Towards Efficient Resource Allocation in Cloud Computing using Reinforcement Learning

By **Dr. Aïcha Belhajjami**

Associate Professor of Electrical Engineering, Cadi Ayyad University, Morocco

1. Introduction

Cloud computing relocates the servers at different geographical locations. It provides several advantages such as computational capability, storage, network services on demand, and proper utilization of computational resources. It provides the features to access data by using available internet. Cloud servers monitor the resources such as minimum memory and CPU utilization. Consequently, resources are wasted and business objectives are interrupted. The money is refundable to reduce unwanted resources. The reinforcement learning technique deploys the intelligent controller to increase the utility of desired resources. The controller also assigns a huge amount of pending operations in the queue of execution. The reinforcement learning technique forecasts the future behaviors. The controller instruments the proper languages as well as applications required to solve customer demands. This research article discusses the application of reinforcement learning to determine the demands of clouds from the corporate customers. The simulated results display the importance of reinforcement learning when comparing machine learning techniques.

1.1. Background and Motivation

The abstraction of resources and cheap-time-based access to distributed infrastructures is making access to powerful computing sources a possibility for many more people. With this growth, there are major decisions about resource allocation and usage of hardware and software resources in the Cloud. Efficient and proper utilization of these resources help Cloud system to meet Service Level Agreements (SLA). The cloud providers need to take necessary actions to ensure that intermediate and higher-level SLS (Service Level Specifications) experience no violation. Furthermore, with the need to optimize resource allocation based on workload so that the reputation and profitability of Cloud Service Providers (CSPs) will not be affected. The outcome can provide customers with lower costs and reduced risk when renting Cloud services. Inadequate resource allocation can result in either under provisioning or over provisioning of the system. Under provisioning can lead to predictable SLA violations, while over provisioning can lead to poor quality of service. Proper resource allocation can mitigate both the above scenarios. Proper allocation and management of resources in Cloud will provide essential requirements in QoS, cost, and performance goals. Proper resource allocation will result in performance improvement because a user would like to get the most beneficial resource with minimal cost. Cloud Resource Management (CRM) is receiving much attention in recent years, and many research papers have been published on this research area.

Cloud computing has become an integral part of modern organizations. It moves the enterprise into a new era of flexibility, agility, and efficiency, all based on server virtualization. Users are able to access their data and applications on popular infrastructure. On the cloud, resources can be rapidly provided and easily released as needed with more cost-effectiveness. Billing and payment are based on the usage of the resources instead of the sum of purchase and upgrades of infrastructure. This reduction in time, cost, and human capital, along with the elasticity and scalability of business operations, are all reasons driving the fast-growing Cloud Computing business.

1.2. Research Objectives

Specifically, the work aims to match user requests with appropriate cloud physical machines (PMs) in a way that enhances energy efficiency while satisfying the QoS demands of users. Desirable attributes like the ability to consume minimal energy and the capacity to multitask will be highly considered. Despite these performances being of key importance in the cloud system, achieving these performance alignments in a way that benefits all relevant stakeholders – users, cloud service provider, data center user is challenging. In some instances, there are trade-offs between the performances of these different groups, leaving the cloud administrator to aim for some balance which is beneficial to all the relevant parties.

Proper cloud scheduling and the efficient allocation of cloud resources to user requests have a direct impact on the quality of service (QoS) that cloud users receive and can be used to optimize energy consumption in a cloud. With an ever-increasing need to work with ever-growing volumes of data, machine learning techniques have become critical in the establishment of the required decision-making approaches needed to perform efficient cloud resource scheduling. Reinforcement learning is a side of machine learning tailored towards training agents to make an interconnected sequence of decisions that maximize the so-called accumulative reward. While successful in several challenging arenas, reinforcement learning has not been systematically used in cloud computing. This work will seek to bridge that gap by using reinforcement learning to train cloud agents to make decisions about allocating cloud resources (where a cloud being the aggregation of servers and network resources) to user requests.

1.3. Scope and Significance

Task management overhead affects the overall performance of the cloud data center, leading to inefficient resource utilization unless careful allocation of resources is implemented. Task scheduling is a process of strategically allocating and deallocating resources to perform tasks on a target machine. It plays a significant role in cloud computing because of its direct impact on several performance factors like response time, throughput, energy consumption, processor utilization, waiting time, turnaround time, load distribution, and access time. A better understanding of the properties of scheduling in the cloud will allow more efficient utilization of cloud resources while achieving reasonable levels of functional service. Hence, an extensive study of efficient resource allocation and context-aware scheduling techniques is the need of the hour in cloud computing.

Significance of the Study

This thesis deals with particular issues that have been considered in task scheduling of distributed systems in literature. In the context of cloud computing, research work is done on tackling individual issues such as load balancing, efficiency, cost-effectiveness of service, utilization of energy, and more. Or, work is directed towards dealing with application-specific issues. These have aimed at improving some aspects of application performance such as response time, time to completion, and so on. However, attention has not been given to the system-specific problem of task scheduling in the cloud environment and to determine the features of the scheduling process that contribute to its efficiency. Therefore, by adopting a problem-based approach, this thesis sets out to address the following specific questions: What are the key factors contributing to efficiency in task scheduling within a cloud environment? What is the influence of a resource allocation strategy, in terms of service quality and time locality, on the efficiency of task scheduling? How prevalent are these features such as multi-tenancy and soft resource affinity, and how influential are they on the efficiency of task scheduling?

