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019.

[97] M. Najm and V. Tamarapalli, "VM Migration for Profit Maximization in Federated Cloud Data Centers," in *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2020.

[98] H. Zhang, Y. Xiao, S. Bu, R. Yu, D. Niyato and Z. Han, "Distributed Resource Allocation for Data Center Networks: A Hierarchical Game Approach," *IEEE Transactions on Cloud Computing,* pp. 1-1, 2018.

[99] X. Yuan, G. Min, L. T. Yang, Y. Ding and Q. Fang, "A game theory-based dynamic resource allocation strategy in geo-distributed datacenter clouds," *Future Generation Computer Systems,* vol. 76, p. 63–72, 2017.

[100] A. Yassine, A. A. N. Shirehjini and S. Shirmohammadi, "Bandwidth On-demand for Multimedia Big Data Transfer across Geo-Distributed Cloud Data Centers," *IEEE Transactions on Cloud Computing,* pp. 1-1, 2016.

[101] B. K. Ray, A. Saha and S. Roy, "Migration cost and profit oriented cloud federation formation: hedonic coalition game based approach," *Cluster Computing,* vol. 21, p. 1981–1999, 2018.

[102] Z. Yu, Y. Guo and M. Pan, "Coalitional datacenter energy cost optimization in electricity markets," in *Proceedings of the Eighth International Conference on Future Energy Systems*, 2017.

[103] S. Xu, L. Liu, L. Cui, X. Chang and H. Li, "Resource scheduling for energy-efficient in cloud-computing data centers," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International*

Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018.

[104] M. M. Moghaddam, M. H. Manshaei, W. Saad and M. Goudarzi, "On Data Center Demand Response: A Cloud Federation Approach," *IEEE Access,* vol. 7, p. 101829–101843, 2019.

[105] A. Ghassemi, P. Goudarzi, M. R. Mirsarraf and T. A. Gulliver, "Game based traffic exchange for green data center networks," *Journal of Communications and Networks,* vol. 20, p. 85–92, 2018.

[106] R. Tripathi, S. Vignesh, V. Tamarapalli, A. T. Chronopoulos and H. Siar, "Non-cooperative power and latency aware load balancing in distributed data centers," *Journal of Parallel and Distributed Computing,* vol. 107, p. 76–86, 2017.

[107] N. Hogade, S. Pasricha, H. J. Siegel, A. A. Maciejewski, M. A. Oxley and E. Jonardi, "Minimizing Energy Costs for Geographically Distributed Heterogeneous Data Centers," *IEEE Transactions on Sustainable Computing,* vol. 3, no. 4, pp. 318-331, 2018.

[108] "BigDataBench 5.0 benchmark suite," [Online]. Available: http://www.benchcouncil.org/BigDataBench/. [Accessed 1 May 2020].

[109] Z. Liu, M. Lin, A. Wierman, S. Low and L. L. H. Andrew, "Greening geographical load balancing," *IEEE/ACM Transactions on Networking,* vol. 23, p. 657–671, 2014.

[110] J. Hamilton, "The Cost of Latency," [Online]. Available: https://perspectives.mvdirona.com/2009/10/the-cost-of-latency/. [Accessed 1 May 2020].

[111] M. J. Osborne and A. Rubinstein, A course in game theory, MIT press, 1994.

[112] "Amazon CloudFront Pricing," [Online]. Available: https://aws.amazon.com/cloudfront/pricing/. [Accessed 1 May 2020].

[113] "Cloud Computing Trends: 2021 State of the Cloud Report," [Online]. Available: https://www.flexera.com/blog/cloud/cloud- computing-trends-2021-state-of-the-cloud-report/. [Accessed 20 May 2021].

[114] "Top cloud providers in 2021: AWS, Microsoft Azure, and Google Cloud, hybrid, SaaS players," [Online]. Available: https://www.zdnet.com/article/the-top-cloud-providers-of-2021-aws-microsoft-azure-google-cloud-hybrid-saas/. [Accessed 20 May 2021].

[115] "Digital Around the World," [Online]. Available: https://datareportal.com/global-digital-overview. [Accessed 20 Dec 2021].

[116] "Global Cloud Computing Market Outlook," [Online]. Available: https://www.businesswire.com/news/home/20211112005486/en/Global-947.3B-Cloud-Computing-Market-Outlook-2026-by-Service-Model-Deployment-Model-Organization-Size-Vertical-and-Region---ResearchAndMarkets.com. [Accessed 20 Dec 2021].

