ENERGY-EFFICIENT AND COST-EFFECTIVE RESOURCE MANAGEMENT FOR GEO-DISTRIBUTED DATA CENTERS

MOH MOH THAN

UNIVERSITY OF COMPUTER STUDIES, YANGON

FEBRUARY, 2021

Energy-Efficient and Cost-Effective Resource Management for Geo-Distributed Data Centers

Moh Moh Than

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

February, 2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

3.2.2021

…..…… ……..……

Date Moh Moh Than

ACKNOWLEDGEMENTS

Firstly, I wish to thank the Union Minister, Ministry of Transport and Communications; Dr. Myat Lwin, the Rector of Myanmar Maritime University; and Professor Dr. Khin Moh Moh Lwin, Head of Department of Computer Science, Myanmar Maritime University for giving the opportunity to attend this doctoral degree courses at University of Computer Studies, Yangon (UCSY) and giving provision to finish this research.

Secondly, I wish to express very special thanks to Dr. Mie Mie Thet Thwin, the Rector of UCSY for her valuable guideline and supporting to this research.

I want to express my deepest gratitude to my supervisor, Dr. Thandar Thein, the Rector of the University of Computer Studies (Maubin), for providing me invaluable guidance and advice throughout my Ph.D. candidature. Furthermore, my sincere gratitude goes to my supervisor for giving a great deal of her time. Her creative ideas have assisted me in broadening my research skills. She helped me to overcome many crisis situations within research and to finish this dissertation.

I wish to extend special thanks to Dr. Khine Khine Oo, Professor and Course-coordinator of the Ph.D. Courses, for her general guidance, excellent help and encouragement. I appreciate her endless patience, positive outlook, ability to provide advice.

I would like to thank Daw Aye Aye Khine, Associate Professor, Head of the Language Department, for commenting my dissertation. She has read and commented upon everything I have written, and she has pointed out incorrect usage in my dissertation.

In addition, I would like to thank a lot to all my teachers for their mentoring, encouragement, and recommending the thesis.

Moreover, I also want to thank all my colleagues from Ph.D. 10th batch from UCSY for their support, care and mutual help.

Last, but not least, my parents, my brothers and sisters who specifically offered physical support, strong moral, kindness and care during my Ph.D. studies.

ABSTRACT

Cloud computing allows services available to customers on a pay-as-you-go model through geographically distributed data centers (GDCs) located around the world. Almost all Internet services are built on GDCs in order to improve performance and reliability. As cloud computing grows in popularity, so does energy consumption and the cost of powering servers in GDCs. Energy and cost reduction have emerged as major challenges for GDCs.

In this thesis, energy-efficient and cost-effective resource management framework is proposed for minimizing the operational cost and energy consumption of servers, and ensuring the service level objective (SLO). The framework accomplishes electricity price prediction, resource demand prediction, ensuring SLO as well as energy-efficient and cost-effective resource allocation.

Electricity prices of GDCs in multi-regional electricity markets are predicted. Resource demand prediction is performed by using ML techniques for handling dynamic workload nature of the resource amount needed in data centers. Ensuring SLO is conducted to the resource demand prediction system, to ensure the cloud providers have enough resources to meet the resource demand. Energy efficiency factors: allocation policies, power management techniques as well as power models are considered to propose energy-saving resource allocation algorithms for energy-efficient allocation of resources. Energy costs are minimized by delivering the requests first to the data center with cheaper electricity price.

Extensive evaluations are carried out using real-world electricity price data of GDC locations and real-world workload traces. According to the evaluation results, the electricity price prediction model has a promising accuracy. The resource demand prediction model predicts the accurate amount of dynamic resource demand while meeting SLO. CloudSim is used to evaluate the performance of the proposed resource allocation algorithms. This work contributes to reducing energy consumption and it also offers cost-saving.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i

ABSTRACT ii

LIST OF FIGURES ix

LIST OF TABLES xii

LIST OF EQUATIONS xiii

1. INTRODUCTION

1.1 Distributed Data Centers 1

1.1.1 Energy Consumption 2

1.1.2 Resource Management 3

1.1.3 Total Cost of Ownership 4

1.1.4 Service Level Objectives 5

1.2 Motivation of the Thesis 6

1.3 Objectives of the Thesis 8

1.4 Contributions of the Thesis 8

1.5 Overview of the Thesis 9

1.5.1 Proposed Energy-Efficient and Cost-Effective Resource Management Framework 9

1.5.1.1 Electricity Price Prediction 11

1.5.1.2 Resource Demand Prediction 12

1.5.1.3 Ensuring SLO 13

1.5.1.4 Energy-Efficient and Cost-Effective Resource Allocation 14

1.5.2 Cloud Simulation Platform 14

1.6 Organization of the Thesis 15

2. LITERATURE REVIEW

2.1 Modeling Approaches for Electricity Price Prediction 16

2.1.1 Statistical Time Series Models 16

2.1.2 Artificial Intelligence-Based Models 17

2.1.3 Machine Learning-Based Models 18

2.2 Predicting Resource Demand and Ensuring SLO 21

2.3 Power Saving Techniques for Cloud Servers 21

2.4 Resource Management 22

2.4.1 Resource Allocation Mechanisms 22

2.4.2 Energy-Efficient Resource Allocation 23

2.4.3 Resource Allocation for GDCs 23

2.4.3.1 Energy Aware Resource Allocation 24

2.4.3.2 Electricity Price Aware Resource Allocation 25

2.4.3.3 Energy-Efficient and Cost-Effective Resource Allocation 26

2.5 Differences between Existing and Proposed Framework 27

2.5.1 Differences between Existing and Proposed Framework for Electricity Price Prediction 27

2.5.2 Differences between Existing and Proposed Framework for SLO Guaranteed Resource Demand Prediction 28

2.5.3 Differences between Existing and Proposed Framework for Energy-Efficient and Cost-Effective Resource Management 28

2.6 Chapter Summary 29

3. STATE- OF- THE- ART TECHNOLOGY

3.1 Data Centers in Cloud Computing 32

3.1.1 Virtualization in Data Centers 32

3.1.2 Resource Management in Data Centers 34
3.2 Apache Spark Processing Engine 35
3.3 Prediction Perceptions in Data Mining 37
3.3.1 Machine Learning Approach 39
3.3.1.1 Machine Learning Algorithms 39
3.3.1.2 Hyper-parameters Optimization 41
3.3.2 Accuracy of Prediction Methods 42
3.3.2.1 Predictors Accuracy Estimation Techniques 42
3.3.2.2 Predictors Measurement Metrics 44
3.4 Heuristics for Energy-Efficient Management 45
3.4.1 Power Management Techniques 46
3.4.2 Energy-efficient Power Models 47
3.5 Resource Allocation Policies 48
3.6 Chapter Summary 50

4. PREDICTING ELECTRICITY PRICE FOR GEO-DISTRIBUTED DATA CENTERS IN MULTI-REGIONAL ELECTRICITY MARKETS

4.1 Multi-Regional Electricity Markets 52
4.2 Electricity Prices Prediction for Multi-Regional Electricity Markets 53
4.2.1 Electricity Price Prediction System Architecture 54
4.2.1.1 Electricity Price Data Pre-processing 55
4.2.1.2 Electricity Price Prediction Model Selection 55
4.2.1.3 Electricity Price Prediction 56
4.3 Experimental Environment 56
4.3.1 System Specification 56
4.3.2 Experiment Electricity Price Traces 56

4.3.3 Experimental Study 57
4.4 Experimental Results and Analysis 57
4.4.1 Experimental Results of Prediction Models Generations 57
4.4.2 Experimental Results of Selected Prediction Models 62
4.5 Chapter Summary 65

5. SLO GUARANTEED RESOURCE DEMAND PREDICTION FOR CLOUD DATA CENTERS

5.1 SLO Guaranteed Resource Demand Prediction System 67
5.2 System Environment 68
5.2.1 System Specification 68
5.2.2 Experiment Workload Traces 68
5.3 Resource Demand Prediction Model Development 69
5.3.1 Resource Demand Data Pre-processing 70
5.3.2 Resource Demand Prediction 71
5.3.2.1 Hyper-parameter Optimization for DT algorithm 72
5.3.2.2 Resource Demand Prediction Models Construction .. 73
5.3.2.3 Resource Demand Prediction Model Selection 73
5.4 Experimental Results for Resource Demand Prediction 73
5.4.1 Experimental Results of Prediction Models Generations 73
5.4.2 Experimental Results of Selected Prediction Models 79
5.5 Ensuring SLO 81
5.6 Chapter Summary 82

6. ENERGY-EFFICIENT AND COST-EFFECTIVE RESOURCE ALLOCATION FOR GEO-DISTRIBUTED DATA CENTERS

6.1 Experimental Evaluation of Energy-Efficient and Cost-Effective Resource Allocation 84

6.1.1 Simulation Platform 85

6.1.2 System Model 87

6.1.3 Incoming Resource Requests 89

6.2 Analysis for Energy Consumption in FCFS and SJF Allocation Policies 89

6.2.1 Comparison for Energy Consumption of Power Management Techniques in FCFS 89

6.2.2 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS 91

6.2.3 Comparison for Energy Consumption of Power Management Techniques in SJF 92

6.2.4 Comparison for Energy Consumption of DVFS with Four Power Models in SJF 94

6.3 Energy-Saving Resource Allocation Algorithms 95

6.3.1 DVFS Enabled First Come First Serve (DFCFS) Algorithm 96

6.3.2 DVFS Enabled Shortest Job First (DSJF) Algorithm 97

6.3.3 Comparison for Energy Consumption of DFCFS and DSJF 98

6.4 Proposed Energy-Efficient and Cost-Effective Resource Allocation (EECERA) Algorithm 100

6.5 Chapter Summary 103

7. CONCLUSION AND FUTURE DIRECTIONS

7.1 Thesis Summary 104

7.2 Scope and Limitations 105

7.3 Conclusion 106

7.4 Further Research Direction 107

AUTHORS' PUBLICATIONS **109**

BIBLIOGRAPHY **110**

LIST OF ACRONYMS **119**

LIST OF FIGURES

1.1 Proposed Energy-Efficient and Cost-Effective Resource Management Framework 10

1.2 Process Flow of Electricity Price Prediction 12

1.3 Process Flow of Resource Demand Prediction 13

1.4 Process Flow of SLO Analysis 14

3.1 Conceptual Reference Model of Cloud Provider 30

3.2 Types of Clouds based on Service Layer 31

3.3 A Layered Virtualization Technology Architecture 34

3.4 The Apache Spark Ecosystem 36

3.5 Different Scheduling Policies Effects on Task Execution 50

4.1 US Multi-regional Electricity Markets 52

4.2 Proposed Electricity Price Prediction System Architecture 54

4.3 MAE results Comparison of Different Machine Learning Algorithms for the Year 2014 Dataset (for ISONE market) 58

4.4 MAE results Comparison of Different Machine Learning Algorithms for the Year 2015 Dataset (for ISONE market) 58

4.5 MAE results Comparison of Different Machine Learning Algorithms for the Year 2016 Dataset (for ISONE market) 58

4.6 MAE results Comparison of M5P Machine Learning Algorithm for Three Annual Datasets (for ISONE market) 59

4.7 MAE results Comparison of Different Machine Learning Algorithms for the Year 2014 Dataset (for CAISO market) 59

4.8 MAE results Comparison of Different Machine Learning Algorithms for the Year 2015 Dataset (for CAISO market) 60

4.9 MAE results Comparison of Different Machine Learning Algorithms for the Year 2016 Dataset (for CAISO market) 60

4.10 MAE results Comparison of M5P Machine Learning Algorithm for Three Annual Datasets (for CAISO market) 60

4.11 MAE results Comparison of Different Machine Learning Algorithms for the Year 2014 Dataset (for ERCOT market)..... 61

4.12 MAE results Comparison of Different Machine Learning Algorithms for the Year 2015 Dataset (for ERCOT market)..... 61

4.13 MAE results Comparison of Different Machine Learning Algorithms for the Year 2016 Dataset (for ERCOT market)..... 62

4.14 MAE results Comparison of M5P Machine Learning Algorithm for Three Annual Datasets (for ERCOT market) 62

4.15 Actual and Predicted Electricity Price (for CAISO market) 64

4.16 Actual and Predicted Electricity Price (for ERCOT market). 64

4.17 Actual and Predicted Electricity Price (for ISONE market) 64

5.1 SLO Guaranteed Resource Demand Prediction System Architecture 68

5.2 Proposed Resource Demand Prediction System Architecture 70

5.3 Procedure of Resource Demand Prediction..... 72

5.4 MAE Results of Generated Models Using Different MaxDepth Values with MinInfoGain = 0.1 (for DAS) 75

5.5 MAE Results of Generated Models Using Different MaxDepth Values with MinInfoGain = 0 (for RICC)..... 77

5.6 MAE Results of Generated Models Using Different MaxDepth Values with MinInfoGain = 0 (for MetaCentrum) 78

5.7 MAE Comparisons of Prediction Models with Default and Optimal Hyper-parameters 79

5.8 Actual and Predicted CPU Demand (for DAS)..... 80

5.9 Actual and Predicted CPU Demand (for RICC) 80

5.10 Actual and Predicted CPU Demand (for MetaCentrum)..... 80

5.11 Under-provisioning Frequency of Predicted Values for Three Datasets 81

5.12 SLO Analysis Result of Three Workload Traces (one-day interval) 82

6.1 CloudSim Extension Architecture 87

6.2 Comparison for Energy Consumption of Power Management Techniques in FCFS (for DAS) 90

6.3 Comparison for Energy Consumption of Power Management Techniques in FCFS (for RICC) 90

6.4 Comparison for Energy Consumption of Power Management Techniques in FCFS (for MetaCentrum) 90

6.5 Comparison for Energy Consumption of Power Management Techniques in SJF (for DAS) 93

6.6 Comparison for Energy Consumption of Power Management Techniques in SJF (for RICC) 93

6.7 Comparison for Energy Consumption of Power Management Techniques in SJF (for MetaCentrum) 93

6.8 DFCFS Algorithm 96

6.9 DSJF Algorithm 97

6.10 Comparison for Energy Consumption of DFCFS and DSJF (for DAS) 98

6.11 Comparison for Energy Consumption of DFCFS and DSJF (for RICC) 98

6.12 Comparison for Energy Consumption of DFCFS and DSJF (for MetaCentrum) 98

6.13 Energy-Efficient and Cost-Effective Resource Allocation (EECERA) Algorithm 101

6.14 Hourly Prices of Data Centers in Multi-region Electricity Markets 102

6.15 Total Energy Cost (for DAS).. 102

6.16 Total Energy Cost (for RICC) 102

6.17 Total Energy Cost (for MetaCentrum). 103

LIST OF TABLES

4.1 MAE Results of the Chosen Model for Each Market Predicted for the Year 2017... 63

5.1 System Specification 68

5.2 Summary of Experiment Workload Traces 69

5.3 Features of Three Workload Traces 69

5.4 MAE Results of Three ML Algorithms 71

5.5 MAE Results of Generated Models Using Different Combinations of Hyper-parameters (for DAS) 74

5.6 MAE Results of Generated Models Using Different Combinations of Hyper-parameters (for RICC) 76

5.7 MAE Results of Generated Models Using Different Combinations of Hyper-parameters (for MetaCentrum) 77

6.1 Server Types Characteristics 87

6.2 Symbols Description 88

6.3 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS (for DAS) 91

6.4 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS (for RICC) 91

6.5 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS (for MetaCentrum) 92

6.6 Comparison for Energy Consumption of DVFS with Four Power Models in SJF (for DAS) 94

6.7 Comparison for Energy Consumption of DVFS with Four Power Models in SJF (for RICC) 94

6.8 Comparison for Energy Consumption of DVFS with Four Power Models in SJF (for MetaCentrum) 95

6.9 Average Turnaround Time of DFCFS and DSJF Algorithms 99

LIST OF EQUATIONS

Equation 3.1..... 43

Equation 3.2..... 44

Equation 3.3..... 44

Equation 3.4..... 44

Equation 3.5..... 44

Equation 3.6..... 44

Equation 3.7..... 47

Equation 3.8..... 47

Equation 3.9..... 47

Equation 3.10..... 48

Equation 3.11..... 48

Equation 6.1..... 85

Equation 6.2..... 85

Equation 6.3..... 85

Equation 6.4..... 100

CHAPTER 1

INTRODUCTION

Cloud computing is a paradigm focusing on the realization and a long-held perception to provide computing as a service [70]. It allows developers and businesses to gain access the infrastructure and hardware resources whenever and wherever they need. Nowadays, more than ever before, organizations and individuals are shifting their workload to data centers.

Using the services provided by cloud systems has increased in recent years and different meanings have been suggested for cloud computing. Under the interpretation from the National Institute of Standards and Technology (NIST) [59]: "Cloud computing is a model for enabling ubiquitous, 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".

The rest of this chapter is as follows: distributed data centers, energy consumption, resource management, total cost of ownerships and service level objective for Data Centers are introduced. Then the next section discusses motivations, objectives, contribution and overview of this research. This chapter finally concludes with the organization of the dissertation.

1.1 Distributed Data Centers

Cloud computing provides users with utility-oriented IT services and resources on demand, charging on a pay-as-you-go basis, similar to other utility pricing models (e.g., water and electricity). Users will not be charged an upfront fee, and billing will be based on (for example, hourly) usage of cloud resources.

Cloud services are delivered through data center sites with tens of thousands of servers spread through geographical regions. The regional diversity of computing

services provides numerous benefits, including high availability, efficient disaster recovery, consistent consumer access in different areas, and access to various energy sources. Cloud service providers such as Facebook, Google, and Amazon run dozens of distributed data centers to deliver cloud services. The infrastructure resources in data centers, which enable cloud service providers to provide the capacity required to meet the demands of the services, are the core building blocks of cloud computing.

Simultaneously, cloud providers operate multiple GDCs in order to meet the growing demands of cloud users. Because energy costs are determined by current regional electricity prices, distributed data center infrastructure alters cloud control rules.

1.1.1 Energy Consumption

Data centers may be private, where the entire facility is devoted to, or shared with, host applications operated by the facility owner, where the facility owner rents portions of the facility to various service providers. These data centers are designed to serve a large number of users and host multiple disparate applications. Because of the rapid growth of cloud computing, the energy consumption of data centers hosting the cloud's infrastructure has become a global environmental issue as well as a significant cost factor.

The servers' electricity consumption cost increases and it can be higher than the cost of the server hardware itself [10]. The energy consumption is therefore a huge portion of Data Centers' total operating costs. The ever-increasing demand of Cloud-based services raises Data Center energy consumption. According to the US Environmental Protection Agency (EPA), data centers in the US consumed about 1.5 percent of total energy, costing approximately $4.5 billion [26].

A data center's total energy consumption comprises not only the energy consumption of the servers to operate, but also the energy consumption to support infrastructure. The amount of energy consumed should be considered not only for environmental reasons, but also because it is related to other problems. Such

problems consist the cost of electricity, space requirements, air conditioning and heat transfer.

The space requirements also translate into the extensive cooling requirements limit the density of servers. Because of the risk of overheating, servers cannot be constructed for being very compact. The space requirements reduce the possibility to build huge data centers [37]. When building such a farm, it is important to consider the cooling and power demands of large server farms already in the planning stage. Major factors considered today to determine the locations of new server farms include natural cooling resources and available power sources [81].

A range of technologies utilized to make Cloud infrastructures more energy-efficient, including Consolidation, Resource Prediction, Resource Allocation, Temperature-aware Scheduling, Dynamic Voltage and Frequency Scaling (DVFS) and VM live-migration and cooling technologies can significantly improve the capacities of Cloud environments. The energy consumption concerned with the Cloud Computing environment for allocating Cloud resources plays a primary role. Therefore, the Cloud infrastructure providers seek for the mechanism not only to maximize its services without Quality-of-Service (QoS) levels violations but also to achieve the tailored energy consumption through on-demand infrastructure provision.

The rank of the energy consumption of hardware resources in Data Centers is the highest. The Cloud infrastructure providers attempt various techniques at reducing the amount of energy consumed by the physical infrastructure underlying the IaaS environment, ranging from energy-efficient hardware and software design strategies. Data centers' energy consumption will continue to rise unless advanced energy-efficient resource management solutions are developed and implemented.

1.1.2 Resource Management

Cloud provides on-demand access to services and shared resources, hosted in Data Centers located across the world. These resources are offered as a service over a network and are now possible due to innovations across computing

technologies, operations, and business models. For the deployment of cloud computing, effective management of these services and shared resources is essential, harnessing its power from the underlying pool of resources. Cloud resource management is the process of allocating computing resources such as networking, storage, memory and CPU, to fulfill the incoming requests of the cloud users and the infrastructure providers.

Consequently, the management of cloud resources requires complex processes to achieve multi-objective optimization. Heterogeneous resources in large-scale environments present challenges for Cloud tasks and resources management and monitoring. The challenges of resource management range from managing efficient allocation of resources and scheduling jobs to managing uncertainties related with the system and the workload.

There are some obstacles to obtaining efficient Cloud management. These include Service Level Agreement, Dynamism, Cost-effective and Energy efficiency. This thesis seeks the best effort to manage the resources of GDCs in terms of ensuring SLO, energy-efficient and cost-effective manner.

1.1.3 Total Cost of Ownership

Enterprises and service providers are increasingly searching for dependable, reliable, and cost-effective options for the selection of data center locations. The facts that influence the data center selection include geography, and costs, real estate capacity and availability of power. Because hundreds of million dollars are necessary to build and operate a data center annually, the costs are crucial.

The total cost of ownership (TCO) used by high-level managers and financial analysts to understand all the components of costs for data centers, includes capital expenditure (CAPEX) and operational expense (OPEX). CAPEX is composed of the hard costs of land, construction, and equipment, as well as soft costs such as design, engineering and equipment lifecycle. OPEX is composed of costs like power consumption and other utilities, predictive and preventative maintenance, etc. Many studies have found that the costs of corrective maintenance

can be up to ten times the cost of a comprehensive predictive/preventive maintenance program over the lifecycle of a data center and its equipment.

The factors affecting total cost of ownership (TOC) for data centers include infrastructure cost, operational cost, energy consumption cost and maintenance cost. Among them, the cost of energy consumption is the most impact factor and in recent this will grow rapidly as the cloud computing flourishes. As servers of GDCs consume enormous amount of electric power, the cloud service providers face high energy consumption and operational cost. Therefore, energy efficiency has been one of the top issues for cloud service providers, not only for economic reasons but also for the environmental footprint of cloud data centers. Due to the extremely high energy consumption of cloud data centers, electricity costs have become a dominant OPEX, even exceeding CAPEX, i.e., the cost of hardware for the cloud service providers. As a result, lowering electricity costs has become a top priority.

A data center may house a large number of servers that consume megawatts of electricity [64]. Electricity costs in the millions of dollars have been a significant burden on OPEX. As a result, both academia and industry have paid close attention to lowering the electricity costs of data center servers [35].

1.1.4 Service Level Objective

A service-level objective (SLO) is a critical component as an obligation of a service-level agreement (SLA) associated between a customer and a service provider.

SLA enables customers to measure the performance of the service provider and confirms that it is providing services as the contract. SLA is typically established for each IT service area, such as application maintenance or data center. Customers contract SLAs with the service provider for each service; each includes a subset of performance metrics.

The service levels or performance metrics associated with an SLA are referred to as SLO. SLO describes, normally in standards, goals, or measurable terms, the services that a provider provides to a customer within a certain time

frame. SLOs for service providers typically involve system availability to meet request resource requirements and application response time.

While considering energy-efficiency for cloud environment, Service Level Agreement (SLA) [24] is also essential because SLA commitments are the heart of a successful agreement and critical factors in long-term success of the provision ability of cloud computing. SLA is a formal agreement of performance and economic constraints according to non–functional requirements and functional requirements [4] such as CPU capacity, Memory size, etc. under which the Cloud providers need to support. The cloud providers attempt to provide enough resources to meet SLAs, and they pay a money penalty to Cloud users while SLA miss occurs.

Cloud service providers need to ensure that resources are allocated within the limits of the cloud environment in order to achieve the needs of cloud users. It needs to also ensure supporting heterogeneous applications with dynamic nature. The number of resources in data centers needed to allocate the requests is often dynamic due to its dynamic workload nature. Because the nature of resource demand in the cloud is dynamic, resource demand prediction is performed to achieve efficient resource allocation. An effective resource allocation scheme allows to be automatically allocated to each service request, the minimum resources required for acceptable fulfillment of SLAs, leaving the surplus resources free for more VMs to be deployed.

1.2 Motivation of the Thesis

As cloud computing grows in popularity and use, service providers face many resource demand challenges. Cloud computing is typically powered by a large scale virtualized data center engine, each of which houses thousands of servers. With the popularization of cloud computing, research institutions and businesses have begun to outsource their IT and computational requirements to cloud-based on-demand services.