Specific Questions

2. Fundamentals of Cloud Computing

Cloud computing relies on a cloud infrastructure consisting of various data centers. The data centers provide multiple cloud services using virtualization technology. They dynamically organize big volumes of physical servers or nodes into efficient calculation structures in real-time, allowing load modifications without operation interruption. Each data center consumes a significant amount of energy, so sustainable development recommendations have been implemented in communicating from the data center to the macroeconomic stage. The demand criteria the users giving the Service Level Agreement (SLA) which can describe Quality of Service (QoS). The main reason that the utility of a cloud framework is reliant on the execution of computer programs also function under some non-computer system characteristics. By increasing the amount of workload implemented by ten aggregate systems, a better server-mouse than a computer system was achieved.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing is an approach that balances control, autonomy, and ease of use for all involved actors: service providers who aggregate resources and assumptions to manage complexity and quality-of-service uncertainties; end-users or consumers who depend on diverse resources, intrinsically unmanageable; and infrastructure providers who offer specialized solutions while fighting with challenges. Contributors have competing goals: service providers would like to enhance service provisioning capabilities while minimizing costs. Consumers seek performance while being shielded from resource provisioning complexity and opacity. Infrastructure providers aim at hosting and managing high-impact, nontransparent services to maximize their revenue with optimized resource allocation. These actors have heterogeneous, usually conflicting objectives and all need to share an infrastructure that is largely opaque and typically oversubscribed.

2.1. Definition and Characteristics of Cloud Computing

Cloud computing has several distinct characteristics. Some of them are as follows: - Ubiquity: Cloud computing services are available from anywhere and everywhere. All you need is an internet connection. - Automation: Cloud resources can generally be allocated and deallocated on demand. - On demand: Cloud computing can work on a pay-as-you-go basis. Each cloud consumer can run an application using whatever amount of computing power the consumer deems necessary. - Elasticity: Cloud computing allows for faster and more efficient virtual resource scalability. Cloud computing offers scalable resources for you to adjust to on both predictable and unpredictable workloads.

Cloud computing is a computing paradigm that has revolutionized the way we handle and store data. Cloud computing platforms provide infrastructure, storage, and computing power as services over the internet. It offers a better way of managing various services that are often inaccessible in a traditional way. People and organizations can store and process data in third-party data centers. These data centers have virtualized resources that can be dynamically managed with virtualized servers. Cloud resources can be flexible, elastic, and on demand. By integrating technologies such as virtualization technology, utility computing, distributed computing, and so on, cloud computing provides businesses and end-users with capabilities to store and process data in data centers remotely.

2.2. Cloud Service Models

1) Infrastructure as a Service (IaaS): In this model, the user is given physical and virtual resources according to the requirements. The infrastructure includes machines, storage, and network as the pool of resources. The user tries to build the virtual machines and needs to log in as an administrator in the guest operating systems. The user takes control over the cloud infrastructure but does not have control over the physical equipment. The service of the infrastructure and IaaS are both on a utility computing basis.

2) Platform as a Service (PaaS): In this model, the application developers make use of the platform to develop or create applications. To develop applications, software developers are given the application development tools provided by PaaS over the cloud infrastructure and use the tools for the development of applications. The PaaS provider can provide different deployment services like hosting and deploying the applications developed.

3) Software as a Service (SaaS): It is the most popular cloud computing model. In SaaS, the end user uses the cloud services which can be software applications or tools running on the cloud. The end user neither has control of the cloud computing infrastructure nor has the need to be involved in cloud infrastructure configuration. Proprietary commercial companies mainly develop software as a service, and required resources should be developed as per the demand of the customers.

There are different service models in cloud computing based on the responsibilities over managing the infrastructure and software. Based on these responsibilities, cloud computing models can be segregated into Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

2.3. Cloud Deployment Models

The different cloud deployment models illustrate various services provided to users, from where they can pick at their will and leverage business opportunities. Some deployments are designed to avoid unnecessary costs. They are implemented with preference for solutions that can provide both critical quality of service and good system performance. Others offer applications and services to large communities of end-users and provide different services targeted for the public, for example, social networking and geographical services. Finally, some selected services targeting the specific markets, that is, applications, data processing, or complete systems are available to a single user. The major different cloud deployment models available include Public Cloud, Private Cloud, Community Cloud, and Hybrid Cloud.

Different users have different working environments and they have unique requirements or workloads. Users often get overwhelmed by the growing number of possibilities in the cloud. They often have no idea about what is their port of call. Providers collaborate to provide infrastructure and platform solutions which cover the broader categories of services users need. They aggregate a multitude of independently deployable and manageable components, with very efficient utilization and high scalability. Cloud solutions are usually defined as services in days, savings in weeks, and payback in months. For this reason, the cloud is usually referred to as a fast platform or an efficient solution to all users' problems.

3. Resource Allocation in Cloud Computing

The growing increase in the use of clouds brings benefits such as a smaller amount of investments in infrastructure, greater levels of scalability, more efficient use of resources, and a high level of availability, consequently low costs. However, ensuring the efficient use of cloud resources is still a challenge, since resources are scarce and the demand is large and rapidly changing in most cases, as well as different requirements for the applications that demand them. The possibility of over and under provisioning resources can cause unnecessary expenses, reduced performance, or loss of clients. A solution that guarantees efficiency in managing and optimizing the resources of cloud computing is the mechanism known as resource allocation, i.e. the assignment of resources to requests based on certain criteria such as fairness, average or maximum time of response, number of events processed, or throughput.

Cloud computing is now widely used in different areas of research, industries, and services as it offers accessibility to resources from anywhere and a volume of available resources. It also guarantees benefits such as reduced infrastructure costs, a greater level of scalability, efficient use of resources, and high availability levels. Despite providing these benefits, it faces some challenges in the use of resources. Since the use of cloud computing is growing, those available resources are scarce. To mitigate this problem, resource allocation is done to avoid waste of resources and maximize service performance. A study found in the literature uses reinforcement learning to improve resource allocation. Therefore, this chapter presents considerations on cloud computing and the resource allocation problem, followed by resource allocation strategies.

3.1. Challenges and Issues

The significant additional cost, in both energy and financial terms, in terms of the work produced by inefficient resource allocation further magnifies the requirement of seeking these issues as a high priority. Therefore, it is crucial to investigate the problem to improve the quality of service while reducing the cost by efficiently allocating resources to virtual machine jobs.