[117] "Gartner Global Cloud Revenue," [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-11-10-gartner-says-cloud-will-be-the-centerpiece-of-new-digital-experiences. [Accessed 20 Dec 2021].

[118] "Google Cloud Locations," [Online]. Available: https://cloud.google.com/about/locations. [Accessed 20 Dec 2021].

[119] "AWS Global Infrastructure," [Online]. Available: http://aws.amazon.com/about-aws/global-infrastructure/. [Accessed 20 Dec 2021].

[120] "Microsoft Teams Up With Accenture, Goldman on Greener Software," [Online]. Available: https://www.bloombergquint.com/business/microsoft-teams-up-with-accenture-goldman-on-greener-software. [Accessed 20 Dec 2021].

[121] "Greenpeace: China's Data Centers on Track to Use More Energy than All of Australia," [Online]. Available: https://www.datacenterknowledge.com/asia-pacific/greenpeace-china-s-data-centers-track-use-more-energy-all-australia. [Accessed 20 Dec 2021].

[122] "Amazon Found Every 100ms of Latency Cost them 1% in Sales," [Online]. Available: https://www.gigaspaces.com/blog/amazon-found-every-100ms-of-latency-cost-them-1-in-sales/.

[123] "Study Shines a Light on Big Tech's AI Investments," [Online]. Available: https://www.nextgov.com/emerging-tech/2021/04/study-shines-light-big-techs-ai-investments/173561/. [Accessed 20 Dec 2021].

[124] W. Khallouli and J. Huang, "Cluster resource scheduling in cloud computing: literature review and research challenges," *The Journal of Supercomputing,* p. 1–46, 2021.

[125] M. F. Manzoor, A. Abid, M. S. Farooq, N. A. Nawaz and U. Farooq, "Resource Allocation Techniques in Cloud Computing: A Review and Future Directions," *Elektronika ir Elektrotechnika,* vol. 26, p. 40–51, 2020.

[126] B. K. Dewangan, A. Agarwal, T. Choudhury, A. Pasricha and S. Chandra Satapathy, "Extensive review of cloud resource management techniques in industry 4.0: Issue and challenges," *Software: Practice and Experience,* vol. 51, p. 2373–2392, 2021.

[127] S. Jayaprakash, M. D. Nagarajan, R. P. d. Prado, S. Subramanian and P. B. Divakarachari, "A systematic review of energy management strategies for resource

allocation in the cloud: Clustering, optimization and machine learning," *Energies,* vol. 14, p. 5322, 2021.

[128] H. Kaur and K. Kaur, "Load Balancing and Its Challenges in Cloud Computing: A Review," *Information and Communication Technology for Competitive Strategies (ICTCS 2020),* p. 731–741, 2021.

[129] D. A. Shafiq, N. Z. Jhanjhi and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," *Journal of King Saud University-Computer and Information Sciences,* 2021.

[130] A. A. A. Alkhatib, A. Alsabbagh, R. Maraqa and S. Alzubi, "Load Balancing Techniques in Cloud Computing: Extensive Review," *Advances in Science, Technology and Engineering Systems Journal,* vol. 6, p. 860–870, 2021.

[131] S. S. George and R. S. Pramila, "A review of different techniques in cloud computing," *Materials Today: Proceedings,* vol. 46, pp. 8002-8008, 2021.

[132] G. Zhou, W. Tian and R. Buyya, "Deep Reinforcement Learning-based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions," *arXiv preprint arXiv:2105.04086,* 2021.

[133] T. L. Duc, R. G. Leiva, P. Casari and P.-O. Östberg, "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey," *ACM Computing Surveys (CSUR),* vol. 52, p. 1–39, 2019.

[134] V. N. Tsakalidou, P. Mitsou and G. A. Papakostas, "Machine learning for cloud resources management–An overview," *arXiv preprint arXiv:2101.11984,* 2021.

[135] S. Goodarzy, M. Nazari, R. Han, E. Keller and E. Rozner, "Resource Management in Cloud Computing Using Machine Learning: A Survey," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[136] T. Khan, W. Tian and R. Buyya, "Machine Learning (ML)-Centric Resource Management in Cloud Computing: A Review and Future Directions," *arXiv preprint arXiv:2105.05079,* 2021.

[137] N. S. Hogade, S. Pasricha and H. J. Siegel, "Energy and Network Aware Workload Management for Geographically Distributed Data Centers," *IEEE Transactions on Sustainable Computing,* vol. 7, no. 2, pp. 400 - 413, 2021.