Resource management is becoming increasingly popular as it pays attention for managing GDC resources to increase revenue of the cloud service providers. The resource management scheme is really important for Cloud Computing because

it affects the three basic criteria for functionality, performance, and cost of the Cloud. The Cloud infrastructure is complex with a huge number of shared resources. Resource management requires the complex policies. Cloud resource management processes have several aspects that need to be considered, including workload dynamism, ensuring SLO, energy efficiency and cost minimization.

Resource demands are often dynamic and vary resulted from changes in overall workload. Because of the dynamic nature of cloud and the need to support heterogeneous applications with various performance requirements, efficient resource provisioning is a challenge.

Cloud users' resource requests frequently change over time, making runtime application behavior difficult. Previous knowledge of application behaviors, should not be required by resource management systems. Furthermore, it should not be time-consuming or expensive to develop. Predicting resource demand is critical for efficient resource provisioning of dynamic workload. Resource requirements must be forecasted ahead of time so that the resource management system can adjust resource provisioning to meet customer demands.

Cloud providers must ensure that they have sufficient resources to meet the demands. Alternatively, the providers will require compensating customers, whose performance criteria were not met, resulting in under-provisioning of resources. Then there is the possibility of customer dissatisfaction. The right number of resources needs to be provided to the running applications. Resource over provisioning wastes resources that could be used to benefit others.

Cloud data centers host a lot of applications with various different SLA requirements. Particularly, they need to have stronger and stricter Quality of Service (QoS) guarantees for a commercial success of this computing paradigm. These assurances that are documented in the form of SLA are essential because only then the customers can be assured that they outsource their jobs to clouds.

The computing resources located in the geographically distributed data centers are large-scale store house for the resources and services, where hundreds-thousands of servers are housed in each person. Recently, the acceptability of cloud services is increasing across the world, and increasing the data center load. Further the high-capacity servers and other associated equipment in the datacenters

consume huge amount of energy to fulfill the consumers' computing needs. For data centers, the high cost of energy consumption has become a critical issue. Energy consumption is the key concern in operational costs of cloud systems. In this context, establishing Cloud resource prediction and allocation mechanism is crucial for the Cloud providers.

1.3 Objectives of the Thesis

To meet the increasing demands for cloud services while maintaining energy-efficiency and cost-savings, cloud service providers must implement energy-efficient and cost-effective data center resource management. The major objectives of the proposed elastic provisioning framework are:

- To develop Electricity Price Prediction Model for minimizing energy-related costs of GDCs in Multi-Regional Electricity Markets
- To allocate scarce resources within the limitations of the cloud environment in order to meet the dynamic resource demand
- To support heterogeneous applications with dynamic nature
- To predict the dynamic workload in advance with high prediction accuracy
- To ensure SLO in terms of acceptance for the right amount of VM requests
- To select power management techniques and allocation policies to be energy-efficient resource management
- To minimize energy-related costs of GDCs in Multi-Regional Electricity Markets
- To propose energy-efficient and cost-effective resource management scheme

1.4 Contributions of the Thesis

This thesis proposes Energy-efficient and cost-effective resource management framework with four components and each is for electricity price prediction, resource demand prediction, ensuring SLO, energy-efficient and cost-effective resource allocation.

Prediction of electricity prices is necessary for reducing electricity bills of GDCs in multi-regional markets. It focuses on predicting electricity prices for GDCs in the US multi-regional energy markets considering time zone of the region.

The ability to predict resource demand is a critical feature for data centers to manage the provisioning of dynamic workload nature resources. The CPU resource demand prediction model is built using powerful Decision Tree (DT) machine learning algorithm, and to achieve a high-accuracy prediction model, hyper-parameter optimization for the DT algorithm is performed.

According to the analysis results, proposed predictor is under provisioning predictor. It can almost eliminate data center under provisioning and can reach SLO, by increasing 3% of the maximum predicted value.

Energy-Efficient and Cost-Effective Resource Allocation (EECERA) algorithm is proposed considering energy-efficiency factors and electricity-price diversity of GDCs. EECERA algorithm is established in CloudSim toolkit for allocating the resources of GDCs to support efficiency of energy level and lowering energy cost.

1.5 Overview of the Thesis

Energy-efficient and cost-effective resource management framework is proposed with four components and each is for electricity price prediction, resource demand prediction, ensuring SLO, energy-efficient and cost-effective resource allocation.

1.5.1 Proposed Energy-Efficient and Cost-Effective Resource Management Framework

Cloud providers usually operate the number of GDCs across the world. Cloud data centers are typically large-scale virtualized data centers each with numerous servers. As servers of GDCs consume enormous amount of electric power, the cloud service providers face high energy consumption and operational cost.

Due to its dynamic workload nature, the considerable amount of resources requested to data centers is often dynamic. To handle dynamic workload nature, the ability to predict resource demand is a critical feature to manage the provisioning in

data centers. The provisioning of resources with the appropriate amount of dynamic resource demand while meeting SLOs becomes a critical issue. Consuming a tremendous amount of energy which causes to high operational cost becomes as an emergent issue in cloud computing.

Energy-efficient and cost-effective resource management framework for GDCs is proposed. The proposed framework considers minimizing the electricity cost, exploring the geographical distribution nature and the electricity price diversity of data centers. It ensures to handle the dynamic resource demand while meeting SLO. It manages the resources and requests of data centers can make a great help for reducing energy consumption and the electricity cost of GDCs.

Providing with the detailed architecture of proposed framework, the intended idea of this chapter is to fully specify the proposed provisioning system's algorithms, necessary requirements for proposed system design, and the related theories of the proposed framework. The remaining of the chapter is as follows:

To manage the resources in energy-efficient and cost-effective manner satisfying SLO, the proposed Energy-Efficient and Cost-Effective Resource Management framework is implemented with four components: electricity price prediction, resource demand prediction, ensuring SLO, energy-efficient and cost-effective resource allocation. Figure 1.1 illustrates the system architecture of the proposed framework. The function of each is as follows:

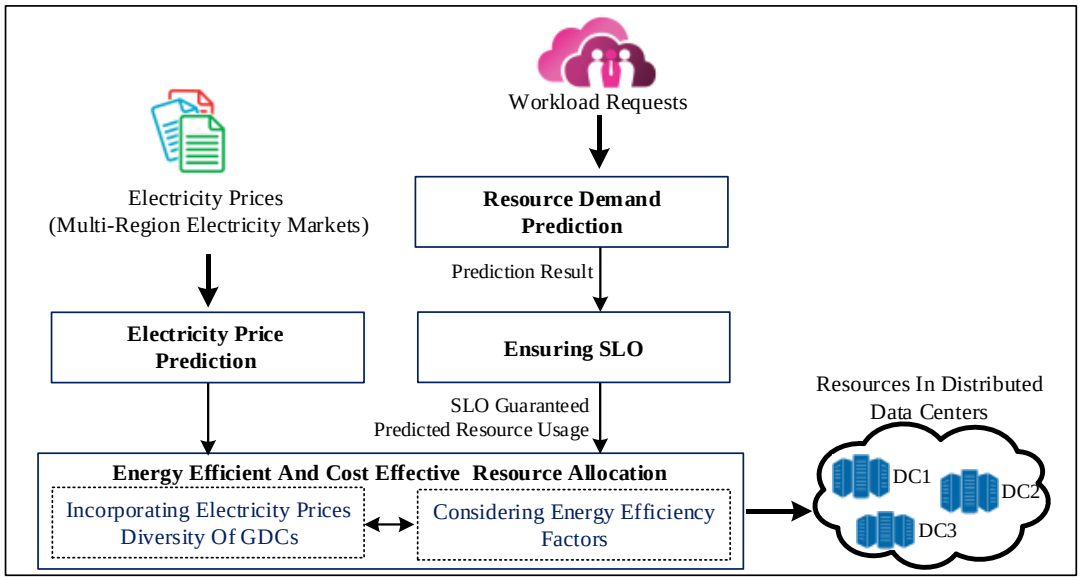


Figure 1.1 Proposed Energy-Efficient and Cost-Effective Resource Management Framework

- *Electricity price prediction* - Electricity prices through GDCs in multi-regional electricity markets are predicted for reducing energy cost by routing the workload requests to the data center favor with minimal electricity price, incorporating electricity price heterogeneity of GDCs.
- *Resource demand prediction* - It predicts the number of resources to decide how many resources to allocate for each request of dynamic workload. The resource usage is estimated and necessary resources are prepared in advance so that the customer can get the necessary services on request.
- *Ensuring SLO* – It keeps track requirements of cloud users as SLO, refers to meet the requested amount of CPU resources. SLO analysis is performed over the prediction results to avoid SLO violations. SLO guaranteed predicted resource usage is applied in resource allocation for preparing the correct type of resources in advance and for planning resource requirements.
- *Energy-efficient and cost-effective resource allocation* – The incoming workload requests are submitted to VMs which are placed on the available hosts in data centers. It allocates the workload requests with proposed Energy-Efficient and Cost-Effective Resource Allocation (EECERA) algorithm in energy-efficient and cost-effective manner. It monitors the power usage of cloud resources and energy consumption of data centers based on energy-saving techniques. It calculates the energy costs for data centers and attempts to minimize the total energy cost based on price diversity of GDCs.

1.5.1.1 Electricity Price Prediction

GDCs electricity prices are predicted to minimize the electricity bill of GDCs. This section focuses on predicting GDC electricity prices of multi-regional electricity markets in the United States. Models are generated on three annual price datasets for each of three markets with three algorithms: M5P, Linear Regression, and Decision Table and then the most accurate prediction model is chosen for predicting the electricity prices for each of three GDCs. The process flow of electricity price prediction is displayed in Figure 1.2.

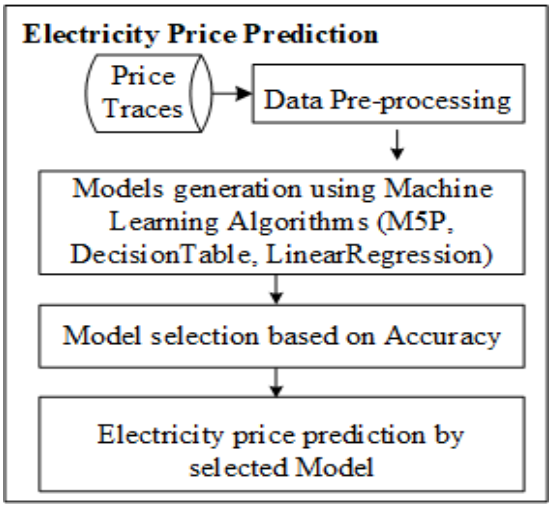


Figure 1.2 Process Flow of Electricity Price Prediction

1.5.1.2 Resource Demand Prediction

Prediction of resource demand is a critical feature for efficient resource management of dynamic workload. The CPU resource demand prediction model is built on the powerful Decision Tree (DT) machine learning algorithm, and hyper-parameter optimization for the DT algorithm is used to achieve a high-accuracy prediction model. By analyzing the accuracy of the models generated with all possible combinations of hyper-parameters, the resource demand prediction model with the lowest error rate is chosen. The prediction model is evaluated using workload traces, and the results show that hyper-parameter optimization can significantly reduce prediction error. Figure 1.3 depicts the procedure for predicting resource demand.

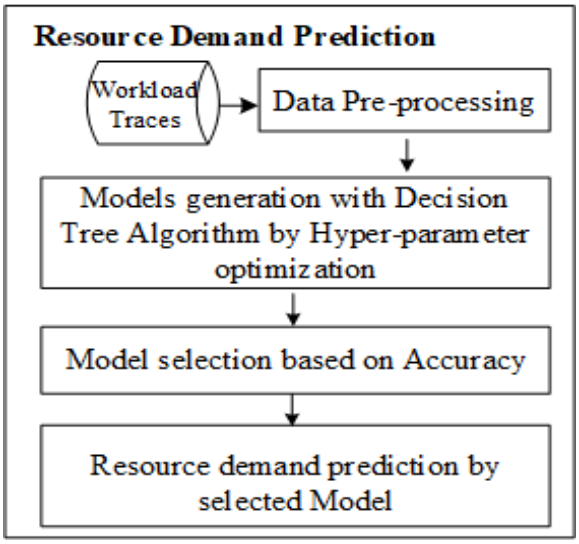


Figure 1.3 Process Flow of Resource Demand Prediction

1.5.1.3 Ensuring SLO

Service Level Agreement (SLA) [24] is also essential because SLA commitments are the heart of a successful agreement and critical factors in long-term success of the provision ability of cloud computing. SLA is a formal agreement of performance and economic constraints according to non–functional requirements and functional requirements [26] such as CPU capacity, Memory size, etc. under which the Cloud providers need to support. The cloud providers attempt to provide enough resources to meet SLAs, and they pay a money penalty to Cloud users while SLA miss occurs.

The cloud service provider must ensure they have sufficient resources to meet customer demand. Alternatively, the cloud provider will require compensation to these customers whose performance requirements have not been met. The prediction accuracy of the predictors is under provision or over provision. The process flow of SLO analysis is shown in Figure 1.4.

According to the prediction results, the proposed predictor is more under provision than over provision. Therefore, SLO analysis is performed by increasing a small amount of predicted value to avoid the under provisioning of the prediction. If the increasing value does not meet the SLO, increase again to meet the SLO of the cloud provider.

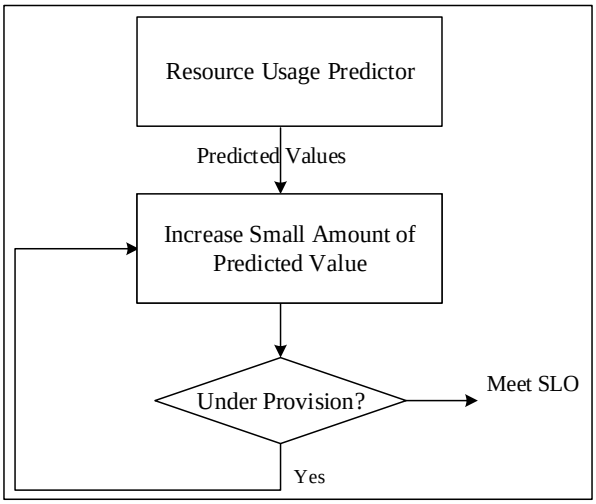


Figure 1.4 Process Flow of SLO Analysis

1.5.1.4 Energy-Efficient and Cost-Effective Resource Allocation

Energy-efficient and cost-effective resource allocation means allocating the incoming requests to the servers through VMs while optimization of energy use and price diversity awareness, in particular toward goals of saving energy consumption and lowering cost. We validated Dynamic Voltage Frequency Scaling (DVFS) is the most power saving one among three power management techniques. We experimented two resource allocation algorithms: DFCFS (DVFS enabled First Come First Serve) and DSJF (DVFS enabled Shortest Job First) using three real workload datasets and found that DSJF consumes less energy than DFCFS. Energy-efficient and Cost-effective Resource Allocation (EECERA) algorithm is proposed based on DVFS and SJF allocation policy to be energy-efficient and incorporating the price diversity of GDCs to be cost-effective. Since the geographically distributed data centers have different electricity prices and processing capabilities, an intuitive approach is to distribute more workloads to the data center with lower price.

1.5.2 Cloud Simulation Platform

Handling on the real cloud is difficult and expensive because experiments are performed in a controlled and dependent environment. As a result, it is assumed that using simulation tools is a preferred choice [9]. Simulators provide dynamic

and flexible environments and configuration. They allow researchers and system administrators to learn and improve the reliability and scalability of the real-time cloud environment.

This section discusses some of open-source simulators for Cloud computing. The benefits of running cloud simulators, and some of them are: No capital cost involved, Leading to better results, Easy to learn, and at an early stage, risks are assessed [50].

There are various cloud simulation tools: CloudSim, GreenCloud, MDCSim, DCSim and iCanCloud, etc. CloudSim is a well-recognized tool that is essentially a toolkit for cloud scenario simulation [73]. In fact, CloudSim allows users to get a thorough understanding of cloud scenarios without having to worry about the specifics of low-level implementation [11].

1.6 Organization of the Thesis

This dissertation is structured as follows: Chapter 1 introduces problem issues in resource management for saving energy and electricity cost of GDCs in overview, highlights the proposed framework for reducing the energy consumption and operational cost of servers, ensuring the service level objective (SLO) and presents motivating factors, contributions and objectives of this research. The summarization of the current research works in Cloud Computing to achieve Energy-efficient and cost-effective resource management for GDCs is discussed in Chapter 2. Next, the related theories of electricity price prediction, resource prediction and allocation in cloud data center are presented in Chapter 3. Electricity price prediction system for GDCs in multi-regional markets can be seen in Chapter 4 and the next chapter, Chapter 5, SLO guaranteed cloud infrastructure resource demand prediction model development is presented. Chapter 6 discusses energy-efficient and cost-effective resource allocation. The experimental results for the prediction models and allocation mechanism are put together in their related chapters. Finally, Chapter 7 summarizes the proposed system and sets some conclusion and finally points out with future research directions.

CHAPTER 2

LITERTATURE REVIEW

Energy-efficient and cost-effective resource management problem in cloud computing has received growing attention. This chapter discusses state-of-the-art research and techniques which reduce a large portion of consumed energy and low electricity cost of data centers. The previous research of academic literature is reviewed in three portions as the proposed framework is carried out for three different tasks. Previous works of electricity price predictions are studied. Previous research relating to predictions of data center resource usage and SLO guarantee are studied. Moreover, previous studies of energy-efficient and cost-effective resource allocations are also premeditated.

2.1 Modeling Approaches for Electricity Price Prediction

A variety of ideas and methods have been tried for electricity price prediction, with various degrees of success. There is no common method for electricity price prediction which can be used for any operator and market. This section outlines many research and different approaches developed for analyzing the price data and predicting future electricity prices for various electricity markets. This aims to provide an overview for the literature and a brief of the previous review publications of electricity price prediction.

2.1.1 Statistical Time Series Models

The authors [44] use auto regressive (AR) model with lags of 1 day, 2 days and 7 days (1 week) taken into account for seasonal dynamics and residual autocorrelation to predict electricity price in short-term. The paper [72] proposed a hybrid approach to predict day-ahead electricity prices. In which a wavelet transform supports a set of ‘better-behaved’ time series, an auto regressive integrated moving average (ARIMA) model is used to perform a linear forecast, and a radial basis function (RBF) network is used to correct the prediction error of the wavelet-ARIMA forecast. Lagarto et. al [52] described an interesting method that

combines multi-agent modeling and time series elements. The next day's hourly prices are predicted using an ARIMA model applied to the conjectural variations of the firms in the Spanish electricity market. They found that the conjectural variations price estimation performs better compared to the naïve method and a pure ARIMA model.

Huurman et. al [39] considered generalized autoregressive conditional heterokedasticity (GARCH) models. In the density estimation of Scandinavian day-ahead electricity prices, they discovered that models statistically augmented with weather forecasts perform better specifications that ignore this information. The paper [41] demonstrated that a better in-sample fit can be obtained by filtering average daily prices with some 'reasonable' outlier detection procedure, then calibrating the stochastic and seasonal components of the model to spike-filtered data. Some authors recommend filtering out spikes before calibrating neural network or AR models when predicting hourly day-ahead prices.

The authors [33] performed hourly prices prediction of California and Spain electricity markets by applying two GARCH models. For Spain market they used 15 months hourly data and for Californian market they used 12 months hourly data. The experimental result was about 9% prediction error on both California and Spain electricity markets.

2.1.2 Artificial Intelligence-Based Models

The authors [77] presented a multi-layer Gated Recurrent Unit (GRU) Neural Network based method for predicting electricity price to apply time series information in neural network architecture fully. They proposed to use multi-layer GRU Recurrent Neural Network (RNN) to predict price. They trained algorithms with three years rolling window and did a comparison to the results of RNNs. Extensive performance analysis is carried out for both daily and monthly electricity price prediction. Their experimental results showed that for the Turkish day-ahead market, three-layered GRUs outperformed all other neural network structures (Long Short Term Networks, neural networks, and Convolutional Neural Networks) and state-of-the-art statistical techniques significantly. By using three-layered GRUs, they reached the best results of MAE: 5.36 Euros/MWh.

The authors [46] proposed a new predicting strategy for short-term electricity price prediction, which complex with time dependent behaviour, nonlinear and volatile. A two-stage feature selection algorithm and an iterative training algorithm are part of their prediction strategy. Two filtering stages are included in the feature selection algorithm to remove irrelevant and redundant candidate inputs. The improved iterative training algorithm is composed of two neural networks, with the first's output serving as one of the inputs to the second. The proposed predicting strategy is applied to the markets of Pennsylvania, New Jersey, and Maryland (PJM). Their proposed improved iterative neural network model outperforms the RNN, PCA with CNN, MLP with LM, CA with CNN, and SVC-RFE with CNN models, which have MAPE values of 9.66%, 7.15%, 8.12%, and 5.86%, respectively.

The paper [3] used training algorithms for RNN based on extended Kalman filter to predict electricity price, for one-step and n-step ahead. It also included the stability proof using Lyapunov methodology. The proposed prediction scheme is applied using data from the European power system by mean of the one-step ahead and n-step ahead prediction.

2.1.3 Machine Learning-Based Models

The researchers [84] researched predicting one-day ahead price of the Norwegian electricity market. They collected from 2001 to 2009 9-year-datasets based on historical consumption, price, weather change and reservoir data. They compared the results of different machine learning algorithms: Model Trees, Linear Regression, Multilayer Perceptron, Gaussian Process, Support Vector Machines (SVM), Radial Basis Function Network and Automated Design of Algorithms. They discovered that SVM gives the best result with average error 3.14%.

J. C. R Filho et al. [31] used machine learning techniques based on clustering and decision trees to predict the short-term price of four electricity markets: Center-east, Northeast, North and South market in Brazilian. The overall short-term price prediction accuracy is 94.02%, 87.87%, 97.01% and 89.55% respectively for four electricity markets.

The paper [1] predicted electricity prices of GDCs in four markets: Independent System Operator New England (ISO-NE), New York Independent System Operator (NYISO), Electric Reliability Council of Texas (ERCOT) and Electricity Market of New Zealand (NZ) to minimize the energy cost of GDCs. They used Gaussian random variables with some estimated variance and known means to predict future prices within a one-day 24-hour time frame. This model employs linear regression to forecast variances based on previous prices from the same day last week, the day before yesterday, and yesterday.

2.2 Predicting Resource Demand and Ensuring SLO

Resource prediction is a basic need to support service management tasks and to design Cloud Computing solutions, such as provisioning and deployment, to select scheduling policies and load balancing, and for capacity planning. Resource demand prediction is needed for efficient resource management of dynamic workload. Resource prediction exploits the relationship between past observation and future prediction. The prediction model constructions are experienced by using ML techniques. Several research concerning cloud data center resource demand prediction with different machine learning algorithms and ensuring SLO are discussed in this section.

With advancements in prediction models by machine learning applications, several studies have considered cloud data centers resource demand prediction applying such models in the literature, e.g., a predictive model for workload forecasting [71], workload prediction and characterization [47], and traffic reduction [80]. In [68], the researchers applied machine learning–based techniques to predict the daily operational workload: the amount of power consumption (PC) and the number of physical machines (PMs) required fulfilling the demands of the cloud data center. They investigated three different methods: Polynomial Regression, Support Vector Regression, and Random Forest Regression (RFR). The results showed that RFR provides the best performance, with a minimum root-mean-square error of 11.68 for PMs and 4869.08 for PC prediction.

The paper [79] studied for predicting the future resource demand based on the characteristics of a tenant in a multi-tenant cloud environment. Their prediction

system deployed statistical methods and data mining techniques: Logistic Regression, Multilayer Perceptron, Support Vector Machine, Probabilistic Neural Network and Reduced Error Pruning Tree (REPTree) to predict the future resource demand for each service tenant. They observed that REPTree produced better result than other ML techniques. They applied resource demand prediction in resource allocation for planning resource requirements of service tenants.

The authors [90] presented WorkloadCompactor, a new system for optimizing rate limit selection that can compact more workloads onto a server while achieving SLO. It used workload traces to automatically select rate limits while also selecting the server on which workloads will be placed. To ensure tail latency SLO, the system enforces rate limits, storage and network priorities, and uses network calculus equations to determine whether workloads can be placed together while meeting their SLOs. It uses this flexibility to better load workloads onto servers as workloads enter and exit. It is used in conjunction with their scalable workload placement algorithm to place workloads on servers orders of magnitude faster than traditional first-fit policies. Experiments with 1000 workloads assigned to servers revealed that WorkloadCompactor decreases the number of required servers by 30-60%.