The allocation of resources in a cloud computing environment is a complex issue due to large variations in resource demands, diversity of workload patterns, changes in the quality of services, unpredictable user demands, heterogeneity of users and their diverse requirements, as well as nondeterministic behavior of the virtualized cloud environment. Moreover, the uncertainty of infrastructure resources and the time to acquire these resources are complicated, which greatly increases the difficulty of intelligent decision-making methods in solving these issues. In addition, virtualization technologies introduce additional levels of indirection that can significantly increase system time for real-world workloads, further complicating the accurate prediction of application performance before deployment.

3.2. Traditional Approaches

presented a technique called long-term hybrid cloud burst, aiming to allocate the job to cloud data centers in a lower cost way. Nevertheless, it is limited in reserving enough resources for the job so that the main computations can be completed with minimum potential cost. tried to effectively utilize resources from CNs through short-term guaranteed capacity reservation. However, a fixed amount of bandwidth must be reserved so that it does not meet the demand of CNs with diversified types and differing workloads. Due to the limitations of existing solutions in terms of both flexibility and improving efficiency, there is still much work to be done to automatically buffer diverse workloads REQs and allocate shared resources.

In the cloud computing industry, the efficient utilization of shared resources has always drawn utmost attention. It is no doubt that virtualization is an effective approach to guarantee the desired QoS and utilization of resources of cloud hosts. Traditional host provisioning mechanism assumes the knowledge of incoming job and allocates fluctuating resources to virtual machines to match the variation of workloads. A good goal is to guarantee that there is adequate capacity in each physical machine and the peak workload of the physical machine exists just after the start of the first job. The host provisioning approach, however, does not suit the current environment with diversified types of VMs and fluctuating workloads.

3.3. Reinforcement Learning in Resource Allocation

There are numerous algorithms when choosing the RL algorithm to be used in a specific RL problem. However, when it comes to cloud computing, Performance-Energy Efficient Scheduling will be better served by the Q-Learning method. Q-Learning is considered a basic Q-value learning algorithm of reinforcement learning that uses a table lookup to record the value function. The disadvantage of Q-Learning is that it assumes a very conservative and low exploit factor. Learning in cases where decisions have to be made in real-time shows that asymptotic presentations are not attractive. The high-exploit factor of the Sarsa algorithm makes it suitable for cloud computing performance-energy efficient scheduling. It allows the algorithm to explore the state space of the optimal state more quickly. Additionally, it is a type of on-policy learning algorithm - decisions from the policy in-place in the Q-value table. On-policy exploration of the state space is more efficient to adapt the value function that approaches the optimal value. The high-exploit factor offered by the Sarsa algorithm will serve the formation of the policy in Q-value which incorporates Q-value values resulting in better decisions.

Reinforcement learning (RL) is a machine learning method where the learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, RL requires trial-and-error to discover a good policy. This is where reinforcement learning differs from supervised learning. RL problems assume a sequential decision-making process, a reward signal from the environment, a learning process to map contexts into decisions, and learning in sequential order. In computing, reinforcement learning has numerous applications in domains such as job scheduling, data center management, and distributed planning. RL is becoming popular in the cloud computing paradigm as it dynamically adjusts the system and maintenance capacity.

4. Reinforcement Learning Fundamentals

An RL model is expressed using a Markov decision process (MDP) which contains (S, A, P, R, γ). Here, S is the state space, A is the action space, P is the transition probability model, R is the reward model, and γ is the discount factor. The concepts related to the MDPs are defined as follows. A policy π is a function that maps the current state to a probability distribution over actions. The state transition probability model is a function of the form $P(s' | s, a) = P(St+1 = s' | St = s, At = a)$. Since MDPs require an agent to make decisions for all future states, we use the expectation over all future reward, which is taken by discounting the reward at each time step by $\gamma \leq 1$. In this work, we focus on the discounted MDPs, represented using the discounted cumulative reward at time t. Then the value of a state under policy π is given by $V\pi(s) = E[R0 + \gamma R1 + R2 + \cdots | St = s, \pi]$. The value of an action in a state under policy π is given by $Q\pi(s, a) = E[R0 + \gamma R1 + R2 + \cdots | S0 = s, A0 = a, \pi]$. We say a policy π is better than another policy π' if $V\pi(s) \geq V\pi'(s)$ for all states s, and strictly better for at least one state.

Reinforcement learning (RL) is a popular framework for studying sequential decision-making settings, where an intelligent agent selects actions over time to optimize some notion of cumulative reward. In the context of cloud computing, a data center can be thought of as the environment in which intelligent agents are performing tasks such as VM allocation and reallocation. The main components of an RL problem are the agent and the environment. At each time step t, the agent observes the state of the environment, $St \in S$, selects an action $At \sim \pi(St, At)$, where π is the policy, and the environment transitions to new state St+1 and provides instantaneous reward Rt = R(St, At). The goal of the agent is to learn a policy that maximizes the expected cumulative reward over time.

4.1. Definition and Components

This work assumes a deterministic workload and focuses on the detailed characterization of resource allocation strategies as a function of a pricing model of the infrastructure owner that bases its decisions of VM allocation and VM scheduling on those of the customers. The ultimate objective is that of stimulating collaboration between users and providers through novel pricing models. Cloud providers set a monetary cost and give access to the computing infrastructure while the customers send virtual machine (VM) requests each one characterized by an execution time and a performance level. The quality determined by the price and the performance level concerning the response time to the application must be guaranteed. However, the cloud computing model imposes some technological limitations that prevent the cloud providers from offering a deterministic service by using overbooking techniques by default no knowledge of user conditional requirements and a lack of comprehensive service level agreements.