[138] H. Ziafat and S. M. Babamir, "A method for the optimum selection of datacenters in geographically distributed clouds," *The Journal of Supercomputing,* vol. 73, p. 4042–4081, 2017.

[139] K. a. P. A. a. A. S. a. M. T. Deb, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation,* vol. 6, no. 2, pp. 182-197, 2002.

[140] A. Atrey, G. Van Seghbroeck, B. Volckaert and F. De Turck, "Scalable data placement of data-intensive services in geo-distributed clouds," in *CLOSER2018, the 8th International Conference on Cloud Computing and Services Science*, 2018.

[141] E. a. M. S. A. a. L. J. Cho, "Friendship and Mobility: User Movement in Location-Based Social Networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.

[142] S. Taheri, M. Goudarzi and O. Yoshie, "Providing RS Participation for Geo-Distributed Data Centers Using Deep Learning-Based Power Prediction," in *International Congress on High-Performance Computing and Big Data Analysis*, 2019.

[143] S. Taheri, M. Goudarzi and O. Yoshie, "Learning-based power prediction for geo-distributed Data Centers: weather parameter analysis," *Journal of Big Data,* vol. 7, p. 1–16, 2020.

[144] E. Baccour, F. Haouari, A. Erbad, A. Mohamed, K. Bilal, M. Guizani and M. Hamdi, "An Intelligent Resource Reservation for Crowdsourced Live Video Streaming Applications in Geo-Distributed Cloud Environment," *IEEE Systems Journal,* 2021.

[145] H. Wang, D. Niu and B. Li, "Dynamic and decentralized global analytics via machine learning," in *Proceedings of the ACM Symposium on Cloud Computing*, 2018.

[146] J. Bi, H. Yuan, L. Zhang and J. Zhang, "SGW-SCN: An integrated machine learning approach for workload forecasting in geo-distributed cloud data centers," *Information Sciences,* vol. 481, p. 57–68, 2019.

[147] P. A. Lima, A. S. B. Neto, P. Maciel and others, "Data centers' services restoration based on the decision-making of distributed agents," *Telecommunication Systems: Modelling, Analysis, Design and Management,* vol. 74, p. 367–378, 2020.

[148] S. Rawas, A. S. Zekri and A. El Zaart, "Power and Cost-aware Virtual Machine Placement in Geo-distributed Data Centers.," in *CLOSER*, 2018.

[149] X. Zhou, K. Wang, W. Jia and M. Guo, "Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers," in *2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS)*, 2017.

[150] Q. Li, Z. Peng, D. Cui, J. He, K. Chen and J. Zhou, "Data Center Selection Based on Reinforcement Learning," in *2019 4th International Conference on Cloud Computing and Internet of Things (CCIOT)*, 2019.

[151] D. Cheng, X. Zhou, Z. Ding, Y. Wang and M. Ji, "Heterogeneity aware workload management in distributed sustainable datacenters," *IEEE Transactions on Parallel and Distributed Systems,* vol. 30, p. 375–387, 2018.

[152] E. Baccour, A. Erbad, A. Mohamed, F. Haouari, M. Guizani and M. Hamdi, "RL-OPRA: Reinforcement Learning for Online and Proactive Resource Allocation of

crowdsourced live videos," *Future Generation Computer Systems,* vol. 112, p. 982–995, 2020.

[153] C. Xu, K. Wang and M. Guo, "Intelligent resource management in blockchain-based cloud datacenters," *IEEE Cloud Computing,* vol. 4, p. 50–59, 2017.

[154] T. Wei, S. Ren and Q. Zhu, "Deep reinforcement learning for joint datacenter and hvac load control in distributed mixed-use buildings," *IEEE Transactions on Sustainable Computing,* 2019.

[155] C. Xu, K. Wang, P. Li, R. Xia, S. Guo and M. Guo, "Renewable energy-aware big data analytics in geo-distributed data centers with reinforcement learning," *IEEE Transactions on Network Science and Engineering,* vol. 7, p. 205–215, 2018.

[156] F. Rossi, V. Cardellini, F. L. Presti and M. Nardelli, "Geo-distributed efficient deployment of containers with Kubernetes," *Computer Communications,* vol. 159, p. 161–174, 2020.

[157] Y. Qin, W. Han, Y. Yang and W. Yang, "Joint energy optimization on the server and network sides for geo-distributed data centers," *The Journal of Supercomputing,* vol. 77, p. 7757–7790, 2021.

[158] T. Tang, B. Wu and G. Hu, "A Hybrid Learning Framework for Service Function Chaining Across Geo-Distributed Data Centers," *IEEE Access,* vol. 8, p. 170225–170236, 2020.