The paper [13] used an Autoregressive (AR) prediction model to accommodate allocation decisions based on the predicted resource demand and proposed a management algorithm for dynamic VMs allocation to physical hosts. They combined bin packing approach with time series prediction techniques in their algorithm. The algorithm adapts to changes in demand and migrates virtual machines (VMs) between physical hosts while providing probabilistic Service Level Agreement (SLA) guarantees. It solved for over-provisioning problem over the predicted resource demand and their algorithm could provide a specified rate of SLA violations by reducing the amount of physical capacity required for a given workload. The algorithm efficiency was assessed using data centers workload traces. Their results indicated that they could improve SLA guarantees and promote better efficiency of resource use by leveraging the AR model. Compared to the static allocation, their algorithm could achieve a substantial reduction in resource consumption up to 50% and could decrease the number of SLA violations.

The authors [61] presented a method for adaptive provisioning based on ML predicting and aimed to reduce over-provisioning. They used SVM regression and fine-tuned its parameters before applying the SVM regression model with various kernel functions: polynomial kernel, RBF kernel, and normalized polynomial kernel, and different parameters configuration to achieve accurate prediction. They combined the forecasting model to estimate the amount of resources to be provisioned, and to fulfill the SLO according to the predicted load. They evaluated their system for 24-hour test interval load of a web server and the results showed that SVM model with normalized polynomial kernels is the best for the amount of over-provisioned resources, and SLO violations.

2.3 Power Saving Techniques for Cloud Servers

The authors [5] proposed an energy efficient VM placement algorithm for cloud data centers that starts with the most energy efficient server. They used three power management techniques: dynamic voltage frequency scaling (DVFS), power-aware (PA), and non-power-aware (NPA), to their algorithms to reduce the power consumption of the hosts. Their algorithm achieved 9%, 23% and 23% more power-efficient than the Minimum Power Difference algorithm, Best Resource selection and Round Robin algorithms.

B. Ahmad et al. [2] addressed to save energy with DVFS technique for the cloud gaming data centers. The DVFS technique is compared against static threshold detection techniques and non-power aware. CloudSim simulation platform is used to provide users with the ability to perform the desired tests for the implementation and evaluation of the proposed experiments. They experimented with game traces as workload. The results demonstrated that DVFS approach consumes less energy than non-power aware or static threshold consolidation technique.

The paper [42] compared four different power models over IaaS Cloud infrastructure: square root, linear, square, and cubic models. They confirmed that the cubic power model uses less power than other models.

2.4 Resource Management

Effective management of virtualized resources is a challenging process for providers, as it often involves choosing the right resource allocation from a wide variety of alternatives. The researchers consider the administrative architecture for not only allocation perspective but also assessment, performance, availability, and energy consumption implications. This section discusses related research and technologies about resource allocation in data centers to be energy-efficient and cost-effective manner.

2.4.1 Resource Allocation Mechanisms

Throughout cloud computing, allocation of resources is critical to the overall performance of the whole system as well as the level of customer satisfaction provided by the system. Furthermore, by ensuring the utmost satisfaction for the customer, the service provider should always ensure the profits that they achieve. The resource allocation will also be economical on both end-user and service provider viewpoints.

The paper [6] discussed the cloud resource allocation algorithms: Shortest Job First (SJF) and First Come First Serve (FCFS). This paper also conducted to evaluate and assess the performance of these algorithms on CloudSim according to the metrics: average turnaround time and average waiting time. Their results showed that SJF takes minimum average turnaround time and average waiting time compared to FCFS.

W. Y. Lin et al. [55] developed a new resource allocation algorithm by a second-priced auction mechanism. In addition, they proposed a framework for dealing with capacity distribution. They ensure efficient capacity allocation under simple decision rules and generate appropriate revenue for the Cloud Service Provider under the proposed mechanism.

K. C. Gouda et al. [34] described the dynamic resource allocation by using priority algorithm that determines the allocation sequence for various requested jobs among different users after taking into account the priority based on some optimum threshold determined by the Cloud provider. The developed resource allocation

algorithm is based on various parameters such as cost, time, and the number of requested processors, etc.

2.4.2 Energy-Efficient Resource Allocation

The authors [12] proposed an energy-aware virtual machine allocation algorithm for allocating cloud data center resources to user tasks saving energy. They proved that their proposed algorithm reduces the amount of energy consumed by data centers.

The researchers [78] proposed a VM allocation algorithm based on interior search: Energy-Efficient Interior Search (EE-IS) for reducing energy consumption and maximizing resource utilization. The proposed algorithm was implemented on CloudSim, and compared to the energy consumed by the Best-fit Decreasing (BFD) algorithm and the Genetic Algorithm was compared (GA). They demonstrated that their proposed EE-IS saved an average of 30% of energy when compared to the energy consumption of BFD and GA.

To handle heterogeneous workload, S. Malik et. al [58] presented an energy-efficient load balancing and resource allocation algorithm. They met their resource demands with minimal resource waste, resulting the reduction in energy consumption. If the requested resources are available, the user requests are routed to the scheduling queue and scheduled. The requests are then categorized based on their performance characteristics and resource demands. In addition, the amount of resources needed by cloudlets is calculated, and appropriate machines are chosen based on the configuration. The goal is to select a set of three machines that consume the least amount of energy. They efficiently calculated the number of active physical machines as well as the skewness factor for the selected machines. Finally, the machine with the least skewness among resources is assigned. They validated their proposed algorithm using CloudSim toolkit.

2.4.3 Resource Allocation for GDCs

Because cloud computing attracts on-demand services, it has resulted in the creation of geo-distributed data centers (DCs) with thousands of computing and

storage nodes. The popularity of cloud computing services has resulted in large computing infrastructures that are difficult to manage and extremely expensive to run. The operational costs of large infrastructures are dominated by power supply, in particular, and several solutions must be implemented to reduce these operational costs and make the entire infrastructure more energy-efficient.

2.4.3.1 Energy Aware Resource Allocation

The authors [48] investigated various parameters that influence energy and carbon costs for GDCs sites. They used the PUE model of a data center as a dynamic function of IT load and hourly temperature changes outside. They evaluated various energy and carbon-aware dynamic VM placement approaches that take into account the availability of renewable, dynamic PUE, and changes in energy consumption have the greatest effect on saving total energy and carbon costs while also lowering brown energy usage. They also measured SLA violations for their proposed VM placement approaches. They also found the minimum values for SLA violation (number of rejected VMs) as compared to the competitive algorithms.

S. Rawas et al. [66] focused on an energy-efficient approach to allocating data-intensive workloads in GDCs. They proposed the Location-Aware and Energy-Efficient (LAEE) workload allocation. In the LAEE algorithm, they applied the genetic algorithm to find an optimal solution considering both the execution time and the transfer time of the data files from the storage location to the computing servers. They combined DVFS technique in their algorithm to save the energy consumption of servers. The efficacy of their proposed allocation method is evaluated using CloudSim. The results showed significant enhancements in the user's QoS (minimizing turnaround time) and saving the energy consumption of data centers.

The study [69] proposed two DVFS-enabled host selection algorithms: for VM placement with a cluster selection strategy aimed at balancing the load among the servers and dynamically tuning the cooling load based on the current workload.

This research looked at three parameters: processor power dissipation in relation to operating frequency, cooling device power consumption, and power usage effectiveness (PUE). In the placement algorithms, the CO2 emission rate and PUE are used to select data centers and clusters, with the goal of reducing the overall carbon footprints. Load balancing is accomplished by identifying a feasible server with a minimal operating frequency for the current workload while maintaining the desired quality of service, with the goal of reducing hot spots in CPU heat dissipation, which have a direct impact on hardware performance and lifetime; the effects of static and dynamic PUE on placement decisions, as well as cooling load power impacts, are analyzed. According to the results, C-FFF reduces power by 2% more than C-PEF.

2.4.3.2 Electricity Price Aware Resource Allocation

The paper [8] studied the data placement, task assignment, routing to lower the overall operational cost in GDCs and data center resizing for big data applications. They investigated the promising cost efficient methods by exploring the advantage of DVFS. They exploited the problems of dynamic frequency scaling technique, task placement among GDCs and the electricity price diversity in order to save the cost. They described the data processing using a two-dimensional Markov chain, which was based on joint optimization expressed as a mixed-integer nonlinear programming (MINLP) problem.

The paper [65] investigated the problem of reducing total electricity costs in a multi-market environment while maintaining service quality in response to location and time variations in electricity prices. They approximated the problem as a linear programming formulation, which they solved using a fast polynomial time method. Through extensive evaluations based on real-life electricity price data for multiple IDC locations, they demonstrated the efficacy of the designed algorithm as well as the total electricity cost reduction.

The paper [87] proposed a spatial task scheduling and resource optimization (STSRO) method to reduce total cost by scheduling all arriving tasks of heterogeneous applications cost-effectively. STSRO enhances use of spatial diversity in distributed data centers. It determines the best configuration for each

server in each data center and jointly specifies the best distribution of all arriving tasks among multiple ISPs. It considers factors such as electricity cost, solar radiation, wind speed, and on-site air density, and can intelligently schedule all arriving tasks to distributed data centers within their delay-bound constraints. The proposed simulated annealing-based bat algorithm solves the cost minimization problem as a constrained optimization problem in each time slot (SBA). According to trace-driven experiments, STSRO achieves lower total cost and higher throughput.

2.4.3.3 Energy-Efficient and Cost-Effective Resource Allocation

The authors of the paper [86] proposed temperature-aware workload management for geo-distributed datacenters. They developed a distributed algorithm based on an m-block alternating direction method of multipliers (ADMM) algorithm, which extends the traditional 2-block algorithm. Temperature diversity is used to reduce cooling energy consumption around the world. They discovered that using trace-driven simulations with historical temperature data, real-world electricity prices, and an empirical cooling efficiency model, they can consistently deliver a 15% – 20% cooling energy reduction and a 5% – 20% overall cost reduction for geo-distributed clouds.

S. Rawas et. al [67] He investigated the problem of VM placement decision in geo-distributed DCs, which results in less access latency, less energy consumption, and less CO2 emissions, with the goal of reducing the operational costs of large-scale cloud providers. The proposed Power and Cost-aware VM Placement (PCVM) model finds an appropriately suitable host machine to process any user request by taking into account WAN latency, PUE, DC CO2 emission rate, and energy consumption. PCVM-NWP is a machine-learning prediction model that predicts the weights of the proposed multi-objective function to improve the performance of the PCVM model. Extensive simulation is performed using the CloudSim simulator to evaluate the proposed model, and the results show that the proposed PCVM model can improve operational cost minimization by reducing DCs power consumption.

In the paper [51] adapted EcoMultiCloud, a flexible load management tool that implements multi-objective load management strategies, to the presence of renewable energy sources to power data centers. They aimed to reduce costs in the load allocation process in a multi-data center scenario, where VMs are assigned to a specific data center while taking into account both energy cost variations and the presence of local renewable energy production, in order to reduce the energy bill. When virtual machines (VMs) are assigned to a data center of the considered infrastructure, costs are reduced by taking into account both energy cost variations and the presence of renewable energy production. The paper proved the viability of incorporating renewable energy to reduce the operational costs of a complex infrastructure like as data centers with geographically distributed sites.

2.5 Differences between Existing and Proposed Framework

The aim of this subsection is to compare some of the existing solutions and proposed system. The comparison can be classified into three different categories based on Electricity Price Prediction, SLO Guaranteed Resource Demand Prediction, as well as Energy-Efficient and Cost-Effective Resource Management are presented.

2.5.1 Differences between Existing and Proposed Framework for Electricity Price Prediction

There is no common method for electricity price prediction which can be used for any operator and market. This thesis illustrates electricity price prediction for geographically distributed data centers in US multi-regional markets and considers about the time zone of the regions as [1].

The performance of three ML algorithms are analyzed and compared the on experimented electricity price datasets of three markets. The most suitable prediction models are chosen to predict the prices and the predicted electricity prices of GDCs are applied in the proposed resource allocation algorithm for saving the cost by incorporating the electricity price diversity along through regions.

2.5.2 Differences between Existing and Proposed Framework for SLO Guaranteed Resource Demand Prediction

The existing studies investigated different machine learning algorithms and selected the best ML algorithm which is suitable for their experimented workload datasets. However, they did not consider enhancing the accuracy of the resource demand prediction model.

In the research, three ML algorithms are examined and the most accurate DT algorithm is selected to develop the CPU resource demand prediction model for data centers. The DT algorithm is enhanced by hyper-parameter optimization to find the model that can predict more accurate result. Apache Spark™ is used as backend processing engine, as it is better suitable for iterative applications, such as Data Mining and Machine Learning [7, 19, 32].

SLO analysis is performed in the prediction system and guaranteed SLO to meet the requested amount of CPU cores. This thesis focuses on a higher priority to avoid under-provision than to avoid over-provision solved in the existing papers [13, 61] since the under-provision may be more likely to cause SLO violations.

2.5.3 Differences between Existing and Proposed Framework for Energy-Efficient and Cost-Effective Resource Management

Each of the existing papers mentioned in section 2.3 and 2.4 focused on minimizing the turnaround time for each request, power consumption of servers, energy and cost of data centers, respectively.

This thesis proposes the resource management framework to save consumed energy and cost of data centers while satisfying SLO. For saving energy, it considers energy-efficiency factors such as resource allocation policies and power management techniques with four energy-aware power models. For lowering cost, it also exploits the electricity prices diversity of GDCs. In the proposed resource allocation algorithms, allocation policies are applied to minimize the turnaround time as well as power management techniques are analyzed to save the power consumption of the servers.

In the thesis, three power management techniques and four power models for two allocation policies are analyzed and compared to select the most power-efficient. According to these comparisons, this thesis proposes two energy-saving resource allocation algorithms: DVFS enabled shortest job first (DSJF) and DVFS enabled first come first serve (DFCFS). We extend CloudSim to allow for energy-efficient and cost-effective resource allocation for geographically distributed data centers, as well as to evaluate the performance of the proposed algorithms.

2.6 Chapter Summary

The aim of this chapter is to highlight the actual requirements and efforts of the research area. There are much research works to resource management for data centers. The previous research is reviewed in three portions as the proposed system is carried out for three different tasks. Previous research relating to predictions for electricity prices of GDCs, resource demand prediction, ensuring SLO, and resource allocation for minimizing energy and cost of GDCs are reviewed.

CHAPTER 3

STATE-OF-THE-ART TECHNOLOGY

Cloud provides the various cloud services with the emergence of virtualization technology. It provides Infrastructure as a Service (IaaS) for on-demand computing and pay-as-you-go models to the end users through geo-distributed data centers (GDCs) scattered across diverse geographies. GDCs power consumption is significantly increasing as cloud computing services are dramatically growing. Consuming a large amount of energy causes high operational cost, and it has become an emergent issue in the cloud computing.

The organization of the chapter is as follows:

The first section, cloud computing technology is described as a brief. The virtualization technology overview is discussed in the next section. Then prediction using machine learning techniques are discussed. The comprehensive energy-efficient management techniques are also presented in the next section.

The conceptual reference model of cloud deployment model is described in Figure 3.1. Cloud consumer could be a bank or any other user who would take services on the cloud. Cloud provider would be a system integrator who would integrate multiple parties' offerings to provide a solution and sign agreement contracts with cloud consumers.

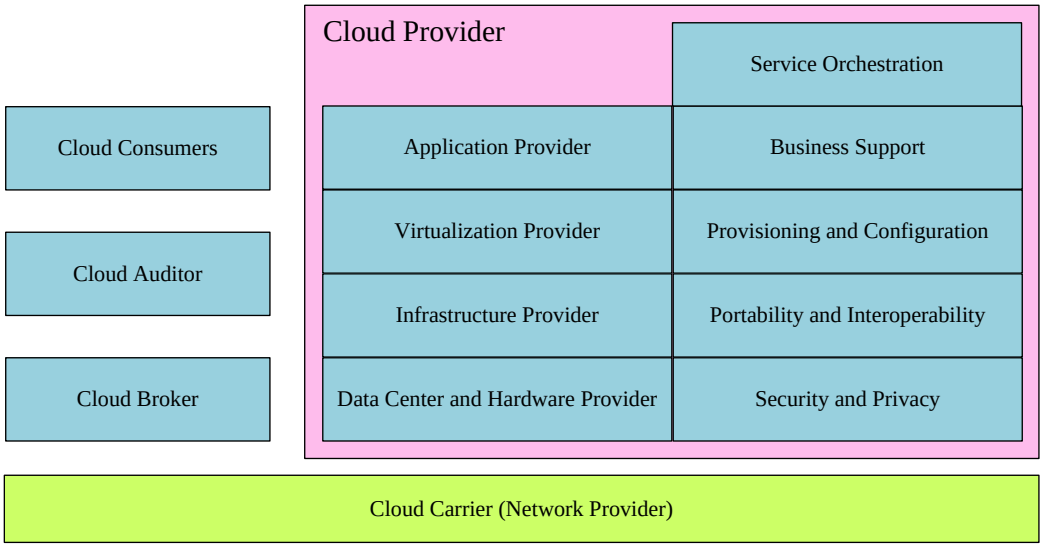


Figure 3.1 Conceptual Reference Model of Cloud Provider

These parties would be (a) Network provider (b) Data center and hardware provider (c) Infrastructure (software) providers (d) Virtualization (software) providers and optionally (e) Application providers. The cloud carrier would be the provider of network infrastructure to connect various branches of banks to the data center. Cloud auditor could be a professional audit company that can conduct data privacy, independent security, and operating process and deployment infrastructure performance audit. Cloud brokers would provide value-added services by aggregation or arbitration at the top of cloud providers ' business services.

Cloud computing is accomplished in a three-tiered structure [43] Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) from high to low level. The types of clouds based on service layers are presented in Figure 3.2.

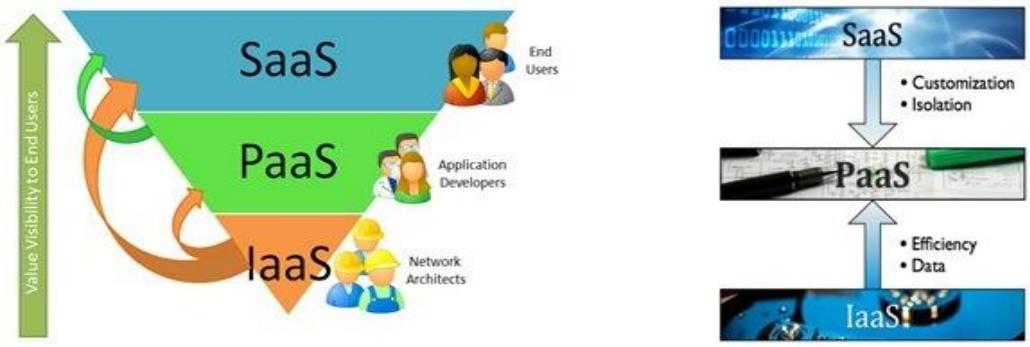


Figure 3.2 Types of Clouds based on Service Layer

There is Software as a Service (SaaS) at the highest layer of cloud system. SaaS is an old cloud computing concept that provides users with on-demand software. Platform as a Service (PaaS) provides developers with a computing platform with a pre-installed operating system on which to create their own software. Developers do not need to be concerned about the operating system or the underlying hardware. At the lowest level of a cloud computing system, there is Infrastructure as a Service (IaaS). It is one essential Cloud Computing deployment model which offers massive heterogeneous virtualized resources and infrastructures such as servers, hardware, memory, CPU, storage, network, and virtual machines (VMs) to cloud consumers. It provides on-demand virtualized resources like storage, computing, and communication [75]. Cloud infrastructure is not managed

by the cloud user, e.g. the user has no control over the delegated resources in data centers.

3.1 Data Centers in Cloud Computing

Data centers are the bedrock of today's IT infrastructure. A data center, on the other hand, consists of thousands to tens of thousands of server machines that collaborate to provide services to clients. These data centers may be owned or federated by the cloud provider. Because of the widespread success and expansion of cloud computing, data center operators are geographically distributing their data center locations. The resources of a cloud computing system are often geographically distributed.

Clients benefit from geographically distributed data centers. Reduced operating expenses through time-of-use (TOU) electricity pricing, on the other hand, is a strong motivator for data center operators to geographically distribute their data centers, and lowering electricity costs is now a focus of data center management. The use of geographically distributed data centers offers many benefits to end users, such as the ability to add additional data centers and increase service availability through redundancy and geographical distribution. Globalization, disaster recovery, and security are driving businesses to diversify their locations across multiple regions.

However, data centers are currently facing a significant impediment in the form of power consumption. The energy consumption of data centers will soon match or exceed that of many other energy-intensive industries, such as air travel. According to recent studies, energy consumption is a significant cost of IT operations, sometimes exceeding the cost of the IT hardware itself. Because of this cost pressure, as well as the realization that data centers can be much more energy efficient, many data center operators have prioritized energy management.

3.1.1 Virtualization in Data Centers

Data center virtualization is essentially a process that involves the design and deployment of a data center on cloud computing using virtualization

technology. This procedure allows for the virtualization of physical servers in a data center, storage, networking, and other infrastructure equipment and devices.

Virtualization can be regarded as a viable solution for addressing the majority of operational issues associated with Data Center construction and maintenance. Virtualization technology has been the primary enabler for these many excellent features of IaaS clouds by allowing providers to manage their resources more flexibly and generally. Virtualization allows for the creation of virtual machines, which are self-contained instances of software or an operating system. Therefore, managing VMs distributed over physical resources becomes a major concern for designing an IaaS cloud computing and faces many challenges. Likely traditional physical resources, virtual machines need configurations such as network setup and planning machine's software environment. Nonetheless, this configuration must be handled on-the-fly in a virtual infrastructure with as little time as possible between the time the virtual machines are requested and the time the virtual machines are provided to the customer.

Virtualization revolutionizes the data center technology through a set of tools and techniques that facilitate to provide and manage dynamically data center infrastructure. It becomes a crucial and enables cloud computing technology. Virtualization is characterized as an abstraction of 4 computing resources: network or I/O, storage, memory, and processing power. Hardware virtualization allows several software stacks and operating systems on a single physical platform. The virtualization layer will partition the underlying physical server's resource into multiple VMs with different workloads as shown in Figure 3.3. The fascinating fact about this layer of virtualization is that it schedules and allocates the physical resources, and makes every VM believe it owns the entire physical resources: RAM, Processor, and Storage Disks, etc.

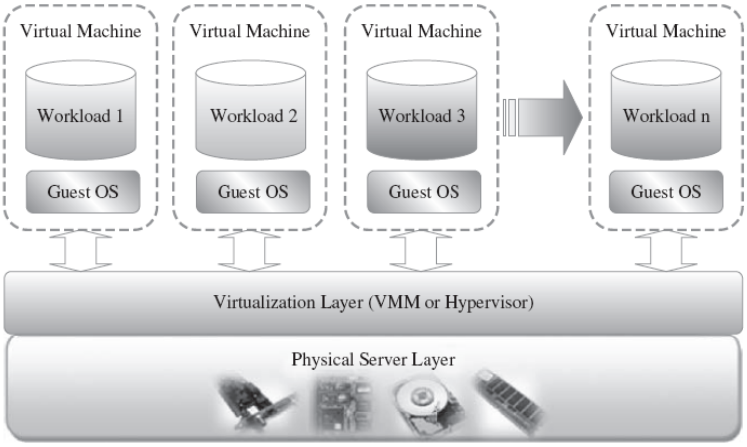


Figure 3.3 A Layered Virtualization Technology Architecture

Virtual machine technology makes resource management in cloud computing environments very flexible and efficient, it maximizes resource utilization by creating several VMs on a single physical machine. Virtualization allows high, agile, and reliable deployment mechanisms and services management, on-demand cloning provision and live migration services that enhance the reliability. For any cloud IaaS vendor, having an effective management's suite to manage VMs infrastructure is crucial.

3.1.2 Resource Management in Data Centers

A cloud computing platform is a hosting platform that rents data center resources and provides the end users with programmatically consumable resource management and monitoring services. Although cloud computing offers numerous technical, operational, and economic benefits, there are still a number of challenges that must be addressed. Resource management is critical for efficient operation in large data centers with many servers.

Resource management is arranging and allocating the resources for computing operations and applications. Because data centers contain thousands of computers, networks, and storage devices, the complexity of resource management and providing service to applications is much greater than that of resource management in a personal computer.

To be effective, resource management should consider several factors, such as dealing with workload dynamism, meeting SLOs for user requirements, and allocating user requests to available resources in an energy-efficient and cost-

effective manner. As the resource demand pattern has the dynamic nature, the resource prediction mechanism is essential to accomplish this approach. The operational costs in these systems are heavily influenced by the resource management algorithms used to assign VMs to PMs.