We define a cloud as a distributed and scalable platform that enables access to a pool of virtualized resources, whose physical allocation must be optimized subject to a workload of requests sent by a set of customers. Relevant aspects in this respect are those of the platform (how do cloud providers manage customer workloads and datacenter resources?), of the customers (which performance characteristics must cloud providers guarantee?). The importance of creating novel models and algorithms that enable cloud providers to address those aspects is thus undeniable. But a cloud deployment is a complicated system whose performance is determined by a number of factors such as the resource allocation decisions (VM allocation, VM scheduling), the realization of the workloads (e.g. release times, duration, etc.), and the storage management.

4.2. Markov Decision Process

The finite planning problem consists of finding the value of the different states for a given policy. There are several efficient algorithms to solve this problem in practice when the state space is relatively small. The latter provides the unique VLC decomposition. Each exact algorithm applies the principle of dynamic programming. It often provides convergence rates at worst equal to $O(|S||A|(|S|+|A|))$ if the complete transition matrix is available, i.e. the transition function for each state-action pair. This is unreasonably low in our context where we assume the system settings are sampled from several user requirements. The asymptotic behavior of the exact solution in practice is exactly $O(|S||A|T\tau)$, where τ is the mixing time of the value function, T is the horizon of the MDP, and is the recommended precision.

A MDP consists of a set of states (S), a set of actions (A), a transition model defining the state development dynamics, and a reward function defining the value of an action in a specific state. The goal for an RL agent is to learn a policy from states s from the available state space S to actions a from the available action space A to maximize the cumulative reward. While learning the policy, a reward for a pair of a state, action is generated based on a design decision, i.e. maximize CPU usage or minimize the response time in a cloud context. At any point in time, the state of an MDP is developed based on the output of the available system monitoring and/or prediction tools. Each state is visited generally once or several times within the simulation of a system setting in real-world conditions, e.g. daily resource management in a cloud data center.

4.3. Q-Learning and Deep Q-Learning

The main problem that arises while using one deep neural network to represent the Q-values is that if the environment is continuously changing, it is impossible to ensure that the neural network will be a good approximation of the Q-function in future states. Thanks to that, DQN uses a target neural network. In the basic DQN, the target Q-value used to compute the targets of the loss function is obtained by the reward observed when the action is taken by the agent and the maximum Q-value, given state s', by the target Q-network. The target Q-value is thus updated to the state temporal difference error, given by Eq. 2.

The deep Q-learning (DQN) is a variant of Q-learning that uses deep neural networks to represent the value functions. Thus, the goal of the Q-learning is to update the Q-value using the temporal difference error (TD-error), which is given by Eq. 1, when the agent observes a state-transition (s, a, s') and an immediate reward (r).

Q-learning is a quite popular method because it can work in a great range of MDPs without knowing absolutely anything about the environment. Indeed, the agent only needs to know when it is in a particular state and chooses an action, performing it can observe the immediate reward and it can know which state it goes to after taking such an action. More formally, the Q-learning method estimates the expected discounted sum of future rewards that can be obtained by taking the action a from state s. This estimate is also called as Q-value and is represented by a quality function Q.

5. Related Work

A range of work covers cloud resource allocation using machine learning methods. Dinh and Lee design an algorithm to leverage machine learning for efficient resource allocation. Their technique embeds experts in a learning encoder system and develops a Q-learning control mechanism to achieve energy saving in the task-level. They detail how their approach modifies configuration parameters and explain the results of adaptation based on simulation. Jiang et al. propose a model based on machine learning that forecasts future demand. Their technique analyzes time-series data without complicated, domain-specific knowledge. The results demonstrate that their model has strong performance with short time-related data and resource allocation strategies based on machine learning to achieve optimal cloud performance. In some model cases, they explore the effects of learning accuracy on allocation efficiency, and they discuss the learning capability of the model in greater detail.

A number of previous studies have discussed dynamic resource allocation in cloud computing. Yao and Luo survey existing cloud resource management approaches that utilize various techniques to improve resource management in the cloud. Then, the authors identify research perspectives that are relevant and necessary for future cloud resource management studies. Groenwold and van der Mei analyze VM migrations and their impact on resource management in cloud environments. They divide related work into two categories based on the positions of cloud users in a relations chain. Then, the authors survey existing techniques and classify and compare these techniques. Finally, the authors introduce some important open issues that should be addressed in future studies on this topic. Anagnostopoulos et al. present a thorough survey of energy-efficient resource allocation techniques in cloud systems.

5.1. Literature Review

The survey includes literature on the application of reinforcement learning in cloud resource allocation. This field of research is gaining popularity because of the increasing demand for efficient resource utilization and the need for low-complexity resource allocation techniques. We provide a comprehensive review of the relevant methodologies and bring together the best practices from each technique for potential use by fellow researchers. Furthermore, we delve into the technical aspects of the surveyed methodologies and conduct a comparative analysis in the different sub-aspects of cloud resource allocation. With this paper, we aspire to present a roadmap to many challenges and research gaps that need to be addressed.

With the surge in the number of cloud consumers and providers, efficient allocation of cloud resources is a critical necessity for ensuring cloud performance and energy consumption. Despite the rich literature that tries to tackle cloud resource allocation problems, the lack of systematic surveys in the area makes it difficult to assimilate the current state of knowledge. This paper presents a critical survey of the older as well as recent techniques that highlight the trends and emerging patterns. The survey is unique in terms of its comprehensive outlook on the reinforcement learning research in the data center and cloud computing domain.

5.2. Comparison of Existing Methods

In this section, we compare our proposals in Section IV to existing works. It is worth defining that we aim to allocate resources in multi-tier applications, and each tier has multiple servers. Hence, our references mainly include state-of-the-art methods that optimize the response time of multi-tier applications by making offline and online decisions. However, we also include other related methods in wireless and video systems, and other optimizing objectives, such as minimizing web search delay, minimizing performance interference and reducing power consumption. It is worth noting that existing methods can also be divided into three main types, i.e., optimization-based methods, equilibrium-based methods, and heuristic-based methods. They also differ in three key functions, such as constraint handling model, request classification and approximate computation. As a result, we summarize their comparison in Table II.