[159] Z. Luo, C. Wu, Z. Li and W. Zhou, "Scaling geo-distributed network function chains: A prediction and learning framework," *IEEE Journal on Selected Areas in Communications,* vol. 37, p. 1838–1850, 2019.

[160] H. Wang, H. Shen, Z. Li and S. Tian, "GeoCol: A Geo-distributed Cloud Storage System with Low Cost and Latency using Reinforcement Learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 2021.

[161] H. Wang, H. Shen, J. Gao, K. Zheng and X. Li, "Multi-Agent Reinforcement Learning based Distributed Renewable Energy Matching for Datacenters," in *50th International Conference on Parallel Processing*, 2021.

[162] "TPC-H Benchmark Specification," Transaction Processing Performance Council, 2017. [Online]. Available: http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-h_v2.17.2.pdf. [Accessed 1 March 2022].

[163] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need," 2017.

[164] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018.

[165] "AI and the Data Center: Challenges and Investment Strategies," [Online]. Available: https://www.informationweek.com/data-centers/ai-and-the-data-center-challenges-and-investment-strategies-. [Accessed 1 June 2023].

[166] *Cisco Global Cloud Index: Forecast and Methodology, 2016–2021,* San Jose, CA: Cisco, 2018.

[167] "The value of geographically diverse data centers," [Online]. Available: https://www.flexential.com/resources/blog/value-geographically-diverse-data-centers. [Accessed 1 June 2023].

[168] "Carbon emissions of data usage increasing, but what is yours?," 1 June 2023. [Online]. Available: https://www.climateneutralgroup.com/en/news/carbon-emissions-of-data-centers/.

[169] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature,* vol. 518, no. 7540, pp. 529-533, 2 2015.

[170] N. Hogade and S. Pasricha, "A Survey on Machine Learning for Geo-Distributed Cloud Data Center Managements," *IEEE Transactions on Sustainable Computing,* vol. 8, no. 1, pp. 15-31, 1 2023.

[171] J. Bi, Z. Yu and H. Yuan, "Cost-optimized Task Scheduling with Improved Deep Q-Learning in Green Data Centers," *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* pp. 556-561, 10 2022.

[172] J. Zhao, M. A. Rodriguez and R. Buyya, "A Deep Reinforcement Learning Approach to Resource Management in Hybrid Clouds Harnessing Renewable Energy and Task Scheduling," *2021 IEEE 14th International Conference on Cloud Computing (CLOUD),* pp. 240-249, 9 2021.

[173] A. De Santis, T. Giovannelli, S. Lucidi, M. Messedaglia and M. Roma, "Determining the optimal piecewise constant approximation for the nonhomogeneous Poisson process rate of Emergency Department patient arrivals," *Flexible Services and Manufacturing Journal,* vol. 34, no. 4, pp. 979-1012, 12 2022.

[174] R. Traylor, *A Stochastic Reliability Model of a Server under a Random Workload,* arXiv preprint arXiv:1511.04130, 2015.

[175] "AIBench: A Comprehensive AI Benchmark Suite for Datacenter, HPC, Edge, and AIoT," [Online]. Available: https://www.benchcouncil.org/aibench/. [Accessed 1 June 2023].

[176] "U.S. Electric Power Industry Estimated Emissions by State," [Online]. Available: https://www.eia.gov/electricity/data/state/emission_annual.xlsx. [Accessed 1 June 2023].

[177] Brockman Greg, Cheung Vicki, Pettersson Ludwig, Schneider Jonas, Schulman John, Tang Jie and ZarembaWojciech, "OpenAI Gym," *arXiv preprint,* 2016.

[178] S. Imambi, K. B. Prakash and G. R. Kanagachidambaresan, "PyTorch," 2021, pp. 87-104.

[179] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *J. Mach. Learn. Res.,* vol. 22, no. 1, 1 2021.

[180] D. Zhang, D. Dai, Y. He, F. S. Bao and B. Xie, "RLScheduler: An automated HPC batch job scheduler using reinforcement learning," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020.

[181] "Generative AI Breaks The Data Center: Data Center Infrastructure And Operating Costs Projected To Increase To Over $76 Billion By 2028," [Online]. Available: https://www.forbes.com/sites/tiriasresearch/2023/05/12/generative-ai-breaks-the-data-center-data-center-infrastructure-and-operating-costs-projected-to-increase-to-over-76-billion-by-2028/. [Accessed 1 June 2023].