Most Internet service providers locate their cloud data centers in areas with volatile electricity prices available on multi-regional electricity markets, where electricity prices can vary in time and location. If data center electricity prices are forecast in advance, the cloud provider can reduce electricity costs by steering the job requests to favor the data center with the lowest electricity price, taking advantage of the diversity of location-based prices. To reduce data center electricity costs, it is necessary to predict the electricity price for data centers in multi-regional electricity markets.

Predicting resource demand is critical for efficient resource provisioning of dynamic workload. The cloud provider's resource requirements must be predicted ahead of time so that the resource management system can adjust resource provisioning to meet the needs of customers.

Resource management finds a good placement of VMs onto available physical machines in the data centers, servicing them to satisfy SLO, saving the energy consumption and operational costs of data centers. It is a type of resource allocation that is used to achieve the efficient usage of computer server resources. Efficient resource allocation algorithms and policies become even more important.

3.2 Apache Spark Processing Engine

Apache Spark [88] is an open and free distributed data processing platform that employs distributed memory abstraction to process large amounts of data efficiently. It is a popular open-source cloud platform that uses resilient distributed datasets (RDDs) [89] to enable fast processing of large volumes of data using distributed memory. It is well-suited for iterative applications such as iterative ML and graph algorithms due to its in-memory data operations. Spark specialises at iterative computation because it can reuse intermediate results and keep data in memory across multiple parallel operations. The execution time of a specific job on the Apache Spark platform, on the other hand, can vary significantly based on input

data type and size, as well as the algorithm design, computing capability (e.g., CPU speed, number of nodes, and memory size) and implementation.

As shown in Figure 3.4, Spark has evolved into a large data processing platform based on memory computing, with an architecture that includes resource management, distributed storage, data processing, and application. Berkeley refers to the entire Spark ecosystem as the Berkeley data analysis stack (BDAS). Spark can communicate with a variety of distributed storage systems, including Hadoop Distributed File System (HDFS), MapR File System (MapRFS), Cassandra, and others. Tachyon, Spark's distributed memory file system, is used to cache data in memory more than 100 times faster than HDFS. Spark supports resource management via standalone (native Spark cluster), Hadoop YARN, or Apache Mesos. The computing engine is the Spark computing framework-based RDDs that serve as the project's foundation. Many distributed subprojects, such as Spark Streaming, GraphX, Spark Structured Query Language (SQL), MLlib, and others, sit on the top of Spark.

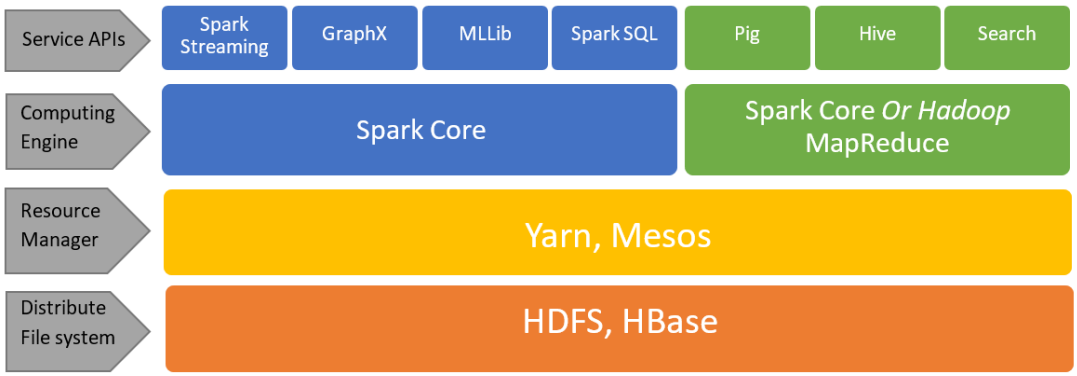


Figure 3.4 The Apache Spark Ecosystem

Spark simplifies the implementation of a wide range of applications, including relational data processing (such as SQL queries) and Machine Learning (ML) algorithms. ML techniques become more feasible and accurate as data volumes increase. Before applying the same solutions to unknown and new data, software can be trained to act and identify to triggers within well-known data sets. Because of Spark's ability to store data in memory and run repeated queries quickly, it's ideal for training ML algorithms. Running broadly similar queries at scale significantly reduces the time required to iterate through a set of possible solutions in order to determine the most effective algorithms.

Spark is a general purpose data processing engine with an API that application developers and data scientists can use to query, evaluate, and restructure data at scale. Developers use it to help in other data processing tasks by leveraging Spark's extensive set of developer APIs and libraries, as well as its extensive support for languages such as Java, Scala, R, and Python. Spark is frequently used in conjunction with HDFS, Hadoop's data storage module, but it can also work well with Cassandra, HBase, MongoDB, Amazon's S3, and MapR-DB.

There are numerous reasons to select Spark, but three stand out:

• *Simplicity*: Spark's capabilities are accessible via a set of rich APIs, all of which are specifically designed for interacting with data at scale easily and quickly. These APIs are well-structured and well-documented, making it simple for application developers and data scientists to put Spark to work quickly.

• *Speed*: Spark is designed to be fast, and it can run in memory as well as on disk. This victory resulted from the processing of a static data set. Spark's performance can be improved by supporting interactive queries on data stored in memory.

• *Support*: Spark works with a wide range of programming languages, such as Java, Scala, R and Python. Despite its close relationship with HDFS, Hadoop's underlying storage system, Spark includes native integration support for a wide variety of popular storage solutions in the Hadoop ecosystem and beyond. Furthermore, Apache Spark community is vibrant, large, and international.

3.3 Prediction Perceptions in Data Mining

The task of predicting continuous (or ordered) values offered a set of input values is known as prediction [36, 83]. This is a type of data analysis used to retrieve models for predicting future data trends. Identifying distribution trends based on available data is also part of prediction. It can predict missing or unavailable numerical data values, and this analysis can help gain a better understanding of the data as a whole. Prediction represents continuous valued functions as a function or mapping, y = f (X), where "X" is the input and "y" is a continuous or ordered value.

Regression, a statistical method is the most widely used method for numeric prediction. Regression is the method of finding a pattern or identifying a function to separate the data into real or numeric continuous values. The output variable takes on continuous values. It can also identify the distribution movement depending on the historical data, because a predictive regression model predicts a quantity, thus, the model 's ability must be reported as an error.

Regression analysis can be used to model the relationship between one or more dependent or response variables and independent or predictor variables. This analysis is appropriate when all of the predictor variables are continuous values.

In the context of data mining, the predictor variables are the attributes of interest that describe the tuple. In general, the predictor variables' values are known, and the associated value of the response variable is one to predict based on the predictor variables.

Many problems can be solved with linear regression, and even more can be solved by applying transformations to the variables to convert a nonlinear problem to a linear problem. Linear regression is used to model continuous-valued functions. Because of its simplicity, it is widely used. Theoretical foundations for using linear regression to model categorical response variables are provided by generalized linear models. Generalized linear models include logistic regression and Poisson regression.

To model discrete multidimensional probability distributions, log-linear models are used. They are capable of calculating the probability value of data cube cells. The log-linear model can be used for data compression and smoothing in addition to prediction. There are two types of trees for prediction: regression trees and model trees. Regression trees were included as a component in the Classification and Regression Trees (CART) learning system. Each regression tree leaf stores a continuous-valued prediction, which is the average of the predicted attribute's value across all training tuples that reach the leaf. A regression model, on the other hand, is contained in each leaf of a model tree—a multivariate linear equation for the predicted attribute. When a simple linear model does not adequately represent the data, regression and model trees are more accurate than linear regression.

3.3.1 Machine Learning Approach

Data mining and machine learning, which are closely related fields, are concerned with extracting knowledge from data [74]. Because they perform well when long data series are available, ML techniques are said to be data dependent. Typically, this entails developing models or identifying patterns in historical examples of system behavior with as little expert intervention as possible. This prediction process necessitates the selection of appropriate predictor algorithms that are computationally light but capable of producing good results after being trained on data from a variety of workloads. A good training data set containing labeled instances from representative executions is also required, as is another validation or test set. If the predictors' guesses on the test set are close to the correct values after training, expect them to be correct on also real future workloads.

3.3.1.1 Machine Learning Algorithms

ML algorithms build the models by extracting knowledge from the data. Then, these models can make predictions on data. ML model predictions allow making highly accurate estimates as to the likely outcomes. Machine learning models require the historical data of the workload for training data. Many prediction models use the ML algorithms such as M5P, Decision Table, Random Forest, etc.

M5P: M5P is a restructured version of the Quinlan M5 algorithm, which is used to generate trees in regression models. M5P combines the traditional decision tree with the option of linear regression functions at the nodes. All of the listed attributes are converted into binary variables, resulting in binary M5P splits. In the case of missing values, M5P employs a technique known as "surrogate splitting," which finds other attributes to split on in place of the original one and uses them instead. During training, M5P uses the class value as a surrogate attribute in the belief that it is the most likely to be correlated with the one used for splitting. After the splitting cycle is completed, all missing values are replaced by the average values of the corresponding attributes of the training examples that enter the leaves. During testing, an unknown attribute value is substituted by the average value of that attribute for all training instances that enter the node, resulting in the selection

of the most common sub-node. M5P produces compact, comparatively understandable models [56].

Linear Regression: Linear regression is an effective, easy way of numeric predicting. It is a staple technique in statistics which works with numerical attributes. It is a linear modeling method that models the relationship between a scalar variable "y" and one or more variables denoted "X," with those models being linearly dependent on unknown parameters predicted from data. Linear regression is typically used to describe a model that expresses the median or any other quantize of the conditional distribution of "Y" given "X" as a linear function of "X." Linear regression has been widely used in practical applications because models that are linearly dependent on their unknown parameters are much easier to build than non-linear ones. A linear regression, in particular, is a natural method to consider when all of the attributes, as well as the result or class, are numeric.

Decision Table: A decision table (DT) is a brief visual representation that specifies the actions that must be taken based on the given conditions. The information shown in decision tables can also be represented as decision trees or in a programming language by using if-then-else and switch-case statements. A decision table is a good way to settle for various combination inputs and their corresponding outputs. It is also known as a cause-effect table, which is a related logical diagramming technique that is primarily used to obtain the decision table [21].

Decision Tree: A decision tree is a supervised model of machine learning which is used to predict a goal by learning from features the rules of decision. The decision tree is a greedy algorithm that executes the feature space through a recursive binary partitioning. The tree predicts the same label for each Bottommost partition (leaf). Each partition is selected greedily by choosing the best split from a set of possible splits, to maximize the information gain at a node tree [22]

The hyper-parameters controlling how the tree’s decisions are chosen will be quite different as well: maximum depth, minimum instances per node, and minimum information gain. Those parameters decide when the tree will stop to build (add new nodes). To avoid overfitting, it must be careful to validate on hold-out test data when tuning those parameters.

Random Forest: The Random Forests predictor [14] is an ensemble predictor that uses random feature sub-spacing [38] and bagging [15] to create accurate decision tree classification or regression ensembles. Both random sub-spacing and bagging are effective ensemble methods in their own right because they both implicitly promote diversity among classifiers by limiting their scope and effectively forcing them to learn a rather highly specialized prediction rule. Each individual ensemble member 'votes' for the prediction that most closely resembles its limited scope during prediction; these votes are then combined using an aggregation method to compute the final ensemble prediction.

3.3.1.2 Hyper-parameters Optimization

Machine Learning algorithms include hyper-parameters that define the model architecture and must be set before they can be run. The process of determining the best model architecture is known as hyper-parameter tuning. In theory, hyper-parameter tuning is an optimization task, similar to model training.

Optimization of hyperparameters can have a significant impact on the prediction performance of machine learning algorithms. The hyper-parameter optimization approach requires determining the parameter setting for the learning algorithm which will produce a model with a low error rate.

The hyper-parameter settings may have a significant impact on the trained model's prediction accuracy. Optimal hyper-parameter settings frequently differ between datasets. As a result, they should be customized to each dataset. Because the training process does not set the hyper-parameters, a meta process that tunes the hyper-parameters is required.

However, because the quality of hyper-parameters is dependent on the outcome of a black box, it cannot be written down in a closed-form formula (the model training process). This demonstrates how much more difficult hyper-parameter tuning is. Grid search and random search were the only methods available until recently.

Grid Search: As the name implies, grid search selects a grid of hyper-parameter values, evaluates each one, and returns the winner. The most fundamental hyper-parameter tuning technique is arguably grid search. Simply put, we create a

model for every possible combination of the hyper-parameter values provided, evaluate each model, and select the architecture that produces the best results. Some guesswork is required to specify the minimum and maximum values. As a result, people will run a small grid to see whether the optimum is at either endpoint, and then they will expand the grid in that direction.

Random Search: In contrast to grid search, random search provides a statistical distribution from which values can be randomly sampled rather than a discrete set of values to explore for each hyper-parameter. Grid search is a subset of random search. Random search evaluates a random sample of grid points rather than the entire grid. As a result, random search is much less expensive than grid search. Previously, a random search was dismissed. Because it does not search over all of the grid points, it cannot possibly outperform the optimum of grid search.

3.3.2 Accuracy of Prediction Methods

The accuracy of a predictor refers to how well a predictor can guess the value of the predicted attribute for new or previously unseen data. One or more independent test sets from the training set can be used to estimate accuracy. Using the measurement metrics in subsection 3.3.2.2, the prediction methods can be evaluated and compared under various estimation techniques.

3.3.2.1 Predictors Accuracy Estimation Techniques

The techniques for accuracy estimation [36, 83] are the random subsampling, holdout, k-fold cross-validation and bootstrap methods. These are the most common methods for evaluating accuracy based on randomly sampled partitions of given data. These estimation techniques lengthen the overall computation time and they are useful for model selection.

Holdout Method and Random Subsampling: In this method, the given data is randomly partitioned into two independent sets, a training set and a test set. Typically, two-thirds of the data is assigned to the training set, with the remaining one-third assigned to the test set. The training set is used to build the model, and the

test set is used to estimate its accuracy. The estimate is pessimistic because only a portion of the initial data is used to derive the model. Random subsampling is a holdout method variation that involves repeating the holdout method k times. The overall estimation accuracy is calculated by taking the average of the accuracies obtained from each iteration.

Cross-validation: In k-fold cross-validation, the initial data is randomly partitioned into *k* mutually exclusive subsets or "folds," D_1, D_2,...,D_k, each of approximately equal size. The training and testing phases are repeated *k* times. Partition D_i is designated as the test set in iteration *I* and the remaining partitions are used to train the model collectively. Each sample is only used once for training and once for testing. The error estimation for prediction is calculated by dividing the total loss from the *k* iterations by the total number of initial tuples. Leave-one-out cross-validation is a subset of k-fold cross-validation, where *k* is the number of initial tuples. Stratified 10-fold cross-validation is generally recommended for estimating accuracy due to its low bias and variance (even if computation power allows for more folds).

Bootstrap: The bootstrap method uses replacement to sample the given training tuples uniformly. There are several bootstrap methods, the most common of which is the .632 bootstrap. To create a training set, the bootstrap method samples the dataset with replacement. A given *d*-tuple data set is sampled with replacement d times, yielding a bootstrap sample or training set of *d* samples. Because the training set contains only 63 percent of the instances despite its size of *n*, the model obtained by training a learning system on the training set and calculating its error over the test set will be a pessimistic estimate of the true error rate. The test-set error rate is combined with the re-substitution error on the training set instances. The re-substitution process provides an overly optimistic estimate of true error and should not be used as error in and of itself. The bootstrap procedure, on the other hand, manages to combine it with test error rate to generate a final estimate, which is mathematically described in Equation 3.1.

$$e = 0.632 \times e_{\text{test instances}} + 0.368 \times e_{\text{training instances}} \quad (3.1)$$

Where *e* is the natural logarithm base with the value 2.718. The entire bootstrap procedure is repeated several times with a different replacement sample for training set each time, and the results are averaged.

3.3.2.2 Predictors Measurement Metrics

The reliable estimate of predictor performance is measured in terms of error. Let D^T be a test set of the form $(X_1, y_1), (X_2,y_2), \ldots, (X_d, y_d)$, where the "$X_i$" are the *n*-dimensional test tuples with the associated known values, "y_i," for a response variable, "*y*," and "*d"* is the number of tuples in "$D^{T"}$ .The accuracy of a predictor is computed by an error based on the difference between actual known value of "y" and the predicted value for each of the test tuples, "X". Loss functions measure the error between actual value, "y_i " and the predicted value, "y_i'". The most common loss functions can be executed in Equation 3.2 and 3.3.

Absolute error : $|y_i - y_i'|$ (3.2)

Squared error : $(y_i - y_i')$ (3.3)

Based on the foregoing, the test error rate, also known as generalization error, is the average loss across the test dataset. Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are the most commonly used evaluation metrics for determining prediction accuracy (MAE).

$$\text{Mean Squared Error (MSE)} = \frac{\sum_{i=1}^{d}(y_i - y_i')}{d} \quad (3.4)$$

$$Root\ Mean\ Squared\ Error\ (RMSE) = \sqrt{\frac{\sum_{i=1}^{d}(y_i - y_i')}{d}} \quad (3.5)$$

$$MeanAbsoluteError\ (MAE) = \frac{\sum_{i=1}^{d}|y_i - y_i'|}{d} \quad (3.6)$$

The MAE does not exaggerate the presence of outliers, whereas the MSE does. The square root of MSE is used to calculate RMSE. This is beneficial because it allows the measured error to be of the same magnitude as the estimated quantity. MAE, MSE, and RMSE can have values ranging from 0 to ∞. The lower the error values, the better the model is.

Furthermore, the performance of the predictors can be evaluated as

- *Speed*-the computational costs to generate and use the predictor,
- *Robustness* - the ability of the predictor to predict correctly given data with missing values or noisy data,
- *Scalability* - the ability to develop the predictor efficiently given large amount of data, and
- *Interpretability*- the level of insight and understanding developed by predictor.

Although various metrics exist to measure the predictors' performance, this research emphasizes on the metric regarding to the model accuracy.

3.4 Heuristics for Energy-Efficient Management

Energy-efficient cloud infrastructures are designed to make it easier for users to access and deploy various service-oriented applications [40]. Cloud environments provide high-performance servers to meet the growing demand for computations and large data volumes [41]. Cloud services are provided by server companies or data centers. Data centers have sprouted up to serve as large server farms, utilizing economies of scale to provide computation and storage resources to one or more businesses. These datacenters typically house hundreds of thousands of servers, each of which consumes enormous amounts of power. The Cloud data center's total computing power is the sum of the computing power of each individual PM.

Thus energy consumption has become an important concern because of its impact on capital costs, operating expenses, and environmental sustainability. Furthermore, cloud energy consumption is proportional to resource utilization, and data centers are one of the world's largest electricity consumers. Data centers require efficient technology due to their high energy consumption.

The power and time required to turn on the servers are multiplied to calculate energy consumption. The goal of energy efficiency is to reduce the amount of energy consumed by servers in data centers. Cloud data centers, on the other hand, can reduce total energy consumed through virtualization because workloads

can share the same server and the unused servers can be turned off using power management techniques.

By considering the parameter of energy consumption (E), it can be minimized by reducing the power consumption (P) as well as also the time period (T) needed to turn on the servers. The main component of a data center's operating cost is the electricity cost consumed during the period of operation, which is reflected in the energy consumption. There are some power management techniques that can be deployed for monitoring and controlling energy consumption of servers.

3.4.1 Power Management Techniques

Because servers are the main energy consumers in data centers, power management techniques are used to reduce server power consumption. Dynamic Voltage and Frequency Scaling (DVFS), Power-Aware (PA), and Non-Power-Aware (NPA) techniques are applied to reduce power consumption while maintaining the quality of services.

NPA - The energy consumption of the servers is calculated without any power-saving mechanism. Because there is no mechanism for reducing energy consumption, total energy consumption is determined by the power consumption of the activated servers and is unaffected by CPU utilization. For both too low and too high extremes of CPU utilization, servers consume the same amount of maximum power. It does not support shutting down the unused machines.

PA - As NPA, energy usage is calculated independently of CPU utilization. Power consumption in the data center can also be reduced by turning off unnecessary machines. It supports shutting down the unutilized servers in the data center. For both too low and too high extremes of CPU utilization, the activated servers consume the same amount of the servers' maximum power.

DVFS - It is the dynamic power management technique [49] that decreases a processor's dynamic power consumption expressed in (3.7) by dynamically changing the voltage and frequency of the processor based on CPU utilization during execution. It scales the system's power by varying both CPU frequency and voltage. DVFS [53] technique can dynamically adjust the frequency or operating

voltage of a processor of a server according to the service requirement without restarting the power supply.

$$P_{dynamic} = a \cdot c \cdot v^2 \cdot f \quad (3.7)$$

Where, $P_{dynamic}$: dynamic power consumption

a : switch activity

c : capacitance

v : voltage

f : frequency

Fan et al. [27] discovered a strong relationship between total power consumption of server and CPU utilization, demonstrating that power consumption of a server rises linearly with the increase in CPU utilization. DVFS reduces dynamic power consumption by dynamically changing the processor's frequency and voltage during execution based on CPU utilization. A significant amount of energy is saved when CPU voltage is reduced based on CPU utilization. As a result, by lowering dynamic power consumption, cloud service providers can increase their revenue. The power consumption of the server can be reduced when it is in an idle state or low workload through DVFS technique. This method can decrease the power consumption of servers and enhance resource utilization. Existing power management methods only emphasize on the control of switching on or off the servers. In this research, DVFS is applied to manage the energy consumption of servers in data centers.

3.4.2 Energy-efficient Power Models

Based on CPU server utilization and power consumption in idle and maximum states, power models can estimate the power consumption of CPU servers. CloudSim comes with a generic "Power Model" implementation that can be extended to support a wide range of power models [57]. The following power models are available in recent CloudSim releases:

Linear model: $P(u) = P_{idle} + (P_{max} - P_{idle}) * u \quad (3.8)$

Square model: $P(u) = P_{idle} + (P_{max} - P_{idle}) * u^2 \quad (3.9)$

Cubic model: $P(u) = P_{idle} + (P_{max} - P_{idle}) * u^3$ (3.10)

Square root model: $P(u) = P_{idle} + (P_{max} - P_{idle}) * \sqrt{u}$ (3.11)

where $\boldsymbol{u}$: utilization of CPU

$\boldsymbol{P_{max}}$: the maximum power of CPU

$\boldsymbol{P_{idle}}$: the idle power of CPU

3.5 Resource Allocation Policies

Resource allocation or scheduling is a critical task in cloud computing. The process of creating VM instances that correspond to incoming requests on the hosts is known as resource allocation. It entails identifying and allocating resources to each incoming user request in order to meet the user's requirements while also meeting the cloud provider's specific goals. These objectives could include reducing energy consumption, lowering costs, and so on. The resource allocator or scheduler determines resource allocation solutions based on resource information such as request information, resource usage and monitoring, and the Cloud provider goal.

This work focuses on allocation policies to optimize processing time and energy consumption of servers. The optimal allocation policy decreases the time and availability of space in a productive manner without compensating the quality of the system [45]. There are many resource allocation policies as the following:

First Come First Serve: It is one of the simplest jobs ordering policy. The request that arrives at the data center first is assigned to the VM first. To make an allocation decision, the data required by the allocator is the arrival time of the request.

Shortest Job First: The request with the shortest runtime (length) in the ready queue is allocated to the VM first. If two requests have the same length, the FCFS policy is used to allocate the next request, i.e. the one that arrives first is allocated to the VM.

Time-Shared and Space-Shared: Figure 3.5 illustrates the difference between time-shared and space-shared policies for VMs and tasks, as well as their impact on application service performance, using a simple VM allocation scenario. In this illustration, a host with two CPU cores receives a request to place two VMs, each of which requires two cores and intends to host four task units. Tasks t1, t2, t3, and t4 will be hosted in VM1, while tasks t5, t6, t7, and t8 will be hosted in VM2.

Figure 3.5 (a) describes a space-sharing policy for both virtual machines and task units: each VM requires two cores, and only one VM can run at any given time instance. As a result, VM2 can only be assigned the core after VM1 completes its execution. The same is true for tasks hosted within the VM: because each task unit requires only one core, two of them run concurrently, while the other two are queued until the former tasks are completed.

In Figure 3.5 (b), a space-shared policy is used when allocating VMs, but a time-shared policy is used when allocating individual task units within a VM. As a result, all tasks assigned to it dynamically context switch until they are completed within a VM lifetime. This allocation policy allows task units to be scheduled earlier, but it has a significant effect on the completion time of task units that are ahead of the queue.

In Figure 3.5 (c), for task units, a space-shared policy is used, and for VM allocation, a time-shared policy is used. Each VM receives a time slice of each processing core, and then slices are distributed to task units using a space-shared policy. Because the core is shared, the amount of processing power available to the VM is less than the previous scenarios. Due to the fact that task unit assignment is space-shared, only one task can be assigned to each core, while others are queued for future allocation.