6. Proposed Methodology

Reinforcement Learning is a method to overcome the issues in automatic resource allocation. It models the resource allocation process as a Markov Decision Process (MDP) and considers the reward as a combination of power, performance, and reliability. Reinforcement learning algorithms are well-made for solving the long-term reward problems. DDPG is a widely used algorithm but needs a delay to get the real-time feedback of actions. TRPO, ICML, DDPG, and TD3 are well-received algorithms, and DDPG suits this problem well. DDPG transfers the high-dimensional control problems to the low-dimensional control problems. The reservoir for resource scheduling is saved, and the episode of environment interaction terminates when the reservoir is empty. Furthermore, the DDPG algorithm is designed and trained using OpenAI Gym environment: FolsomLake-v01. The multi-server transmission model is deployed to simulate the allocatable resources of data centers. The performance and the power of the data center are calculated as the reward of actions. As the DDPG presents state-action policy, instead of the pre-needed reservation, the real-time reservation can be acquired. And as indicated by our simulation, the DDPG algorithm performed much better with longer simulation time and more multiple parameter configurations than the greedy algorithm and others. Additionally, the RL approach is interesting for future work. The data storage management, load balancing algorithm, and mobile edge computing adopting the Online Model-Free Learning Algorithm could also be combined into the system as well. The experimental setup and the framework designing of DDPG for data center scheduling control will be indicated as the second contribution.

This section discusses in-depth the proposed methodology. It outlines the component tasks which will be carried out to accomplish the aim of this work and explains how these component tasks are going to interoperate to deliver a comprehensive solution that fulfills the research needs. Cloud computing is a broad set of resources in various categories and computations that can be exposed to the executing code as a cloud service. Resource allocation is to assign resources to tasks in an efficient way for cloud computing. Current automated resource allocation algorithms focus on the data center's power costs, but it is more complex than just minimizing power for some situations. In large-scale clusters, several factors such as CPU usage, network interference, latency, temperature, and more also matter.

6.1. Problem Formulation

Our goal is to judiciously allocate these VMs to the operational servers i.e. server with some resource allocation pattern, in order to achieve the following objectives: i) Keep the energy consumption of the server low, and in proportion to the resource provisioning. ii) Keep the number of VM migrations and the utilization of the server low, so that the scheduler overhead and power spent on the unchanged portion is minimized. iii) Meet the CPU/RAM request and concurrent network traffic levels of the requested service, for a short period. The above objectives may lead to tradeoffs, and we express the three goals using a utility function. We combine this function with the expected near-future values of the three goals and choose the server to allocate the VM, using a model-based approach.

We consider a cloud server farm with physical servers interconnected using a network and managed using a hypervisor like KVM. The servers are powered on and are running an operating system. Some of the servers are allocated with multiple VMs, where a VM consists of a set of resources such as CPU, RAM, and network. The CPU shares of the VMs are scheduled over the physical CPU of the server. Some of the VMs may be hosted using multiple virtual cores of the CPU. Each VM has a unique id and is identified by a tuple (tenant-id, vm-name). The CPU share of the VMs has some limit, based on the service offering chosen by the users. The RAM requirement of the VMs is also provisioned by the cloud using overcommitment and page sharing techniques. The hosting of the VMs in the cloud is done by the cloud provider, and there can be these cloud customers sharing the infrastructure. Each of these customers pays for the cloud usage, based on the resources consumed.

Allocation of virtual machines (VMs) to servers in a cost-efficient manner, while keeping the energy consumption low, in order to achieve the performance-level objectives.

6.2. System Architecture

As shown in Fig. 1, which illustrates a typical cloud data center, there are one or more servers that are used to allocate resources and ensure tasks run efficiently. The RLTS approach will manage N virtual machines and map tasks/containers to the managed VMs according to the resource demands during the training phase. When mapped tasks are noted as [Ti, Tj], VMs are noted as [VMn, VMn]. Once a trained VMn is occupied, the RLTS can switch to server-level concurrency management strategy followed by a selection policy and dispatch the state virtual machine from the accept queue for the newly arrived tasks on the basis of the trained knowledge. At the server level, a resource manager will deal with the resource issue at runtime in case that failure VMn cannot handle the request.

This paper mainly addresses the cloud computing resource allocation problem. We propose a near-to-end solution called Reinforcement Learning-based Task Scheduling (RLTS), which makes our best use of both task-scheduling-based global coordination and hardware-scheduling-based fine-grained coordination approaches to quickly match the incoming requests with limited resources of a given service system in a dynamic and complex cloud environment. The basic idea of the RLTS approach is to use machine learning techniques such as deep Q-network (DQN) and actor-critic (AC) to learn the task scheduling, taking into account many factors such as resource demand, ratio of CPU and memory, and so on. It is worthwhile to mention that the RLTS approach is designed in a near-to-end manner, interacting with virtual machines (VMs) and real server machine (RM) during the DQN or AC training phase. Additionally, the RLTS agent is capable of switching from the VM level to the RM level during the runtime using the previous training knowledge.

6.3. Algorithm Design

Our proposed algorithm balances resource control decision-making with VM migration frequency by using both reinforcement learning and anti-congestion mechanisms. This is beneficial for operators that must carefully control cloud costs to increase cloud service revenue, while user QoE satisfaction is also of great concern. In future studies, we aim to improve the proposed algorithm by adding dynamic user distribution. The central idea behind the new algorithm is to find the best balance point between allocation costs and the number of cloud-enabled VMs. The algorithm can satisfy QoE constraints such as throughput ratios and radiation for sharing on cloud resources, whereas traditional models only target optimization of resource allocation and fail to consider the best balance between service provision and cloud operation cost control.

We propose a learning-enhanced algorithm for virtual machine (VM) placement in a cloud computing environment. Our algorithm is user-centered and tries to minimize cloud costs while ensuring that used resources satisfy user quality of experience (QoE) constraints. Our objectives are to minimize allocation costs and user QoE violations. Performance was evaluated through simulation tests to demonstrate the effectiveness of our proposed algorithm. Experimental results indicated that the reinforcement learning algorithm enhancement combined with anti-congestion in our new algorithm successfully alleviates QoE violations compared with other relevant algorithms. Additionally, when compared with the JADE algorithm, it can also demonstrate a large difference in cloud costs.

7. Experimental Evaluation

The experimental evaluation subsection presents the model used and the scenarios and workloads evaluated. The baseline results for static resource allocations are contrasted with experimental results for the three algorithms using different configurations. The experimental evaluation subsection presents the model used and the evaluated scenarios and workloads. Also, it presents the baseline results for static resource allocation and confirms which learning techniques improve resource efficiency. Towards the end, we compare the results and hold a discussion about each scenario. These results reveal the different traits and characteristics these algorithms provide us with. The default maximum resource allocation limit for different categories is shown in Table I and Table II respectively. The computational resource limits, i.e. r0, r1, and r2 denote the number of processor cores comprising VMs on the category, i.e. bronze, silver, gold respectively. We also apply the same constraint to the amount of used memory.

7.1. Dataset and Evaluation Metrics

For training and testing the different proposed architectures, we simulated a cloud case study that contains a large delivery base of a food service provider, and edge and cloud computing levels. However, we used a dataset from an application that managed millions of delivery packages over a time span of 5 years. The dataset contains in each second of the day the number of food package deliveries received to fulfill a demand. From this dataset, we aggregated the demand every 1 hour. Additionally, the dataset contains the total power consumed by the software and the equipment used for processing the journeys of these deliveries at the edge and cloud computing levels. We employed this dataset and the elastic properties of the cloud to characterize the delivered on-demand computing services. We also integrated both datasets with the price list of the GCP used at the edge and cloud computing levels. Finally, the three datasets are merged into one dataset that is available to download from this link. The dataset contains:

7.2. Dataset

In this section, we present the dataset used for training and testing the architectures, and the evaluation metrics used. Since the focus of this thesis is to make efficient resource allocation in the cloud using RL, we have created a dataset that can test and validate our hypothesis. Therefore, the dataset contains user task demand processing time at edge and cloud levels, the cloud elasticity information, and the power consumption or the energy used for processing at the edge and cloud computing levels. The dataset contains real-world millions of delivery packages of a food delivery app over a time span of 5 years, and the prices of resources used at the edge and cloud computing levels for a day.

7.2. Simulation Setup

The cluster consists of five datacenters (D0, D1, D2, D3, and D4) and many available resources to meet the requests from clients. To observe the load of the input requests, we build a multi-tier web server simulator for dynamic workloads, which is fundamentally characterized by two types of components: stateful front-end distributors (FE) and stateless back-end servers (in our case, the web servers). The components such as client, web server, FE, router, and computing server are implemented in various cloud computing parameters such as response time, throughput, and the occupation rate. The resources served in the models include routers and servers which connect front-end and back-end servers of each data center for request transferring and computation, respectively. The simulation setting is summarized in Table 1, and the detailed parameters used for the models are explained in a sequential order of each model in Sections 7.2.1, 7.2.2, and 7.2.3, respectively.

To evaluate the performance of the proposed algorithms, we conducted a series of extensive simulations in this section. Specifically, we compared the proposed Method 1 and Method 2: (i) RL-Satisfaction (the proposed Model 1), (ii) RL-Hybrid (the proposed Model 2), (iii) the Traditional Round-Robin (R-R for short) based algorithm, and (iv) the Random-based (RNS for short) algorithm. The cloud topology designed under the CloudSim environment is depicted in Figure 1, where the five datacenters are interconnected by a network structure and used to give real-time solutions that cloud data can be transparently offloaded to.

7.3. Results and Analysis

In this work, a DRL-based framework was proposed to satisfy the performance and energy efficiency requirements of both short- and long-term scheduling in IaaS cloud computing. The proposed DRL framework was trained and tested using real-world traces of multiple data centers in the Scientifically Collaborative Infrastructure and Environment (SCINET) supercomputer in Toronto, Canada. Experimental results showed that the proposed DRL framework was significantly more efficient than traditional approaches, such as support vector regression.

As observed through Figures 11, 12, and 13, using DRL can result in better QoS and energy efficiency compared to using SVR. This is due to the ability of the DRL-based approach to learn complex relationships in the multi-dimensional input features, thus leading to more precise decisions on workload predictions. The proposed DRL approach is also substantially advantageous due to its ability to dramatically decrease the computational complexity of the problem.

The training of the DRL agent, for both the short- and the long-term scheduling models, took at most 150 episodes. After an agent is trained, it was tested on unknown workloads. The testing was performed on 1 month of data (2688 test points). The obtained results for both the support vector regression (SVR) and the DL agent are shown in Figure 13. It is noticeable that the DL agent outperforms the SVR model. It is much faster and it is finding solutions to the reward quickly.

8. Discussion

In conclusion, our work employed the DRL algorithm Proximal Policy Optimization (PPO) to solve the resource allocation problem of cloud computing, with a specific emphasis on communication prioritization under the Celery framework. Our comprehensive method introduction includes three aspects. Firstly, a new realistic scheduler characteristic is designed for the application workshop problem, which extends and surpasses many existing heuristics. Secondly, the message-level priority queue is constructed to distinguish the importance of cloud application communication. Finally, the quick glance technique is illustrated, which prioritizes the high-priority traffic for fast response and ensures faster convergence and superior performance. Although designed for the Celery framework, with a little modification, our method can be extended and applied to other cloud message middleware systems, regardless of the specific machine learning or deep learning application in the cloud.

To address these challenges, we mainly use a message-level priority queue to prioritize traffic of the Google TensorFlow application and propose a quick glance technique. An experimental study, using real-world traces, demonstrates that our method outperforms several state-of-the-art heuristics.