Finally, in Figure 3.5 (d), a time-shared allocation is used for both VMs and task units. As a result, the processing power is shared concurrently by the VMs, and the shares of each VM are divided concurrently among the task units assigned to each VM. Queues do not exist for task units or virtual machines.

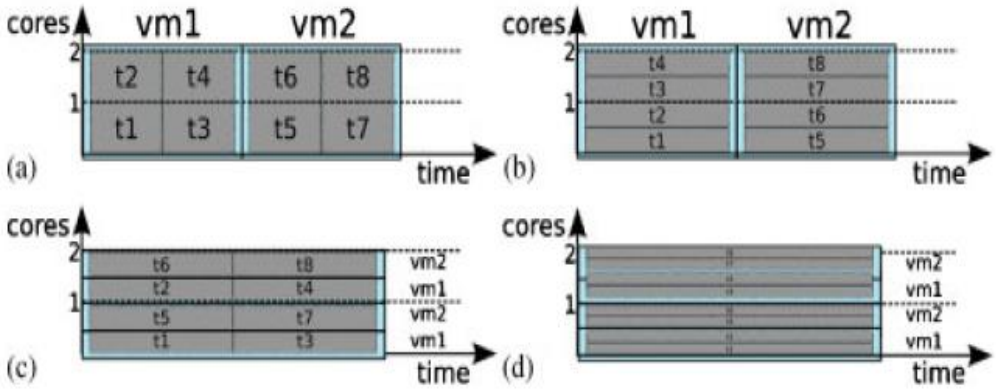


Figure 3.5 Different Scheduling Policies Effects on Task Execution (a) Space-shared for both tasks and VMs, (b) Space-shared for VMs and time-shared for tasks, (c) Time-shared for VMs, space-shared for tasks, and (d) Time-shared for both tasks and VMs

3.6 Chapter Summary

Virtualization technology is the backend technology of the cloud data centers. Cloud providers face many challenges for resource demands because cloud grows in usage and popularity. Spark processing engine is introduced for prediction. Several machine learning techniques for prediction have been investigated by various methods. Energy-efficient techniques and resource allocation are also presented.

CHAPTER 4

PREDICTING ELECTRICITY PRICE FOR GEO-DISTRIBUTED DATA CENTERS IN MULTI-REGIONAL ELECTRICITY MARKETS

Cloud computing offers services available to consumers in a pay-as-you-go model through geo-distributed data centers (GDCs) scattered across diverse geographies. GDCs power consumption is significantly increasing as cloud computing services are dramatically growing. Consuming a tremendous amount of energy which leads to high operational cost has become an emergent issue.

Energy costs can be minimized by routing the incoming job requests to prefer the data center with cheaper electricity prices by exploiting spatial and temporary prices variety, especially in multi-regional electricity markets. When electricity prices of GDCs are predicted in advance, the cloud provider can lower the cost of energy. Since GDCs are with various electricity prices, an intuitive way is to distribute more workloads to data centers with lower price knowing the future electricity prices of GDCs. The electricity price prediction is required to reduce GDCs electricity bills. This chapter discusses electricity price prediction for GDCs in multi-regional markets.

This chapter begins the introduction for multi-regional electricity markets in the United States. This chapter presents predicting electricity prices of GDCs in US multi-regional markets considering about the regions' time zones as [1]. The developments of the prediction models are based on Machine Learning Techniques for electricity price prediction. Not only the model development but also the model validation is exhibited in this chapter. Experiments are performed with three ML algorithms on nine annual electricity price datasets and the most reliable predicting model is chosen to predict the electricity prices.

4.1 Multi-Regional Electricity Markets

In power markets, generators and consumers are typically linked to an electricity grid, a complex network of transmission and distribution lines. For example, the United States is divided into several such grids. Furthermore, due to the grid's complex transmission conditions and diverse generator profiles, electricity prices typically fluctuate as a function of regional load variation. In other words, electricity prices in local power markets may fluctuate in response to variable power demands; for example, when load reaches a certain level, a step change in Locational Marginal Price (LMP) may appear.

In the United States, the regional transmission organizations operate the electric power grid on a regional basis. A regional transmission organization manages the entire power market while also ensuring transmission reliability. Electricity prices vary by location because different regions have different power generation profiles [7]. Electricity prices may vary hourly throughout the day in some regions with deregulated electricity markets, whereas electricity prices in the majority of regions remain constant throughout the day [60]. Figure 4.1 [28] depicts the electricity markets in the United States.

Figure 4.1 US Multi-regional Electricity Markets

To determine electricity prices, LMP methodology has been used as a dominant approach in energy market planning and operation. Independent System Operators (ISO), including PJM and ISO-New England, have implemented or are considering implementing the LMP methodology to determine how do electricity prices change in local power markets in response to power demands.

4.2 Electricity Prices Prediction for Multi-Regional Electricity Markets

Because of the widespread success and expansion of cloud computing, data center operators have geographically distributed their data center locations. To improve reliability and performance, almost all Internet-scale services are built on geographically distributed data centers (GDCs). Servers in data centers typically consume enormous amounts of electricity, resulting in high operational costs, and this has been identified as a major challenge in cloud computing. Electricity bills account for 30-50 percent of a data center's operational costs. According to a Natural Resources Defense Council report [62], US data centers alone consumed 91 billion kilowatt-hours of electricity, which is equivalent to the power consumption of all New York households for two years. The ability to reduce operating expenses by utilizing time-of-use (TOU) electricity pricing is a strong motivator for data center operators [54], and reducing electricity costs is now a focus of data center management.

Energy costs can be reduced by directing requests to data centers with lower electricity prices by incorporating spatial and temporal price diversity, particularly in multi-regional markets. Most Internet service providers locate their data centers in areas with volatile electricity prices available on multi-regional markets, where electricity prices can vary in time and location.There are several advantages to distributing workload among geo-distributed data centers. Workloads can be shifted to locations in different time zones in order to concentrate workload in the regions with the lowest electricity prices at the time.

If data center electricity prices are forecast in advance, the cloud provider can reduce electricity costs by directing job requests to the data center with the lowest electricity price, taking advantage of the diversity of location-based electricity prices. To reduce GDC electricity costs, it is necessary to predict the electricity price for GDCs in multi-regional markets. TOU pricing knowledge is typically used to either minimize electricity costs across all GDCs or to maximize profits when a revenue model for computing is included. For GDCs to reduce their electricity bills, an accurate prediction of electricity prices is required. This section forecasts electricity prices for GDCs operating in multi-regional markets.

4.2.1 Electricity Price Prediction System Architecture

Figure 4.2 depicts the system architecture for electricity price prediction, which consists of three major steps: electricity price data pre-processing, model selection, and electricity price prediction. Model selection and model assessment are two distinct goals when estimating prediction accuracy.

Model Selection: Comparing MAE results of different machine learning algorithms to select the most accurate prediction model.

Model Assessment: Estimating the prediction performance on new data using the selected model. Model assessment is handled in price prediction section.

The best approach for both model selection and assessment targets is to divide the dataset into three distinct parts: *Training dataset* used to build the models, *Testing dataset* needed to estimate the test error of the predictive models, and *Validating dataset* required to assess the selected model.

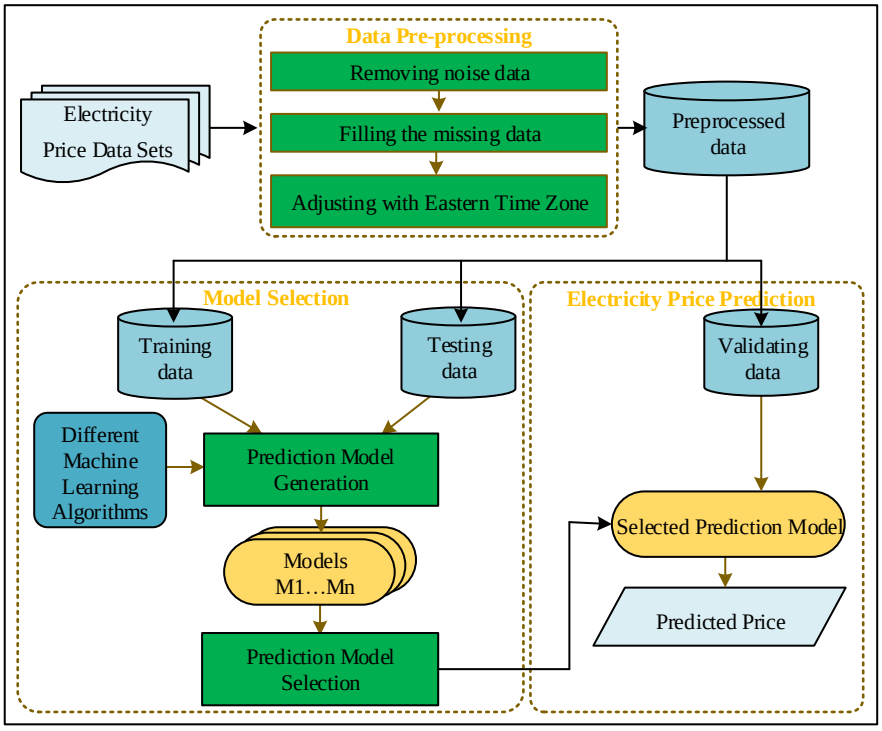


Figure 4.2 Proposed Electricity Price Prediction System Architecture

4.2.1.1 Electricity Price Data Pre-processing

Data pre-processing is effective in eliminating the irrelevant data, reducing the dimensionality, increasing the accuracy, and improving the result comprehension.

Ideally, all of the days for the entire period should be available, but some of this information was lost and some records were duplicated. Electricity price datasets are pre-processed by removing duplicated records in advance and replacing missing values with the most frequently occurring value for that feature, or with the most likely statistical value. In this step, we change the price data so that it refers to the Eastern Time Zone as [1].

This study takes into account three GDCs located in three different markets that have both temporal and geographical variation in electricity prices, for example, the time zones of New England, California, and Texas are GMT-7, GMT-8, and GMT-6, respectively. It keeps time in the Eastern Time Zone and thus adjusts the time difference between New England, California, and Texas by 0, 3, and 1 hour, respectively.

4.2.1.2 Electricity Price Predictive Model Selection

Following the data pre-processing phase, a set of models is generated using various ML algorithms, and the most accurate model is chosen. The machine learning evaluation process is carried out in this section to select the best machine learning algorithm.

Various ML algorithms are used to generate different prediction models by using training and testing data for each dataset. Linear Regression, M5P, and Decision Table algorithms are used to generate models in the experiment. For each dataset, the same input attributes are applied to three machine learning algorithms, making it simple to compare the results and determine which model can predict the most accurately.

The best accurate predictive model is chosen by comparing the MAE results of the developed models using three machine learning algorithms on nine real-world electricity prices datasets.

4.2.1.3 Electricity Price Prediction

That is used for model evaluation in order to forecast future electricity prices after selecting the best predictive model for each market. The model obtained for each market is validated by feeding the newly validated data for all markets over the same time period, allowing GDCs electricity prices to be compared.

4.3 Experimental Environment

In this section, system specification for testing environment and description of real-life electricity price traces and experimental study are also presented.

4.3.1 System Specification

Experiments are performed on Dell computer with Intel (R) Core (TM) i3 CPU @ 2.00 GHz and 4 GB RAM. The WEKA 3.8 toolkit is used to create predictive models and evaluate prediction accuracy. Predictive models are implemented as Java program on Eclipse.

4.3.2 Experiment Electricity Price Traces

The majority of cloud providers construct their GDCs in various regions where dynamic electricity prices are available in competitive markets. To better capture the variations in electricity prices, three GDCs exploited in different regional electricity markets in the United States are assumed. GDCs are assumed to be connected to the power grid and capable of purchasing electricity from a local node.

As the electricity prices for each data center, we used the publicly available data from the markets website [25]: Electricity Market of Independent System Operator New England (ISONE), California (CAISO), and Electric Reliability Council of Texas (ERCOT) obtained from New England, California, and Texas respectively. GDC sites for DC1 in ISONE, DC2 in CAISO, and DC3 in ERCOT are connected to the hubs: Z.SEMASS, PGAE, and LZ_AEN respectively. Hourly electricity prices ($/MWh) are experimented with for each of three markets in the years 2014, 2015, 2016, 2017, and the first two months of 2018. These datasets

include three features: the name of the regional Hub, the date and time, and the Locational Marginal Price (LMP). All attributes of the electricity markets are considered to apply the prediction model.

4.3.3 Experimental Study

All experiments are carried out for each of the years: 2014, 2015 and 2016 annual dataset is divided into 90% used for training and 10% used for testing in each dataset of three markets: ISONE, CAISO and ERCOT to conduct machine learning evaluation process. Validating dataset of the year 2017 and two months: January and February of 2018for each of three markets are applied for evaluating the chosen model.

Hourly prices ($/MWh) data for each of three years: 2014, 2015, and 2016 annual datasets for each of three markets are tested to determine which annually trained model can predict well. As a result, there are nine datasets: three annual datasets for each of the three markets.

To compare the MAE results of the generated models for each market, three annual datasets with three machine learning algorithms: Linear Regression, M5P, and Decision Table are evaluated.

4.4 Experimental Results and Analysis

This section illustrated the experimental results of each purpose of predicting electricity prices over three markets.

4.4.1 Experimental Results of Prediction Models Generations

For ISONE electricity market, the comparative MAE results of predictive models for each of the years: 2014, 2015 and 2016 annual dataset developed by three machine learning algorithms are shown in Figures 4.3- 4.5. It can be clearly observed in the figure that the models developed by the M5P algorithm achieve less MAE result than Decision Table and Linear Regression for all three annual datasets. And the MAE result of M5P model for the year 2016 dataset is the lowest as shown in Figure 4.6.

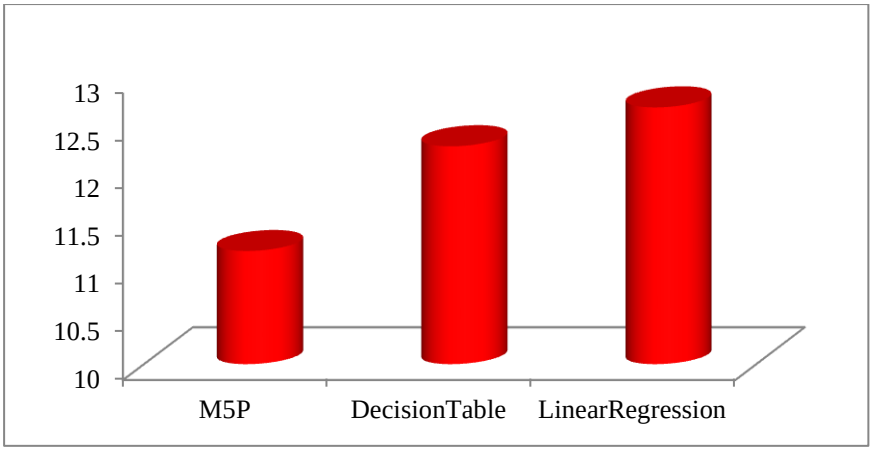


Figure 4.3 **MAE results Comparison of Different Machine Learning Algorithms for the Year 2014 Dataset (for ISONE market)**

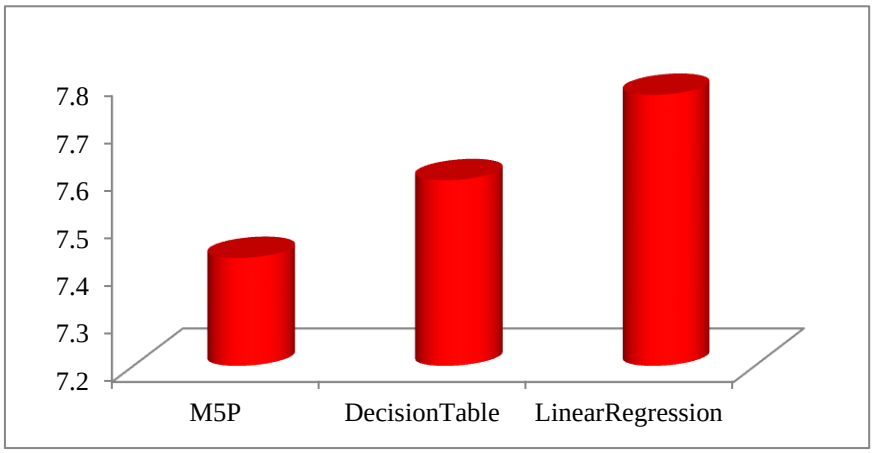


Figure 4.4 **MAE results Comparison of Different Machine Learning Algorithms for the Year 2015 Dataset (for ISONE market)**

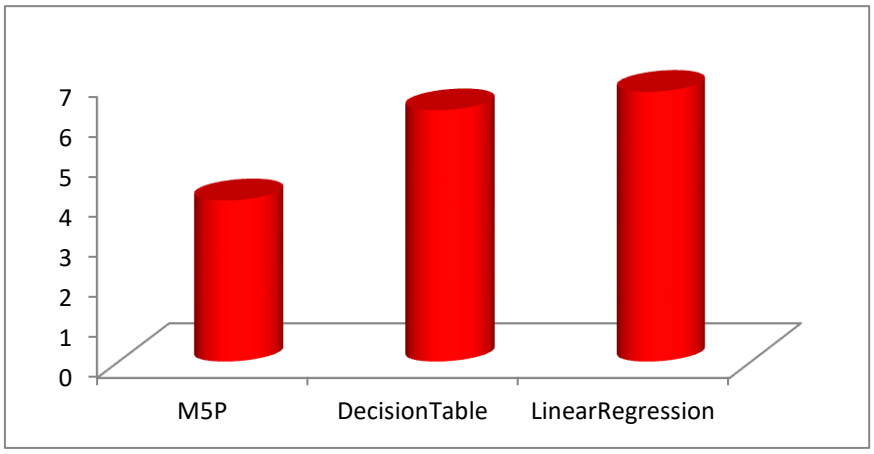


Figure 4.5 **MAE results Comparison of Different Machine Learning Algorithms for the Year 2016 Dataset (for ISONE market)**

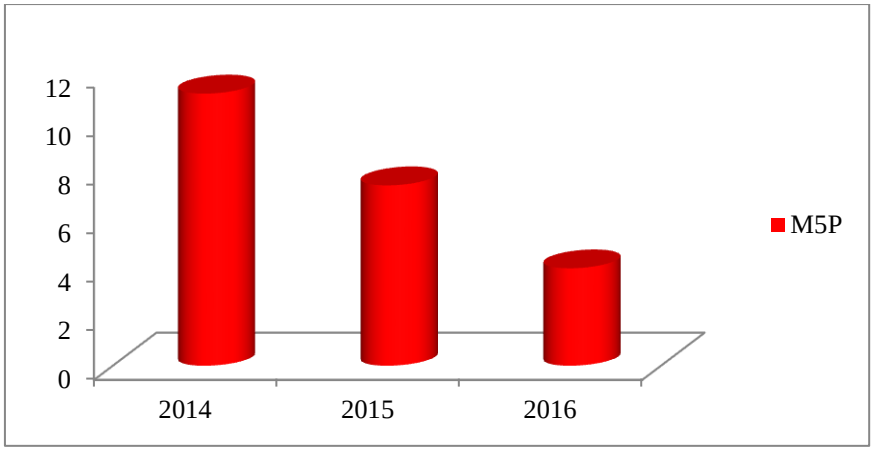
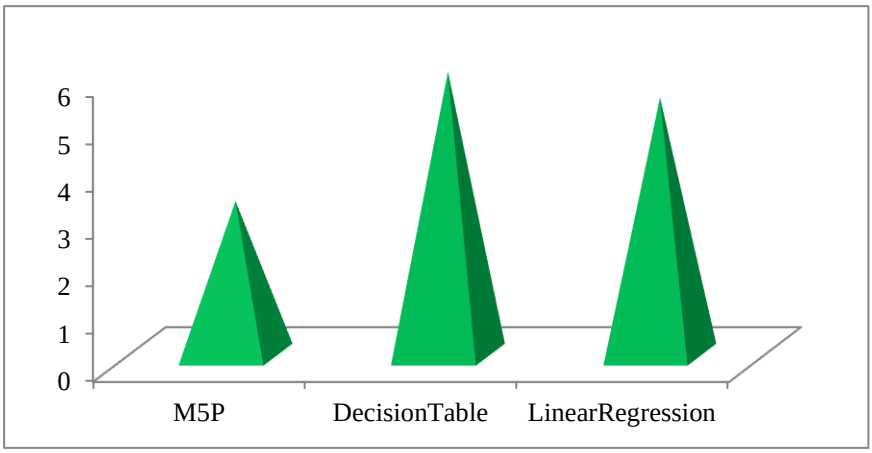


Figure 4.6 MAE results Comparison of M5P Machine Learning Algorithm for Three Annual Datasets (for ISONE market)

For CAISO electricity market, the comparative MAE results of predictive models for each of the years: 2014, 2015 and 2016 annual dataset developed by three machine learning algorithms are shown in Figures 4.7- 4.9. It can be clearly observed in the figure that the models developed by the M5P algorithm achieve less MAE result than Decision Table and Linear Regression for three annual datasets. And the MAE result of M5P model for the year 2016 dataset is the least as shown in Figure 4.10.



. Figure 4.7 MAE results Comparison of Different Machine Learning Algorithms for the Year 2014 Dataset (for CAISO market)

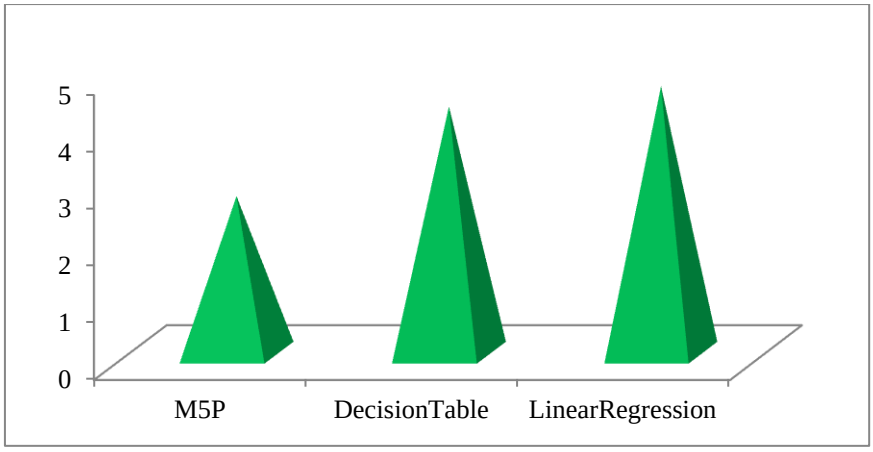


Figure 4.8 MAE results Comparison of Different Machine Learning Algorithms for the Year 2015 Dataset (for CAISO market)

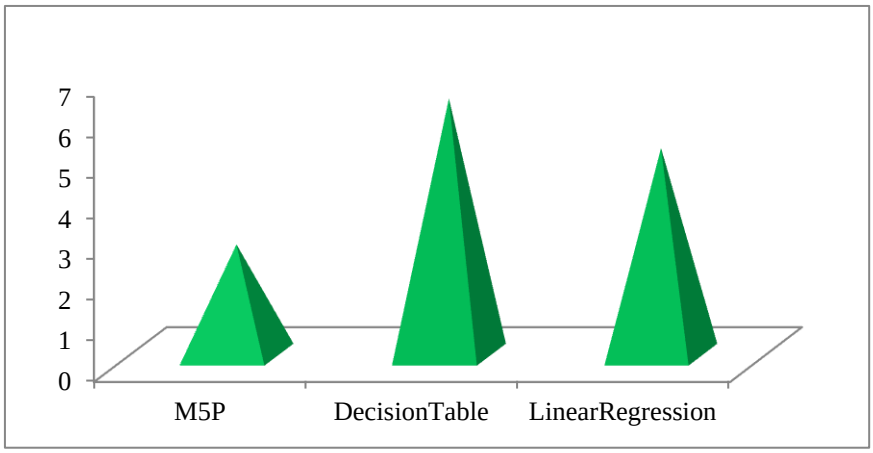


Figure 4.9 MAE results Comparison of Different Machine Learning Algorithms for the Year 2016 Dataset (for CAISO market)

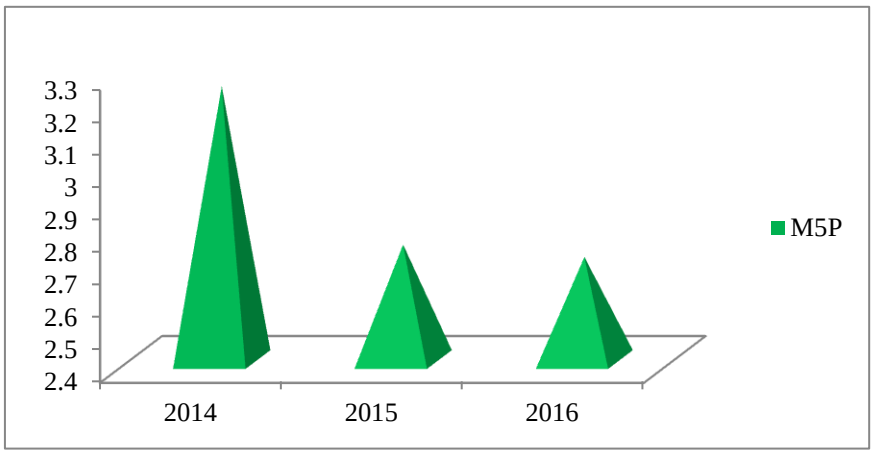


Figure 4.10 MAE results Comparison of M5P Machine Learning Algorithm for Three Annual Datasets (for CAISO market)

For ERCOT electricity market, the comparative MAE results of predictive models for each of the years: 2014, 2015 and 2016 annual dataset developed by

three machine learning algorithms are shown in Figures 4.11- 4.13. It can be clearly observed in the figure that the models developed by the M5P algorithm achieve less MAE result than Decision Table and Linear Regression for three annual datasets. And M5P model for the year 2016 dataset also achieves the least MAE as shown in Figure 4.14.