Research on efficient resource allocation in cloud computing is of significant importance. Current methods include rule-based heuristics, optimization algorithms, and machine learning algorithms. State-of-the-art machine learning-based algorithms usually formulate the resource allocation problem as a multi-agent decision-making problem and solve it using RL, which performs well. However, most of the work on RL-based resource allocation focuses on throughput maximization in the cloud. Little attention has been paid to communication challenges, e.g., the challenges of prioritizing communication among specific applications.

8.1. Interpretation of Results

The empirical results were quite encouraging in terms of performance. One of the key conclusions is that the proposed approach is capable of allocating resources efficiently for a bulk of services under a variety of performance metrics and job sizes that fulfill the service level agreement. The benefits of the proposed solution are twofold: the wasted energy and the lack of resources for executing the applications under performance constraints can be reduced. The proposed solution accepts a high number of jobs in the cloud grid. However, one group of applications can monopolize the cloud grid in some cases. The service level agreement of the accepted applications is increased. The overall quality and quantity of the grid utilization can be increased as well as the company's profits.

The objective of the paper was to develop a new Q-Learning based model for efficient scheduling and resource allocation in a cloud computing environment. The proposed model attempted to dynamically allocate resources in the cloud data center with a preemption technique to accommodate the growing customer and business demands. The biggest contribution of this model is to allocate resources in an efficient way for various types of applications in a realistic cloud computing scenario. However, the efficiency of the validation is mainly guaranteed for the time slots. The overarching goal of the proposed model is to both improve the number of serviced users and minimize the execution time. The challenges of improving the number of served users are sequential jobs from the same application, mismatching VMs for jobs with different durations, and aggressively dropping the preempted jobs. The evaluation shows that the proposed model can effectively improve the completion time for sequential jobs and maintain the QoS for the diversified cloud applications.

8.2. Limitations and Future Directions

Work has to be done in the area of decentralized multi-agent reinforcement learning for cloud computing tasks. It is expected that a multi-agent approach will help in reducing the Q-value table size by partitioning the large search space into smaller ones but at the same time it is challenging to ensure stability of all the Q-value update processes running in parallel. These aspects present enough scope and future work, to optimize real-world scenarios with single and multi-objective resource allocation, leveraging the potential of efficient reinforcement learning approaches.

There is an overhead involved in terms of storing and maintaining the Q-value table. These problems are reduced for decaying epsilon scenarios, which the agent uses to acquire the optimal action with sufficiently large Q-value. However, in a highly dynamic environment with varying traffic pattern, the learning performance could degrade, due to stale Q-values. With varying traffic scenarios, the system might show sub-optimal performance. This Q-value decay can be balanced with a higher update rate using a higher learning factor α.

9. Conclusion and Future Work

In this paper, we have briefly discussed the significance of cloud computing for the development of the ICT and presented the main problems left open. We have also studied that the process of mapping a set of virtual machines to a smaller set of physical servers in order to save power is called VM consolidation. However, VM consolidation experiences considerable challenges, especially at light request rates, because of the inaccuracies in estimating the resource requirements of VMs, the overhead of migration, and the uncertainty of VM behaviors. Then, we have presented several RL-based techniques for addressing these challenges. Finally, we have discussed our potential future work.

One of the main challenges in cloud computing is how to efficiently allocate resources. However, there are different sources of uncertainties related to virtual machine (VM) behaviors in clouds, leading to significant waste of computer resources, specifically at light request rates. In this paper, we discuss these sources of uncertainties and propose new techniques based on reinforcement learning to solve such uncertainties to use VM more efficiently. We have conducted a series of experiments to demonstrate the benefits of our approach and shown that our techniques lead to substantial overall energy and cost savings. In addition, reinforcement learning could alleviate effectively the effect of underutilized overhead and therefore could make the existing consolidation techniques much more feasible.

9.1. Summary of Findings

We conclude this survey with five challenges to the community. First, most of the recent papers study a single QoS metric (e.g., delay), which may lead to different solutions. For example, it is well known that the solutions for the throughput maximization and the delay minimization problems are different, leading to interference optimization and routing problems, respectively. Due to the trade-offs, the QoS requirements are usually different, while different services have different preferences in some sense. Second, most of the existing reprising algorithms require a notable amount of overhead in terms of computational and communication resources. The high overhead arises from both the control and data planes and is composed of several factors, including the need for a control plane to collect relevant information, to solve complex optimization problems or to train learning models, and to configure resource management policies, among others, as well as the need for a data plane to execute management policies.

This paper surveys how reinforcement learning (RL) can be used to tackle resource allocation problems of cloud computing. We first present the cloud computing system models and provide several representative policies used in RL, including epsilon-greedy, Q-Learning, actor-critic, and deep RL. We then categorize the existing literature on cloud computing resource allocation into five types, i.e., QoS-aware resource management, resource partition management, resource scheduling, resource allocation with interference, and multi-objective resource allocation. Every category's work is reviewed critically and its challenges are also presented in the respective conclusion. We believe this survey serves as an up-to-date research synthesis that gives new inspiration for researchers to propose better resource allocation strategies using RL in the cloud.

9.2. Contributions to the Field

The major contributions of the proposed work are summarized as follows: 1) An idea to leverage device status in control policies to enhance resource switch efficiency 2) A hierarchical RL technique that combines both the channel scanning phase and the transmission phase 3) An idea to improve the scalability of RL for WMN by decomposing the learning problem into the M learner and the D learners 4) A hierarchical DRL technique with a master-slave architecture to improve the learning efficiency and channel utilization performance

The proposed work aims at developing efficient cost-aware reinforcement learning policies that not only optimize the resource allocation in cloud computing systems but also reduce the overall operational cost. We propose a hierarchical DRL method which improves the learning efficiency and performance. Additionally, controlling resource sharing by defining the QoS level thresholds, the SLA violation rate can be further improved. To model the operational cost of a cloud computing system under various workloads, a multi-objective optimization problem is formulated. Since the resource requirements of workloads are continuously changing and a large number of workloads running on the cloud computing systems with a dominance relationship can result in a high operational cost, we categorize the tasks into specific groups and minimize the resource under-light utilization and the SLA violation rate during the tasks resource allocation. The RL policy will control the resource on-cloud task allocation and reallocation dynamically by taking device status and runtime variation into account.