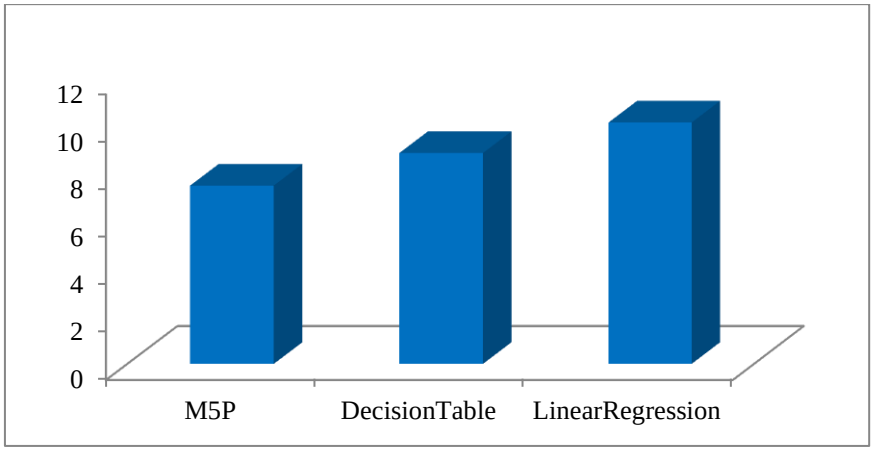


Figure 4.11 MAE results Comparison of Different Machine Learning Algorithms for the Year 2014 Dataset (for ERCOT market)

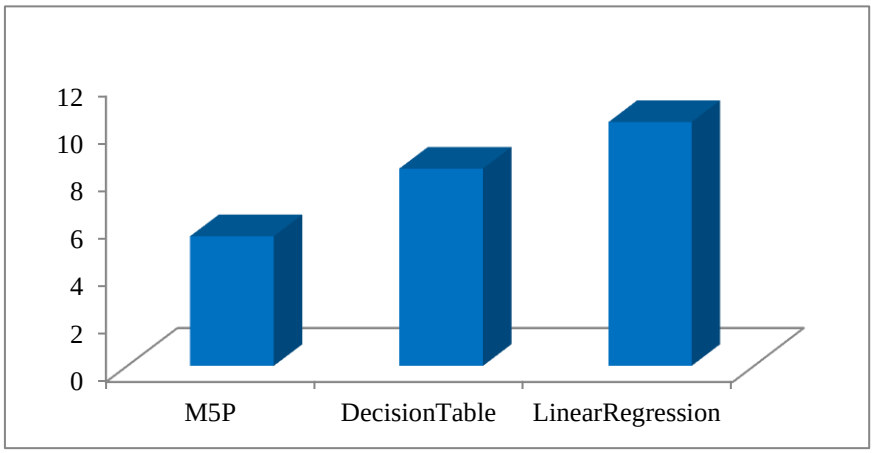


Figure 4.12 MAE results Comparison of Different Machine Learning Algorithms for the Year 2015 Dataset (for ERCOT market)

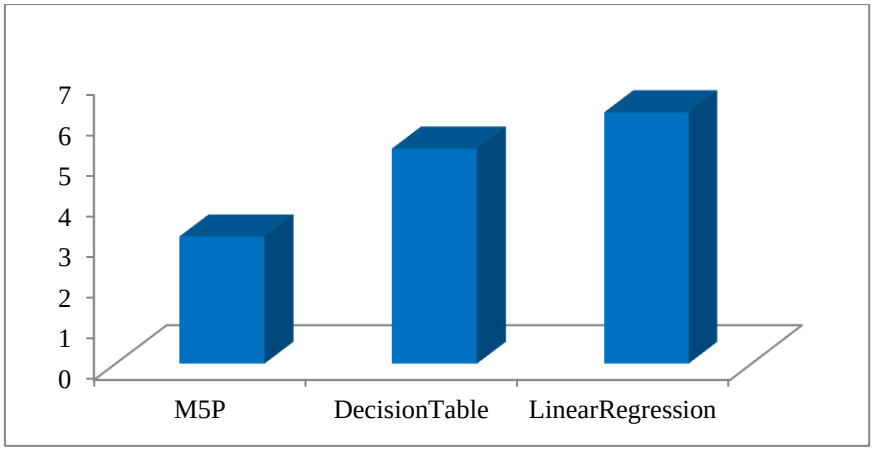


Figure 4.13 MAE results Comparison of Different Machine Learning Algorithms for the Year 2016 Dataset (for ERCOT market)

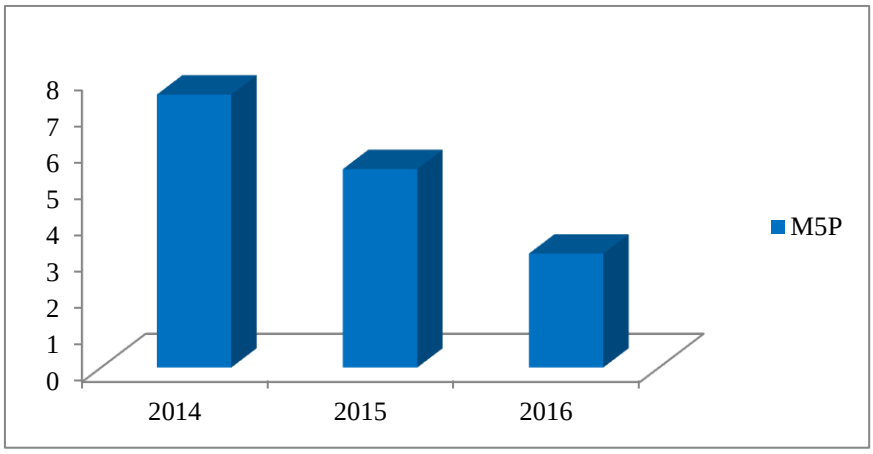


Figure 4.14 MAE results Comparison of M5P Machine Learning Algorithm for Three Annual Datasets (for ERCOT market)

There, according to the above experiment results, to predict the electricity price, the predictive model for each market developed by M5P trained with the year 2016 dataset was chosen.

4.4.2 Experimental Results of Selected Prediction Models

The prediction results are analyzed for each of three markets for twelve months of the year 2017, taken into account high and low periods of the electricity demand. MAE values achieved by using the chosen model for each market are displayed in Table 4.1.

Table 4.1 MAE Results of the Chosen Model for Each Market Predicted for the Year 2017

Months	ISONE	CAISO	ERCOT
January	5.75	2.88	4.75
February	4.41	3.67	3.55
March	6.12	6.26	4.59
April	4.01	7.12	5.12
May	5.48	9.16	5.97
June	5.19	12.14	8.30
July	5.56	5.55	11.72
August	4.47	13.82	7.90
September	5.63	11.21	5.22
October	5.25	10.20	5.16
November	6.74	7.76	4.41
December	10.92	3.77	3.09
Average	**5.79**	**7.79**	**5.82**

The average MAE value is 7.79 for the CAISO market where the maximum is 13.82 for August and the minimum is 2.88 for January. The average MAE value for ERCOT is 5.82, where the maximum MAE is 11.72 for July and the minimum MAE is 3.09 for December. The average MAE value is 5.79 for ISONE, the maximum MAE is 10.92 for December, and the minimum is 4.01 for April. The maximum MAE results are due to the peak prices whenever there are price volatility and fluctuations. Although the models can follow the regular price trend, they cannot predict price volatility and spikes.

Figures 4.15, 4.16, and 4.17 depict real-life and expected 24-hour electricity price sequences for those markets on January 2, 2017. Although there is a small difference in the real and predicted price values of each data center in each market, the relative order of the predicted electricity price sequences matches the relative order of the respective real-life electricity price sequences. If the relative order condition mentioned above holds true, the data center with the cheaper price can be correctly prioritized at any time of day. Using the predicted electricity price sequences of GDCs, energy cost minimization can be concerned.

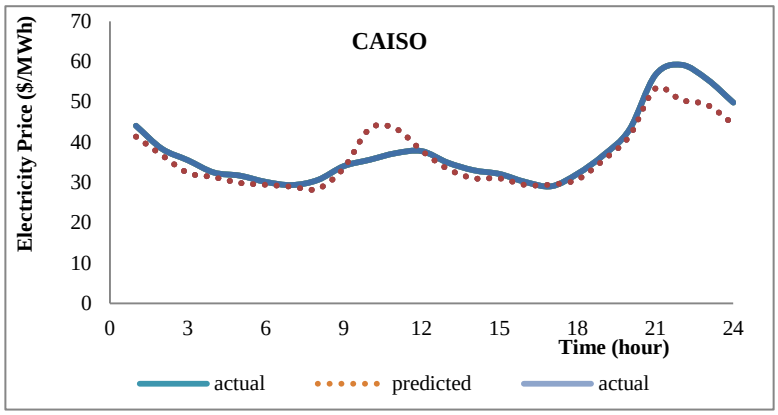


Figure 4.15 Actual and Predicted Electricity Price (for CAISO market)

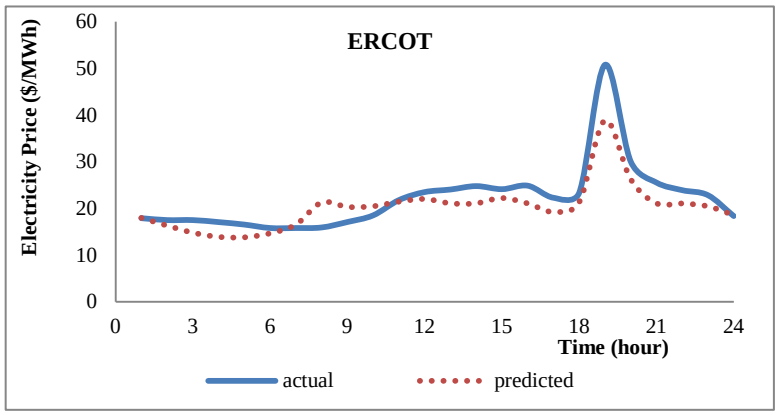


Figure 4.16 Actual and Predicted Electricity Price (for ERCOT market)

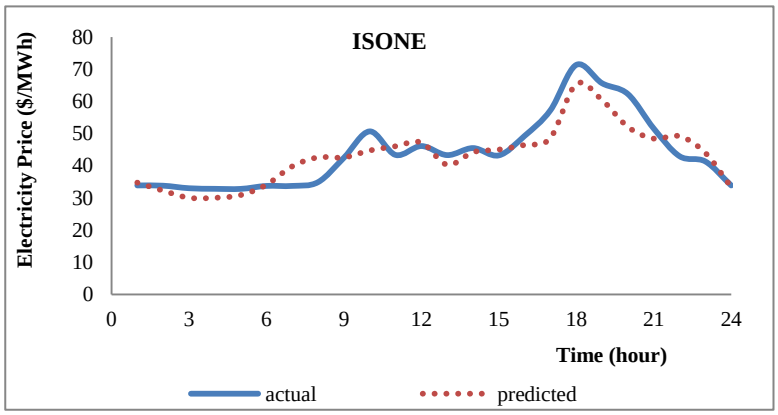


Figure 4.17 Actual and Predicted Electricity Price (for ISONE market)

In general, the models' results are quite reasonable, given that predicting electricity prices is difficult due to special characteristics of the price series such as non-constant mean, volatility, and seasonal impacts. Electricity prices can fluctuate over time, and no methods have been developed to accurately predict such fluctuations. The errors are reasonable in consideration of the dynamic existence of price data in the time series.

4.5 Chapter Summary

This chapter used Machine Learning techniques to predict electricity prices for GDCs in multi-regional electricity markets in the United States. Experiments are conducted with nine electricity price datasets (three annual years: 2014, 2015, and 2016 datasets for each of three markets: ISONE, CAISO, and ERCOT) and three machine learning algorithms: Linear Regression, M5P, and Decision Table. To forecast electricity prices in three markets, the most accurate predictive model is chosen. The model prediction results show that the M5P model outperforms the Linear Regression and Decision Table models developed on the same data. The experimental results demonstrate the performance and effectiveness of the chosen models for both annual trained data and machine learning algorithms. The chosen models offer promising accuracy in predicting electricity prices, which will help to reduce the cost of GDCs in multi-regional electricity markets.

CHAPTER 5

SLO GUARANTEED RESOURCE DEMAND PREDICTION FOR CLOUD DATA CENTERS

Cloud computing enables modern business owners to rent and use the services or resources that they require to run their businesses on a pay-as-you-go basis. Cloud data centers are typically large in size, with tens of thousands of servers [16]. Because of the fluctuating workload demand, the number of resources required for incoming requests is frequently dynamic. The key feature in data centers for handling dynamic workload nature resource provisioning ahead of needs is dynamic resource demand prediction. Predicting resource demand accurately is critical for allocating resources effectively and dynamically.

Infrastructure as a service (IaaS) allows resources that are available in the form of VMs. When VMs are statically provisioned in response to peak resource demand, it results in high resource costs for users and low resource utilization for cloud providers. As a result, it is critical to implement a proactive resource provisioning strategy to guarantee that cloud computing users have a good experience. Proactive resource provision can anticipate future resource demand and provide resources ahead of time.

Machine learning (ML) is a method of computational learning that has shown promising results in the domain of prediction. ML algorithms build the models by extracting knowledge from the data. Then, these models can make predictions on data. Machine Learning algorithms include hyper-parameters that define the model architecture and must be set before they can be run. The predictive performance of machine learning algorithms can be greatly influenced by hyper-parameter optimization. Model selection is an important task in ML because it determines the best parameters for the model. Optimization of hyper-parameters can have a significant impact on the predictive performance of Machine Learning Algorithms [29, 30, 85].

The cloud service provider must ensure that they have sufficient resources to meet the resource requirements for the incoming requests. Under-provisioning resources will result in SLO violations, resulting in significant financial penalties.

Over-provisioning wastes resources, resulting in lower utilization. To avoid both issues, the resource requirements of the cloud provider must be predicted ahead of time in order for the resource management system to adjust resource provisioning to meet the needs of customers. While achieving SLO, resource provisioning with the appropriate amount of dynamic resource demand becomes a critical issue.

This chapter describes SLO Guaranteed Resource Demand Prediction (SGRDP) system based on ML techniques on the Apache Spark Platform. In the system, we develop the resource prediction model to predict the number of CPU cores for the user requests. We compare the performance of powerful ML algorithms and choose the best predictive model to incorporate into our resource demand prediction procedure. The MAE is used to assess the model's performance. The CPU resource demand prediction model is developed on the selected predictive algorithm, and to achieve a high-accuracy prediction model, hyper-parameter optimization is performed. Real-world data center workload traces are used to evaluate the prediction model. SLO analysis based on CPU requirements of tasks is performed by padding a small percentage of maximum predicted value to guarantee SLO.

5.1 SLO Guaranteed Resource Demand Prediction System

This chapter proposes SLO guaranteed resource demand prediction (SGRDP) system that can predict CPU resource requirements needed for future requests. The framework of the SGRDP system is presented in Figure 5.1 that consists of two phases which are integrated resource demand prediction model development and Ensuring SLO. The CPU resource demand prediction model is built using the powerful Decision Tree (DT) machine learning algorithm, and to achieve a high-accuracy prediction model, hyper-parameter optimization for DT is performed.

SLO considered in this study is based on the criteria to meet the requested CPU cores of the submitted jobs. To avoid SLO violations, SLO analysis is performed over the prediction results by padding a small percentage of maximum predicted value which almost eliminates the under-provisioning of the prediction system. Resource demand prediction is developed for a cloud service provider while

not only providing high prediction accuracy but also providing SLO guarantee to their customers.

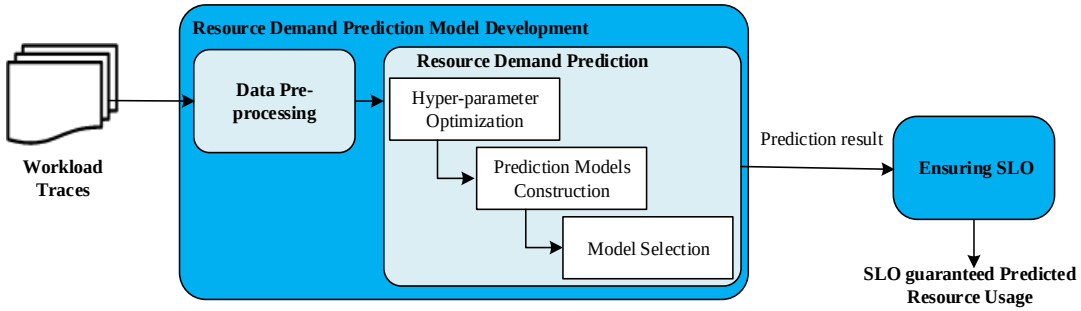


Figure 5.1 SLO Guaranteed Resource Demand Prediction System Architecture

5.2 System Environment

In this section, system specification for experimental environment and description of experiment workload traces are presented.

5.2.1 System Specification

The Spark data processing engine is built on a virtual machine (VM), in this study. Table 5.1 lists the device specifications as well as the required software components.

Table 5.1 System Specification

OS	Ubuntu 16.04 LTS
Host	Intel ® Core i7-7500U CPU @ 2.90GHz, 1TB Hard Disk, 8GB Memory
VM	100 GB Hard Disk, 4GB RAM
Software Component	Apache Hadoop 2.6.0 Apache Spark 1.5.2 Scala version 2.10.4

5.2.2 Experiment Workload Traces

We use three traces of jobs submitted to an HPC (High-Performance Computing) Cluster and Grid Computing available from the Parallel Workload

Archive [63] to evaluate the performance of the proposed system. Each workload trace (dataset) contains thousands of jobs. Table 5.2 shows a summary of these workload traces, and Table 5.3 describes their characteristics.

Table 5.2 Summary of Experiment Workload Traces

Datasets	No. of Jobs
DAS (Distributed ASCI in Netherlands Supercomputer)	225,711
RICC (RIKEN Integrated Cluster of Clusters)	447,794
MetaCentrum (Czech National Grid Infrastructure MetaCentrum)	103,656

Table 5.3 Features of Three Workload Traces

DAS	RICC	MetaCentrum
JobID	JobNo.	JobID
SubmitTime	ArrivalTime	SubmitTime
WaitTime	WaitTime	WaitTime
RunTime	RunTime	RunTime
NumCPUs	ReqCPUs	NumCPUs
UsedMemory	ReqMemory	UsedMemory
Status	Status	clusterID
UserID	UserNo.	
GroupID	GroupNo.	
QueueNo.		
PartitionNo.		

5.3 Resource Demand Prediction Model Development

Resource demand prediction model predicts the amount of resource to decide how many resources to allocate for each request in a data center in future. The system for constructing cloud data center resource demand prediction model is shown in Figure 5.2.

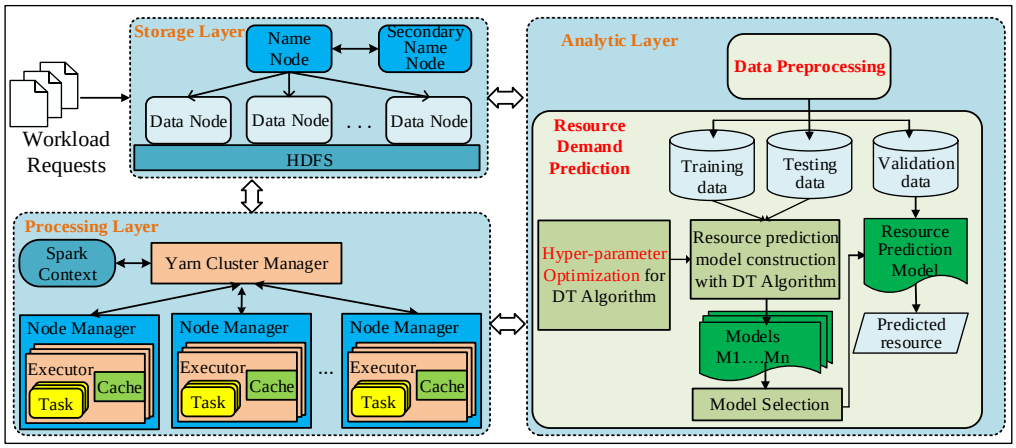


Figure 5.2 Proposed Resource Demand Prediction System Architecture

The proposed system for developing the resource demand prediction model employs the Hadoop Distributed File System (HDFS) [82], the Spark processing engine, and the Spark Machine Learning Library (Spark MLib) [76]. It is made up of three layers, each of which serves a specific purpose:

Storage Layer: HDFS is a scalable and dependable data storage system that is located in the Storage Layer. HDFS is a master/slave architecture, with a single NameNode serving as the master server. NameNode is responsible for carrying out file system namespace operations (i.e. opening, closing, and renaming files and directories). It is also in charge of the mapping of blocks to DataNodes. The actual data is stored in HDFS by DataNodes.

Processing Layer: Yarn Cluster Manager and the Spark processing engine are used to reliably and fault-tolerantly process massive amounts of data in parallel on commodity hardware clusters.

Analytic Layer: Data pre-processing, hyper-parameter optimization, model development and evaluation, and model selection are all performed to generate the resource demand prediction model. The Spark engine runs all Analytic Layer processes in parallel. Spark MLib is used to implement resource demand prediction procedures.

5.3.1 Resource Demand Data Pre-processing

This entails pre-processing historical workload traces by removing unnecessary data from raw data. It filters out or reduces noise data and replaces

missing values with the most frequently occurring value for that attribute. It identifies and removes outliers, as well as resolves inconsistencies. The experimental workload traces are divided into three disjoint parts: training data, testing data and also validation data for building prediction models, for assessing the generated models and for validating the selected model respectively.

5.3.2 Resource Demand Prediction

The MAE results of the models generated by three powerful ML techniques: Linear Regression (LR), Random Forest (RF) and Decision Tree (DT) algorithms are analyzed to select the most suitable prediction algorithm. All techniques are trained and tested using the pre-processed resource demand data. Table 5.4 shows the MAE results of the prediction models for all datasets.

Table 5.4 MAE Results of Three ML Algorithms

Datasets	LR	RF	DT
DAS	2.99	2.36	**1.59**
RICC	1.54	7.65	**1.35**
MetaCentrum	0.51	0.45	**0.38**

The MAE results in Table 5.4 show that the DT model performs better than the LR and RF models, as its results produce a smaller MAE for all datasets. DT is a promising model and thus, it is selected for predicting the amount of CPU resource requirements.

DT works reasonably well with default hyper-parameters values specified in software packages. However, choosing the most suitable values for the DT parameters can be framed in terms of hyper-parameters optimization to improve the performance of DT. The DT algorithm is used to generate resource demand prediction models based on all possible combinations of hyper-parameter pairs and all of the features provided in the training data.

All possible combinations of hyper-parameter values are set to generate the resource demand prediction models. The prediction results of the generated models are analyzed to determine which model has the lowest error in predicting resource demand. After selecting the best prediction model, it is validated to see if it can

predict the correct number of CPU demands. Figure 5.3 depicts the procedure for predicting resource demand.

Procedure: Resource, Demand, Prediction

Input*: workload traces*

Output*: predicted resource demand*

1. **Begin**
2. *Generate the prediction model by using default hyper-parameter values of DT*
   -Evaluate accuracy of the model (minMAE ←MAE)
3. **for** *(MaxDepth=0, MaxDepth<=30, MaxDepth ++)*
4. **for** *(MinInfoGain=0, MinInfoGain<=1, MinInfoGain +=0.1)*
5. *Generate the prediction model using MaxDepth and MinInfoGain for DT*
   -Evaluate accuracy of the model (print MAE)
6. **if** *(MAE< minMAE) minMAE ←MAE*
7. **endif**
8. **endfor** *// MinInfoGain*
9. **endfor** *// MaxDepth*
10. *Predict resource demand by using the selected model resulted with minMAE*
11. **End**

Figure 5.3 Procedure of Resource Demand Prediction

5.3.2.1 Hyper-parameter Optimization for DT algorithm

Hyper-parameter optimization for the Decision Tree (DT) algorithm is conducted over training data intended to develop the effective model. This investigates the best parameter values for DT.

In the hyper-parameter optimization approach, we must determine the parameter setting for the learning algorithm that will result in models with a low error rate.

The Maximum tree depth (MaxDepth) - the maximum depth to which the tree can grow - was chosen for optimization because it has been shown to affect Decision Tree performance [23]. The following parameter is Minimum Information Gain (MinInfoGain) - no split candidate leads to a greater information gain than

MinInfoGain [18]. For the corresponding datasets, the following step sizes and ranges are set for each of the chosen hyper-parameter values:

Maximum Tree Depth: A decision tree can have depths ranging from 0 to 30. MaxDepth is set to 31 different values ranging from 0 to 30 with a step size of 1.

Minimum Information Gain: There is no split candidate that results in a greater information gain than MinInfoGain. For all three datasets, MinInfoGain for split is set to 11 different values ranging from 0 to 1, with a step of 0.1.

For each dataset, these settings produce a uniform point cover of 341 (31 MaxDepth values x 11 MinInforGain values) different combination parameter sets.

5.3.2.2 Resource Demand Prediction Models Construction

The DT algorithm is used to generate prediction models based on all possible combinations of hyper-parameter pairs and all features provided in training data. In this study, 341 models are generated for each dataset with the same input and different parameters using all possible combinations of 31 MaxDepth values and 11 MinInfoGain values. The developed models' accuracy is evaluated using testing data.

5.3.2.3 Resource Demand Prediction Model Selection

For each dataset, the resource demand prediction model with the lowest error rate is chosen by analyzing the accuracy of the models generated with all possible combinations of hyper-parameter values. The selected model is fed validation data to see if it can predict the correct number of CPU demands.

5.4 Experimental Results for Resource Demand Prediction

The experimental results comparison and analysis for resource demand prediction are presented in this section.

5.4.1 Experimental Results of Prediction Models Generations

The accuracy of resource demand prediction varies depending on the parameter values and workload dataset characteristics. In order to generate

prediction models based on the training dataset, the DT algorithm is fed a set of all different combination parameter sets for each dataset. On a test dataset, the prediction performance of the models under consideration is evaluated, and the MAE obtained for each model, along with the set of parameter values that generated it, is recorded.

Table 5.5 displays the MAE values of 341 generated models against all possible combinations of hyper-parameter set pairs for the DAS dataset.

Table 5.5 MAE Results of Generated Models Using Different Combinations of Hyper-parameters (for DAS)

MaxDepth	MinInfoGain										
	0.0	***0.1***	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16	13.16
1	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96	5.96
2	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37	2.37
3	1.93	1.93	1.93	1.93	1.93	1.88	1.88	1.88	1.88	1.88	1.88
4	1.49	1.49	1.49	1.49	1.49	1.59	1.59	1.59	1.59	1.59	1.59
5	***1.59***	1.59	1.52	1.52	1.51	1.60	1.60	1.60	1.60	1.60	1.60
6	1.34	1.34	1.46	1.46	1.46	1.55	1.55	1.55	1.55	1.55	1.55
7	1.26	1.28	1.46	1.46	1.46	1.56	1.56	1.56	1.56	1.56	1.56
8	1.27	1.18	1.45	1.46	1.46	1.55	1.55	1.55	1.55	1.56	1.56
9	1.80	***1.17***	1.44	1.44	1.44	1.54	1.54	1.55	1.55	1.54	1.54
10	1.92	1.20	1.46	1.47	1.47	1.53	1.52	1.54	1.54	1.54	1.54
11	1.60	1.18	1.45	1.45	1.45	1.51	1.51	1.54	1.54	1.53	1.54
12	1.66	1.22	1.48	1.46	1.46	1.52	1.52	1.55	1.54	1.54	1.55
13	1.55	1.24	1.50	1.47	1.48	1.53	1.53	1.56	1.56	1.54	1.55
14	1.58	1.23	1.48	1.46	1.46	1.53	1.53	1.55	1.55	1.54	1.55
15	1.52	1.23	1.48	1.45	1.46	1.52	1.52	1.54	1.54	1.54	1.55
16	1.53	1.23	1.48	1.45	1.46	1.53	1.52	1.55	1.54	1.54	1.55
17	1.53	1.23	1.48	1.45	1.46	1.53	1.52	1.54	1.54	1.54	1.55
18	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
19	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
20	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
21	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
22	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
23	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
24	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
25	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55

26	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
27	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
28	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
29	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55
30	1.52	1.23	1.48	1.46	1.46	1.53	1.52	1.54	1.54	1.54	1.55

After analyzing the prediction results of 341 generated models for the DAS dataset with a maximum MAE of 13.16 and a minimum MAE of 1.17, the model with the lowest MAE was found to have a MinInfoGain of 0.1. As shown in Table 5.5, the error rate does not vary significantly with increasing MaxDepth values (above 16) for all MinInfoGain values.

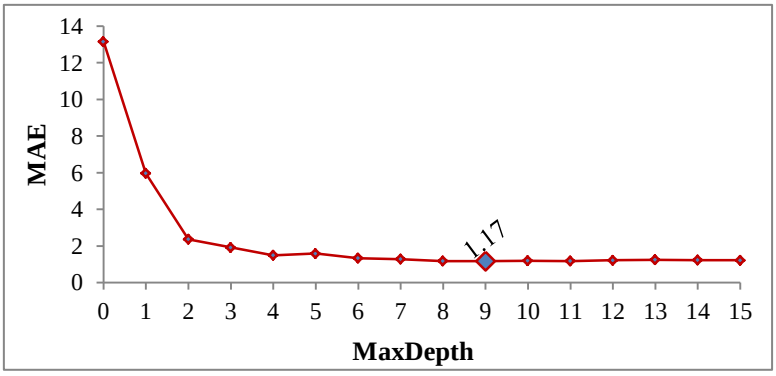


Figure 5.4 MAE Results of Generated Models Using Different Maxdepth Values with MinInfoGain = 0.1 (for DAS)

Figure 5.4 depicts the MAE results of generated models against 15 different MaxDepth values with the optimal MinInfoGain of 0.1 for the DAS dataset. For MaxDepth from 8 to 11, the error rate is generally lower. When the prediction results of the generated models are compared, the model with hyper-parameter pair (MaxDepth = 9 and MinInfoGain = 0.1) produces the best MAE of 1.17. Because it is the model with the lowest MAE, it is chosen to forecast future resource demand.

Table 5.6 displays the MAE values of 341 generated models against all combinations of hyper-parameter set pairs for the RICC dataset. After analyzing the prediction results of 341 generated models for the DAS dataset with maximum MAE of 21.69 and minimum MAE of 0.98, the model with the lowest MAE has MinInfoGain of 0. The error rate does not vary significantly with increasing MaxDepth values (above 16) for all MinInfoGain values, as shown in Table 5.6.

Table 5.6 MAE Results of Generated Models Using Different Combinations of Hyper-parameters (for RICC)

MaxDepth	MinInfoGain										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	21.69	21.69	21.69	21.69	21.69	21.69	21.69	21.69	21.69	21.69	21.69
1	15.69	15.69	15.69	15.69	15.69	15.69	15.69	15.69	15.69	15.69	15.69
2	8.16	8.16	8.16	8.16	8.16	8.16	8.16	8.16	8.16	8.16	8.16
3	3.40	3.40	3.40	3.40	3.40	3.40	3.40	3.40	3.40	3.40	3.40
4	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79
5	**1.35**	1.35	1.35	1.35	1.35	1.35	1.41	1.41	1.41	1.41	1.41
6	1.21	1.21	1.21	1.35	1.35	1.35	1.41	1.41	1.41	1.41	1.41
7	***0.98***	0.99	0.99	1.13	1.13	1.13	1.27	1.27	1.27	1.27	1.27
8	1.07	1.09	1.09	1.23	1.23	1.23	1.36	1.36	1.36	1.36	1.36
9	1.34	1.36	1.36	1.50	1.50	1.50	1.64	1.64	1.64	1.64	1.64
10	2.14	2.16	2.16	2.30	2.30	2.30	2.43	2.43	2.43	2.44	2.44
11	2.34	2.35	2.36	2.50	2.49	2.49	2.63	2.63	2.63	2.64	2.64
12	2.31	2.33	2.33	2.47	2.47	2.47	2.61	2.61	2.61	2.61	2.61
13	2.22	2.24	2.24	2.38	2.38	2.38	2.51	2.51	2.51	2.52	2.52
14	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
15	2.30	2.32	2.32	2.46	2.46	2.46	2.60	2.60	2.60	2.60	2.60
16	2.27	2.28	2.29	2.43	2.42	2.42	2.56	2.56	2.56	2.57	2.57
17	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
18	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
19	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
20	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
21	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
22	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
23	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
24	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
25	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
26	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
27	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
28	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
29	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56
30	2.26	2.28	2.28	2.42	2.42	2.42	2.56	2.56	2.56	2.56	2.56

After analyzing the prediction results of generated models for the RICC dataset with maximum MAE of 21.69 and minimum MAE of 0.98, the model with the lowest MAE has MinInfoGain of 0. For all MinInfoGain values, the error rate

does not vary significantly with increasing MaxDepth. Figure 5.5 clearly shows the MAE results of the generated models against 15 different MaxDepth values with the optimal MinInfoGain = 0 for the RICC dataset.

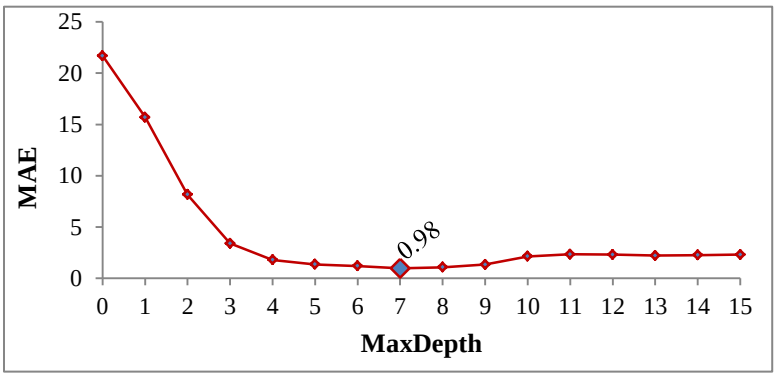


Figure 5.5 MAE Results of Generated Models Using Different Maxdepth Values with MinInfoGain = 0 (for RICC)

For MaxDepth from 5 to 9, the error rate is generally lower. When the prediction results of the generated models are compared, the model with hyper-parameter pair (MaxDepth = 7 and MinInfoGain = 0) produces the best MAE of 0.98. It is used to predict future resource demand because it has the lowest error.

Table 5.7 displays the MAE values of 341 generated models against combinations of hyper-parameter set pairs (0 to 30 MaxDepth values and 0.0 to 1.0 MinInforGain values) for the MetaCentrum dataset. After analyzing the prediction results of generated models for the MetaCentrum dataset with maximum MAE of 0.72 and minimum MAE of 0.25, the model with the lowest MAE has MinInfoGain of 0.0. As shown in Table 5.7, the error rate does not vary significantly with increasing MaxDepth values (above 16) for all MinInfoGain values.

Table 5.7 MAE Results of Generated Models Using Different Combinations of Hyper-parameters (for MetaCentrum)

MaxDepth	**MinInfoGain**										
	0.0	***0.1***	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72
1	0.64	0.64	0.64	0.64	0.64	0.64	0.72	0.72	0.72	0.72	0.72
2	0.54	0.54	0.54	0.54	0.62	0.62	0.72	0.72	0.72	0.72	0.72
3	0.49	0.52	0.52	0.52	0.62	0.62	0.72	0.72	0.72	0.72	0.72
4	0.42	0.50	0.50	0.50	0.62	0.62	0.72	0.72	0.72	0.72	0.72
5	**0.38**	0.47	0.47	0.48	0.61	0.62	0.72	0.72	0.72	0.72	0.72
6	0.34	0.45	0.46	0.48	0.62	0.62	0.72	0.72	0.72	0.72	0.72

7	0.53	0.45	0.47	0.49	0.62	0.62	0.72	0.72	0.72	0.72	0.72
8	0.38	0.44	0.46	0.48	0.61	0.62	0.72	0.72	0.72	0.72	0.72
9	0.28	0.42	0.45	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
10	0.32	0.42	0.45	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
11	***0.25***	0.42	0.45	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
12	0.51	0.41	0.44	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
13	0.52	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
14	0.50	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
15	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
16	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
17	0.53	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
18	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
19	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
20	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
21	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
22	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
23	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
24	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
25	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
26	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
27	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
28	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
29	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72
30	0.51	0.40	0.43	0.47	0.61	0.62	0.72	0.72	0.72	0.72	0.72

Figure 5.6 shows MAE results of the generated models against 15 different MaxDepth values with the optimal MinInfoGain of 0 for the MetaCentrum dataset.

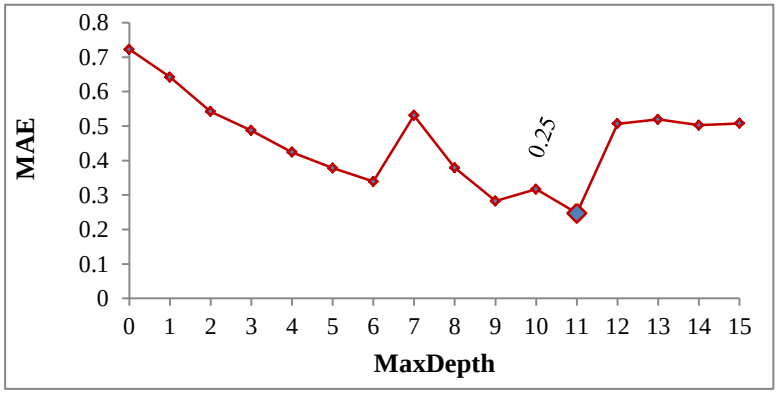


Figure 5.6 MAE Results of Generated Models Using Different Maxdepth Values with MinInfoGain = 0 (for MetaCentrum)

At MaxDepth of 6, 9, 10, and 11, the error rate is generally lower. When the prediction results of the generated models are compared, the model with the hyper-parameter pair (MaxDepth = 11 and MinInfoGain = 0) produces the best MAE of 0.25. Because it has the lowest error rate, it is used to predict the future resource demand.

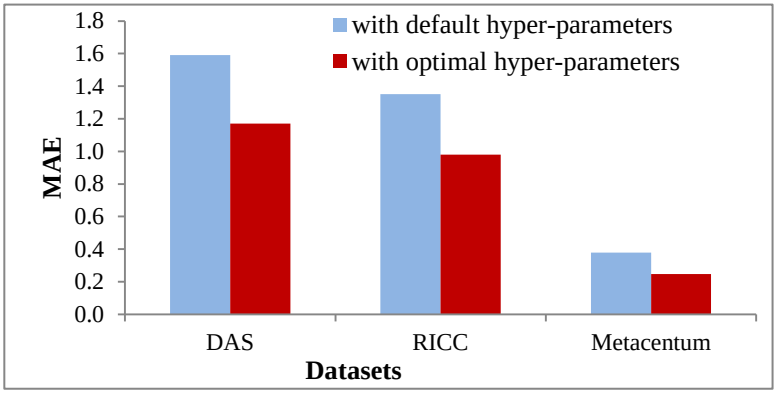


Figure 5.7 MAE Comparisons of Prediction Models with Default and Optimal Hyper-Parameters

MAE comparisons of prediction models generated by setting default and optimal hyper-parameters for all three datasets are shown in Figure 5.7. For the DAS dataset, the MAE value of the generated model using default hyper-parameters is 1.59, while the MAE value using optimal hyper-parameter pair is 1.17, saving MAE by 26%. For the RICC dataset, the MAE value of the generated model with default hyper-parameters is 1.35, while the MAE value with optimal hyper-parameter pair is 0.98, saving MAE by 28%. For the MetaCentrum dataset, the MAE value of the generated model with default hyper-parameters is 0.38 and 0.25 with optimal parameter pair, saving MAE by 35%. As a result, selecting the best hyper-parameters can have a significant impact on the performance of the resulting model. The results present that specifying the optimal hyper-parameter saves MAE by 26%, 28%, and 35% for the corresponding datasets, respectively.

5.4.2 Experimental Results of Selected Prediction Models

Figure 5.8 shows the actual and predicted CPU resource demand for the incoming requests of the validation dataset for DAS.

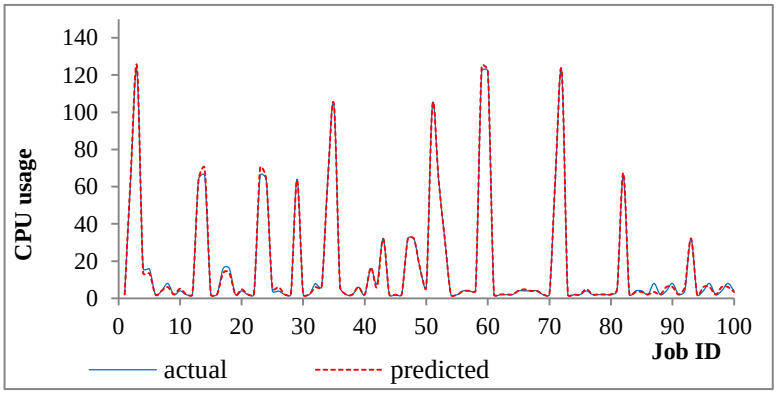


Figure 5.8 Actual and Predicted CPU Demand (for DAS)

The actual and predicted CPU demand for the incoming requests of the validation dataset for RICC is illustrated in Figure 5.9.

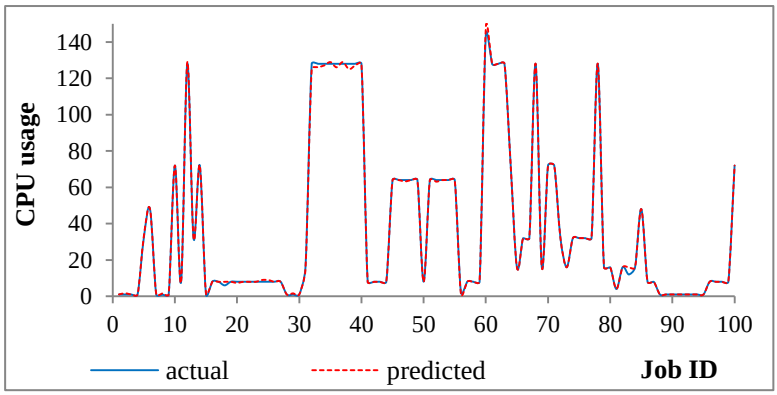


Figure 5.9 Actual and Predicted CPU Demand (for RICC)

The comparison of the actual and predicted CPU demand for the incoming requests of the validation dataset for MetaCentrum is presented in Figure 5.10.

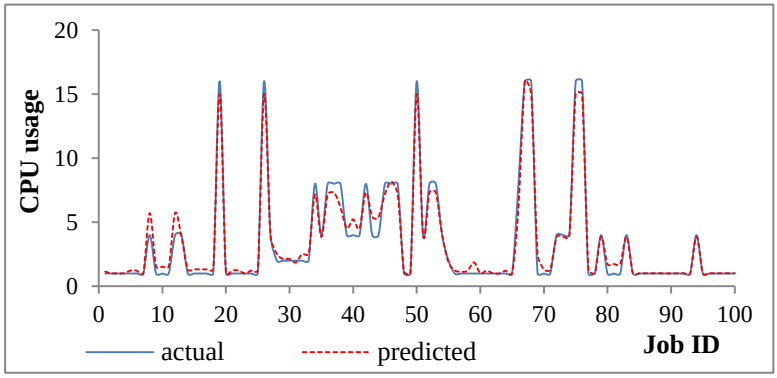


Figure 5.10 Actual and Predicted CPU Demand (for Metacentrum)

The results present that for all three workload traces, the actual and predicted CPU demand by our proposed models are nearly identical.

The MAE result for validation data of DAS is 0.39, that of RICC is 0.45 and that of MetaCentrum is 0.59. All MAE results are nearly zero and it can be found

that the CPU usage of actual and predicted by the proposed models for DAS and RICC workload traces are nearly identical. For MetaCentrum, the model produces a very small amount of prediction errors.

5.5 Ensuring SLO

In this study, SLO guaranteed between cloud service providers and consumers is based on the criteria to meet the requested CPU cores of the submitted jobs. The cloud provider must ensure that sufficient resources are available to meet customer demand. Alternatively, the provider will be required to compensate the customers who did not meet the performance criteria.

It is important to recognize that while predictive models that have a good predictive accuracy they eventually generate prediction errors. The prediction errors are due to underestimation or overestimation. If it is underestimated, certainly the users' resource demands are not met. This thesis prioritizes avoiding under-provisioning over avoiding over-provisioning because the former is more likely to result in SLO violations.

Figure 5.11 shows the under-provisioning frequencies after analyzing the actual and predicted value of three datasets. The under-provisioning frequencies are 73% for DAS, 52% for RICC and 67% for Meta Centrum respectively.

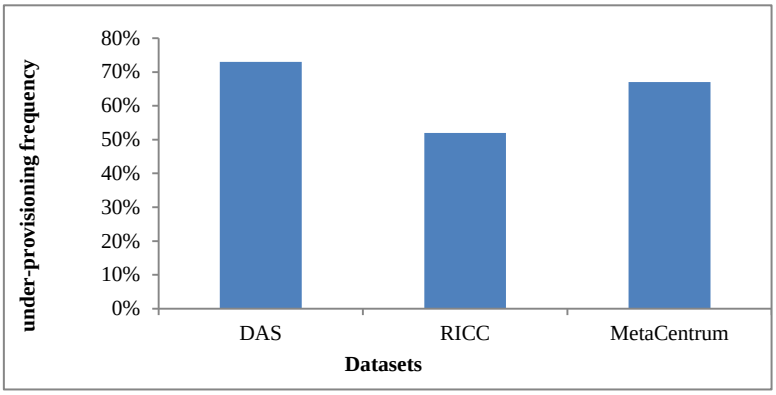


Figure 5.11 Under-provisioning Frequency of Predicted Values for Three Datasets

According to the prediction results shown in Figure 5.11, the proposed predictors are more under-provision than over-provision. To avoid this risk, using the predicted value generated from the model can be adjusted in an attempt to accommodate these prediction errors. SLO analysis is performed over the prediction results to avoid SLO violations by padding a small percentage of the maximum predicted value which almost eliminates the under-provisioning of the prediction system. The predicted value is increased with a small amount (1-5% in our experiment) of the maximum predicted value. We evaluated the SLO analysis results on resource demand prediction results of three workload traces over one-day prediction interval as shown in Figure 5.12.

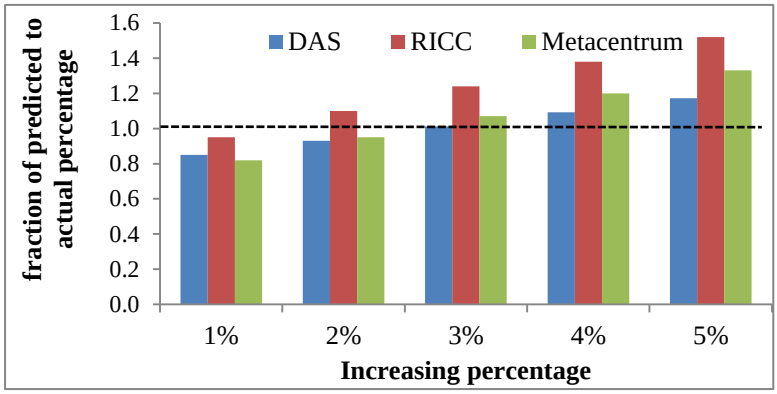


Figure 5.12 SLO Analysis Result of Three Workload Traces (one-day interval)

The fraction of the predicted to the actual value for one-day interval which is greater than or equals to 1 means it meets SLO. According to the analysis results shown in Figure 5.12, it can remove under-provisioning of data center and can meet SLO by increasing a small percentage with 3% of the maximum predicted value. The amount of SLO guaranteed predicted resource usage is applied in resource allocation to plan ahead of time for resource requirements.