9.3. Future Research Directions

9.3.1. Incorporation of Spot VM's prices stochasticity. Both hybrid and traditional reinforcement learning algorithms use estimates of future rewards and transition probabilities. The strategies learned by the policies are naturally affected by the presence of uncertainties in the future states or in the rewards. This can lead to a changing situation for the learner as a result of the induced errors of the decision-making model. Such errors could impact the generalization power of the policies learned in previous time-steps. In our contribution, in the learning problems under consideration, the expected future rewards are actual observations which strictly depend on the future states of the market or other exogenous shocks. Although these approaches are particularly data and compute-intensive due to the necessity of reshaping learning, as prices are quite noisy, learning off-line supervised price dynamics based on some (unfortunately non-negligible) experts' feedback could improve the policies learned on-the-job.

In this section, we discuss and lay out some of the future research directions and open problems from the thorough research tested and outlined in the previous sections. It is our hope that the proposals presented in this appendix may serve as guidelines and benchmarks for both the scientific community and industrial sectors with relevant interest in cloud resource management.

10. References

1. L. Tang, Z. Yi, and X. Yang, "Reinforcement Learning for Dynamic Resource Allocation in Cloud Data Centers," IEEE Transactions on Cloud Computing, vol. 9, no. 2, pp. 501-514, 2021.

2. A. Zomaya, and Y. C. Lee, "Energy Efficient Utilization of Resources in Cloud Computing Systems," IEEE Transactions on Cloud Computing, vol. 1, no. 2, pp. 100-111, 2013.

3. H. Li, and Y. Xiao, "Adaptive Resource Allocation for Preemptable Jobs in Cloud Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 8, pp. 2160-2173, 2016.

4. Tatineni, Sumanth. "Climate Change Modeling and Analysis: Leveraging Big Data for Environmental Sustainability." *International Journal of Computer Engineering and Technology* 11.1 (2020).

5. Gudala, Leeladhar, Mahammad Shaik, and Srinivasan Venkataramanan. "Leveraging Machine Learning for Enhanced Threat Detection and Response in Zero Trust Security Frameworks: An Exploration of Real-Time Anomaly Identification and Adaptive Mitigation Strategies." *Journal of Artificial Intelligence Research* 1.2 (2021): 19-45.

6. Tatineni, Sumanth. "Enhancing Fraud Detection in Financial Transactions using Machine Learning and Blockchain." *International Journal of Information Technology and Management Information Systems (IJITMIS)* 11.1 (2020): 8-15.

7. Y. Wang, and X. Zhang, "Adaptive Scheduling Algorithm for Dynamic Resource Allocation in Cloud Computing Environment," IEEE Transactions on Services Computing, vol. 10, no. 1, pp. 31-43, 2017.

8. D. Wu, J. He, and C. Wu, "Mitigating Impact of Short- and Long-Lasting Failures in Large-Scale Cloud Storage Systems," IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 11, pp. 3174-3186, 2017.

9. G. Das, M. Bishnu, and P. Biswas, "Multi-Objective Optimization Based Job Scheduling for Cloud Computing Environment," IEEE Transactions on Services Computing, vol. 10, no. 2, pp. 208-220, 2017.

10. S. M. Khan, and I. Ahmad, "Heuristics-Based Replication Schemes for Fault Tolerance in Cloud Data Centers," IEEE Transactions on Cloud Computing, vol. 6, no. 4, pp. 1154-1165, 2018.
11. W. T. AlZawi, and Y. K. W. Leung, "Cost-Efficient Resource Provisioning for Real-Time Applications in Cloud Computing," IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 86-99, 2020.
12. C. Qu, R. N. Calheiros, and R. Buyya, "Auto-Scaling Web Applications in Clouds: A Taxonomy and Survey," ACM Computing Surveys, vol. 51, no. 4, pp. 1-33, 2018.
13. S. S. Gill, and R. Buyya, "Bio-inspired Algorithms for Resource Management in Edge Computing: A Survey, Taxonomy, and Future Directions," ACM Computing Surveys, vol. 52, no. 6, pp. 1-40, 2019.
14. S. S. Gill, and A. Goswami, "Deep Reinforcement Learning Enabled Resource Management for Fog Computing Environment," Future Generation Computer Systems, vol. 112, pp. 287-300, 2020.
15. L. Wang, and X. Liu, "Adaptive Scheduling with Reinforcement Learning for the IoT Edge-Cloud Environment," Future Generation Computer Systems, vol. 109, pp. 82-94, 2020.
16. J. Xu, and S. Li, "Dynamic Resource Allocation for Cloud Computing Using a Two-Layered Reinforcement Learning Approach," IEEE Transactions on Cloud Computing, vol. 9, no. 1, pp. 263-273, 2021.
17. X. Xu, F. Liu, and H. Jin, "Dynamic Resource Allocation Using Reinforcement Learning for Cloud Computing Applications," IEEE Transactions on Cloud Computing, vol. 9, no. 2, pp. 430-443, 2021.
18. H. Wu, and X. Yang, "Collaborative and Reinforcement Learning Based Energy Management in Cloud Data Centers," IEEE Transactions on Cloud Computing, vol. 9, no. 1, pp. 42-55, 2021.
19. T. Li, and Y. Cao, "Reinforcement Learning-Based Resource Management for Cloud IoT," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4104-4114, 2020.
20. A. Sadeghi, and M. Zafer, "Energy-Aware VM Placement and Migration in Cloud Data Centers Using Reinforcement Learning," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1413-1426, 2021.