5.6 Chapter Summary

Accurate resource demand prediction becomes a critical feature for efficient dynamic workload resource management. This chapter presented a model for predicting resource demand based on Apache Spark. Hyper-parameter optimization is performed to create a resource demand prediction model with a low error rate. The prediction system is experimented using real-world data center workload data from high-performance cloud computing paradigms. The results show that using the

optimal hyper-parameters to generate a prediction model can improve the resulting model's performance, saving MAE significantly for each corresponding dataset, respectively. It has been demonstrated that the proposed models' predicted CPU demand for all three workload traces are as identical as the actual CPU demand. SLO is ensured by padding 3% of the maximum predicted value, which eliminates the prediction system's under-provisioning of the prediction system.

CHAPTER 6

ENERGY-EFFICIENT AND COST-EFFECTIVE RESOURCE ALLOCATION FOR GEO-DISTRIBUTED DATA CENTERS

The powerful backend technology of cloud computing is virtualization technology, which achieves high-level availability, scalability, and agility of cloud computing features. Cloud infrastructure resources, such as storage, servers, memory, and network capacity, are virtualized and made available to customers as Virtual Machines (VMs) in cloud data centers. The allocation of resources for this environment is thus linked to the allocation and assignment of resources for incoming VMs.

To reduce total operational costs, both the energy efficiency of servers inside data centers and the location-based price variations in electricity must be considered. Resource allocation entails identifying and allocating resources to each incoming request in such a way that cloud user requirements are met while the cloud service provider's energy consumption or cost-saving goals are met. Because servers consume the majority of the power in a data center, power management methods are used to reduce power consumption by temporarily shutting down the server when it is idle.

This study aims to accomplish specific goals for the management of Data Center resources, such as cost-effectiveness and energy efficiency, by allocating resources in a way that minimizes the amount of energy needed to execute the given workload.

6.1 Experimental Evaluation of Energy-Efficient and Cost-Effective Resource Allocation

Energy-efficient and cost-effective resource allocation means allocating the incoming requests to the servers through VMs while optimization of energy use and price diversity awareness, in particular toward goals of saving energy consumption and lowering cost.

The energy-efficient and cost-effective resource allocation algorithm is proposed in this chapter to allow allocating resources to drop the energy cost of the Cloud infrastructure providers. Its approach is accomplished to save the energy consumption of the physical servers while lowering the cost of the cloud service providers.

Energy consumption is computed by multiplying the power consumption and the time turned on the servers.

$$E = P \cdot T \tag{6.1}$$

By considering the parameter of energy consumption (E) expressed as in (1), it can be minimized by reducing the power consumption (P) as well as also the time period (T) needed to turn on the servers.

The cost of electricity consumed during the period of operation reflects energy consumption, which is the main component of a data center's operating cost. To save energy and cost of data centers, this research considers energy efficiency factors and electricity prices differences across GDCs depending on their locations.

In evaluating the allocation algorithms, turnaround time is an important metric. Turnaround time is the time interval from the time since the task entered into the ready queue for execution until the task completed its execution. The average turnaround time is calculated as follow:

$$ATT = \frac{\sum_{i=1}^{N} TT_i}{N} \tag{6.2}$$

$$TT_i = FT_i - AT_i \tag{6.3}$$

where, ATT : Average Turnaround Time

TT_i : Turnaround Time of i^{th} job

N : Number of jobs

FT_i : Finish Time of i^{th} job execution

AT_i : Arrival time of i^{th} job

6.1.1 Simulation Platform

The proposed energy-efficient and cost-effective resource allocation algorithm is implemented as a new resource allocation optimization heuristic in the

CloudSim toolkit [17], for research work on the Cloud. CloudSim is a simulation toolkit (library) for cloud computing scenarios. Classes can be replaced or extended, new policies can be implemented, and new utilization scenarios can be added. CloudSim is not a ready-to-use in which we set parameters and collect data for the research. CloudSim, as a library, requires us to write a Java program that uses its components to create the desired scenario.

CloudSim is a scalable simulation framework that provides novel support for experimentation, modeling, and simulation of virtualized Cloud-based Data Center environments and Cloud management services for VMs, storage, bandwidth, and memory across a variety of domains, capabilities, and configurations. CloudSim includes a feature for configuring data center resources. CloudSim supports the simulation and modeling of large-scale Cloud Computing environments, such as service brokers, Data Centers, a virtualization engine, resource provisioning and allocation policies, network connections, and federated Cloud environments from both private and public providers. It allows modeling the cloud infrastructure providers with different hardware resources. It models the large scale data centers with the internal broker, physical servers, and virtual machines.

Because of these collective characteristics, CloudSim can be used to easily construct new heuristics to evaluate performance bottlenecks in resource management strategies relating to service delivery and provisioning policies. Despite the fact that CloudSim Architecture supports Cloud infrastructure service management, there is no consideration for energy consumption and cost minimization at the multi-data center level. Since the proposed resource allocation algorithm covers all these factors, we establish it into the CloudSim Architecture to enable energy-efficient and cost-effective resource allocation simulations for GDCs. The attachment of the proposed Energy-efficient and Cost-effective Resource Allocation Service for GDCs embedded in CloudSim Architecture is shown in Figure 6.1.

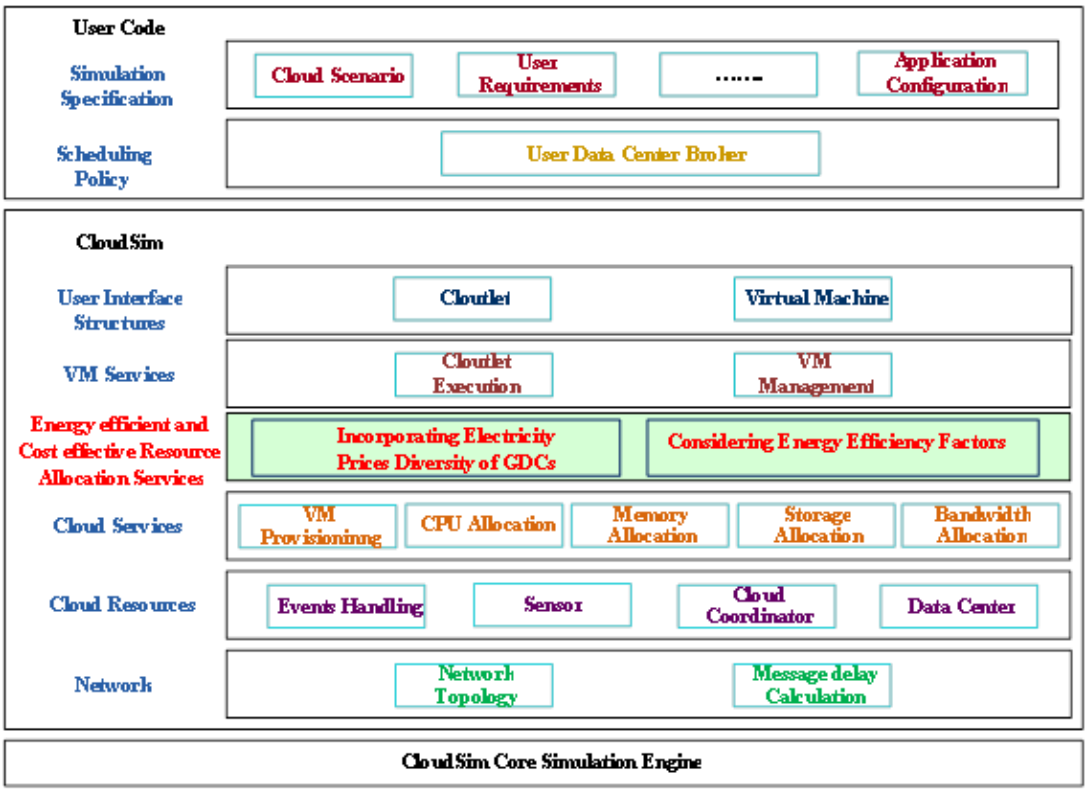


Figure 6.1 CloudSim Extension Architecture

6.1.2 System Model

We consider an IaaS provider with 3 data center sites located at three cities in US multi-regional electricity markets. Data centers Map website [20] was used to select these cities: Boston in New England, San Jose in California, and Dallas in Texas, each with three different time zones. Table 6.1 shows the five different characteristics of 45 heterogeneous physical servers in each data center.

Table 6.1 Server Types Characteristics

Server Name	Processor	Number of Servers for each Type	Number of Core	Power
Dell PowerEdge R720	Intel® Xeon® E5	9	16	750 W
HPE ProLiant DL560 Gen10	Intel Xeon-G 5120	9	28	800 W
Dell PowerEdge SR850	Intel® Xeon® Gold 6126	9	32	750 W
PowerEdge R840	Intel® Xeon® Gold 5122	9	64	750 W
PowerEdge R940	Intel® Xeon® Gold 5122	9	64	1100 W

To verify and validate the concept and the functionality of the proposed resource allocation algorithms, a cloud infrastructure environment is implemented. Then, to extend the basic CloudSim classes, new classes for the allocation algorithms are created. We model the problem so that High-Performance Computing jobs can be executed from logs of workload traces on parallel machines. The job requests are submitted to the cloud data centers for processing by the service broker, who acts on behalf of the cloud users. Because the focus of this work is on the allocation of VMs to hosts, a one-to-one mapping is used to assign cloudlets to VMs. VM termination is used during simulation to ensure that the simulation is completed. The proposed algorithms are used in the allocation of VMs to hosts. A space-shared policy is used to assign cloudlets to VMs, ensuring that tasks are executed sequentially in each VM. Each job unit has its own dedicated core under this policy. The PowerModel class is used to implement power management techniques. This class includes the getPower() function, which returns the host's power consumption based on the applied power management technique.

The system is made up of the following components, whose symbols are shown in Table 6.2.

Table 6.2 Symbols Description

Symbol	Description
n	Number of VM requests
m	Number of Data Center sites
l	Number of hosts in each Data Center site
P	Server power consumption
$\mathbf{E}_{k}$	Energy consumption of k^{th} host
$\mathbf{Host}_{k}$	k^{th} host in Data Center j
$\mathbf{Pr}_{j}$	Electricity price of Data Center j
$\mathbf{Vm}_{i}$	i^{th} virtual machine
$\mathbf{Length}_{i}$	runtime of i^{th} job request
$\mathbf{DC}_{j}$	Data Center j
$\mathbf{E}_{total}$	Total energy consumption of hosts in GDCs
$\mathbf{C}_{total}$	Total energy cost of all hosts in GDCs

We analyze three power management techniques with four power models in each of two allocation policies. After implementing two energy-saving resource allocation algorithms and comparing their energy consumption, EECERA algorithm is proposed for better resource allocation with less energy consumption and less cost of GDCs.

6.1.3 Incoming Resource Requests

The experiments are based on the incoming resource requests of three real workload traces (datasets): DAS, RICC and MetaCentrum which are the jobs from the logs of the workload traces as mentioned in Chapter 5.

6.2 Analysis for Energy Consumption in FCFS and SJF Allocation Policies

Energy consumptions of three power management techniques: DVFS, NPA, and PA, as well as four energy-aware power models: cubic, square, linear, and square root, are compared and analyzed in each of two allocation policies: FCFS and SJF, to determine which is the most efficient for better energy management in data centers.

In this experiment, the energy consumption metric is calculated and reported as energy consumption over time (kWh). The total amount of energy consumed by servers to execute the workload requests is referred to as energy.

6.2.1 Comparison for Energy Consumption of Power Management Techniques in FCFS

Figures 6.2-6.4 show the comparison of DVFS, NPA and PA techniques for FCFS policy in terms of energy consumption under different task numbers from each of three workload datasets: DAS, RICC and MetaCentrum.

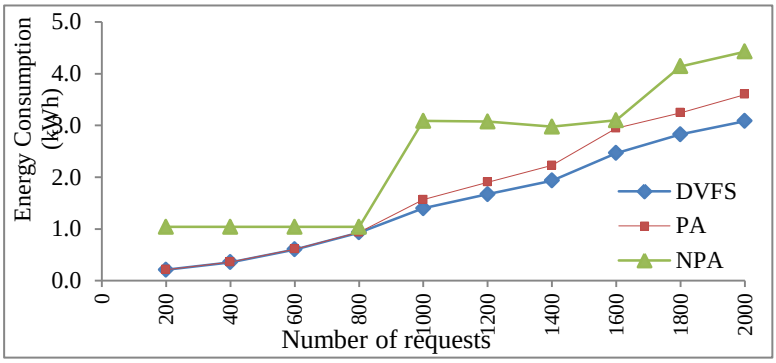


Figure 6.2 Comparison for Energy Consumption of Power Management Techniques in FCFS (for DAS)

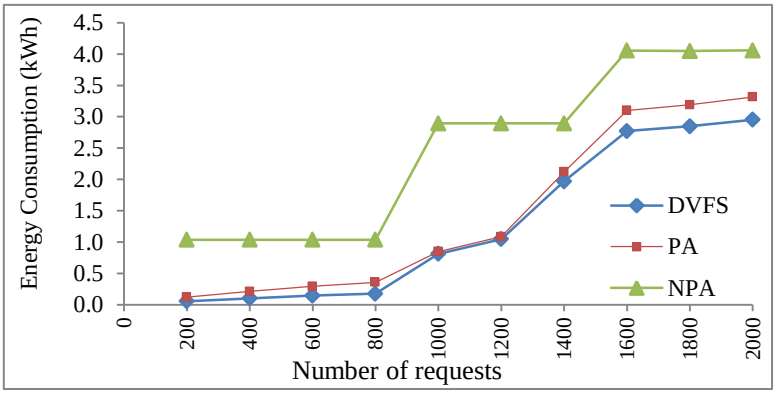


Figure 6.3 Comparison for Energy Consumption of Power Management Techniques in FCFS (for RICC)

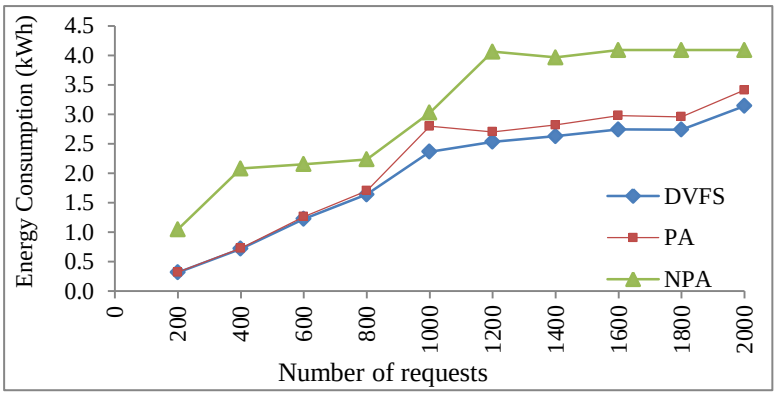


Figure 6.4 Comparison for Energy Consumption of Power Management Techniques in FCFS (for MetaCentrum)

The parameters: FCFS allocation policy and the number of data centers computational resources are fixed, and the number of tasks and power management techniques are variable. It can be seen that energy consumption of NPA is the highest. It increases significantly when the all servers of the next data center are activated for the increased number of tasks (as an example: RICC – when the number of tasks increases from 800 to 1000 and 1400 to 1600). The PA and DVFS

results show that energy consumption is also increasing simultaneously when the number of tasks increases, and their results are nearly the same for every number of tasks. DVFS has the least energy consumption because it consumes power based on the CPU utilization of the activated servers, whereas PA and NPA use maximum power for all servers, though PA supports shutting down the deactivated servers and NPA does not support. Therefore, DVFS can be used to save energy usage of servers for the FCFS policy.

6.2.2 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS

Tables 6.3-6.5 present energy consumption (kWh) with various power models: square root, linear, square and cubic of DVFS in FCFS allocation policy for the different number of requests for each of three datasets.

Table 6.3 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS (for DAS)

Number of Requests	square root	linear	square	cubic
200	0.21092	0.20887	0.20661	0.20563
400	0.35750	0.35662	0.35527	0.35435
600	0.60562	0.60406	0.60289	0.60259
800	0.93314	0.93242	0.93125	0.93037
1000	1.39992	1.39670	1.39478	1.39431
1200	1.67048	1.67048	1.67048	1.67048
1400	1.93523	1.93368	1.93142	1.92991
1600	2.46704	2.46623	2.46473	2.46338
1800	2.82794	2.82721	2.82599	2.82504
2000	3.08603	3.08438	3.08238	3.08138

Table 6.4 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS (for RICC)

Number of Requests	square root	linear	square	cubic
200	0.05716	0.05716	0.05666	0.05659
400	0.10476	0.10361	0.10275	0.10254
600	0.14685	0.14611	0.14538	0.14510
800	0.17843	0.17778	0.17700	0.17661
1000	0.81771	0.81339	0.80895	0.80704

1200	1.05271	1.04889	1.04533	1.04371
1400	1.97530	1.97027	1.96196	1.95536
1600	2.77189	2.77059	2.77025	2.77023
1800	2.84846	2.84661	2.84552	2.84533
2000	2.95514	2.95378	2.95194	2.95084

Table 6.5 Comparison for Energy Consumption of DVFS with Four Power Models in FCFS (for MetaCentrum)

Number of Requests	square root	linear	square	cubic
200	0.31797	0.31761	0.31693	0.31631
400	0.72143	0.71860	0.71494	0.71289
600	1.22641	1.22585	1.22485	1.22397
800	1.64079	1.63944	1.63781	1.63700
1000	2.36917	2.36558	2.35966	2.35510
1200	2.53431	2.53311	2.53126	2.52996
1400	2.63215	2.63003	2.62824	2.62771
1600	2.74487	2.74284	2.74066	2.73974
1800	2.73883	2.73872	2.7385	2.73829
2000	3.14415	3.14257	3.14049	3.13929

It is observed that DVFS with cubic power model consumes the least energy consumption model among the power models for the requests. Therefore, the cubic model is applied for DVFS in FCFS policy to minimize the energy consumption of servers in the data centers.

6.2.3 Comparison for Energy Consumption of Power Management Techniques in SJF

Figures 6.5-6.7 show energy consumption comparisons of DVFS, NPA and PA techniques in the SJF policy for the different number of tasks from each of three workload datasets.

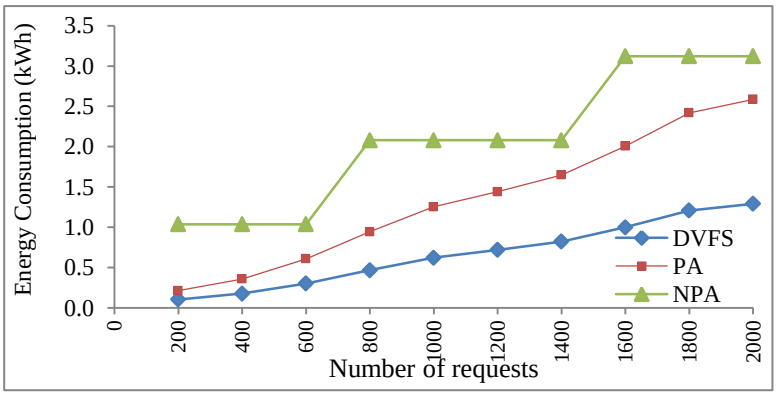


Figure 6.5 Comparison for Energy Consumption of Power Management Techniques in SJF (for DAS)

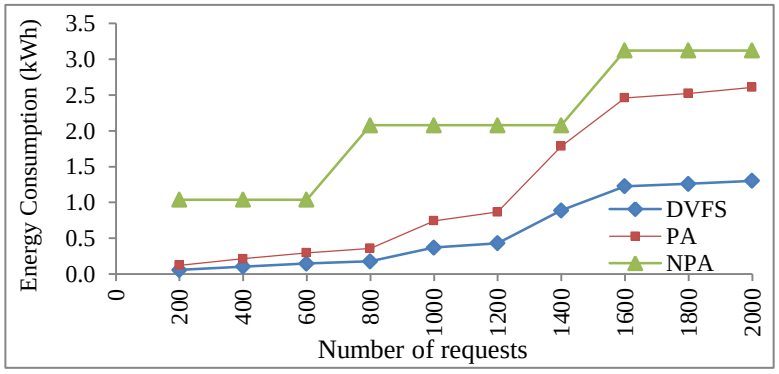


Figure 6.6 Comparison for Energy Consumption of Power Management Techniques in SJF (for RICC)

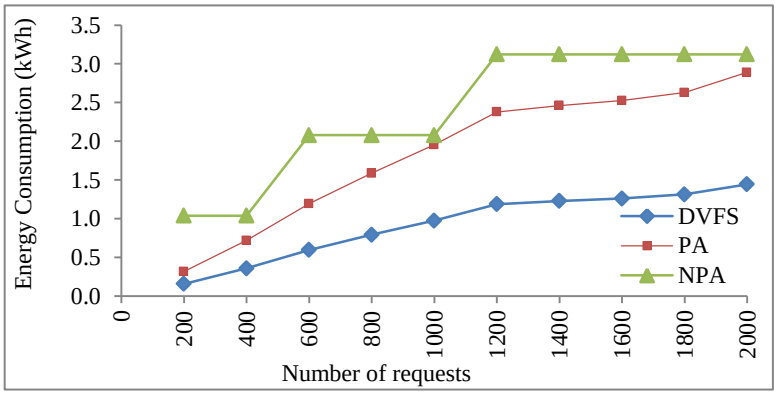


Figure 6.7 Comparison for Energy Consumption of Power Management Techniques in SJF (for MetaCentrum)

The parameters: SJF allocation policy and the number of data centers computational resources are fixed, and power management techniques and the number of tasks are variable. It is observed that NPA consumes the highest energy. It increases significantly when the all servers of the next data center are activated for the increased number of tasks (as an example: MetaCentrum – when the number of tasks increases from 400 to 600 and 1000 to 1200). The PA and DVFS results

show that the energy consumption is also increasing simultaneously when the number of tasks increases. It is clearly found that DVFS is the least energy consumption technique than PA and NPA because these techniques use the maximum power for all activated servers, while DVFS consumes power based on the CPU utilization of the activated servers, and shuts down the unutilized servers. Therefore, a large amount of energy is saved through DVFS technique for SJF allocation policy.

6.2.4 Comparison for Energy Consumption of DVFS with Four Power Models in SJF

Tables 6.6-6.8 Tables 6.3-6.5 present energy consumption (kWh) with various power models: square root, linear, square and cubic of DVFS power management technique in SJF policy for the different number of requests for each of three datasets.

Table 6.6 Comparison for Energy Consumption of DVFS with Four Power Models in SJF (for DAS)

Number of Requests	square root	linear	square	cubic
200	0.10550	0.10447	0.10334	0.10285
400	0.17881	0.17837	0.17770	0.17723
600	0.30291	0.30212	0.30154	0.30139
800	0.46936	0.46750	0.46573	0.46510
1000	0.62455	0.62300	0.62122	0.62027
1200	0.72019	0.72019	0.72019	0.72019
1400	0.82388	0.82329	0.82234	0.82161
1600	1.00082	0.99899	0.99706	0.99620
1800	1.20950	1.20918	1.20865	1.20823
2000	1.29253	1.29188	1.29110	1.29071

Table 6.7 Comparison for Energy Consumption of DVFS with Four Power Models in SJF (for RICC)

Number of Requests	square root	linear	square	cubic
200	0.05821	0.05716	0.05666	0.05659
400	0.10476	0.10361	0.10275	0.10254
600	0.14685	0.14611	0.14538	0.14510
800	0.17843	0.17778	0.17700	0.17661
1000	0.37145	0.37133	0.37110	0.37090
1200	0.43174	0.43095	0.42990	0.42927

1400	0.89015	0.88819	0.88470	0.88168
1600	1.22865	1.22711	1.22519	1.22418
1800	1.26036	1.25903	1.25724	1.25616
2000	1.30356	1.30326	1.30276	1.30235

Table 6.8 Comparison for Energy Consumption of DVFS with Four Power Models in SJF (for MetaCentrum)

Number of Requests	square root	linear	square	cubic
200	0.15832	0.15815	0.15783	0.15752
400	0.35828	0.35790	0.35719	0.35656
600	0.59601	0.59571	0.59511	0.59455
800	0.79246	0.79181	0.79081	0.79010
1000	0.97508	0.97401	0.97295	0.97242
1200	1.18847	1.18804	1.18739	1.18693
1400	1.22934	1.22856	1.22791	1.22771
1600	1.26101	1.26030	1.25953	1.25921
1800	1.31430	1.31425	1.31416	1.31406
2000	1.44378	1.44317	1.44235	1.44188

The cubic power model is the most energy-efficient one among the power models for the job requests. Therefore, the cubic model is applied for DVFS in SJF policy to minimize the energy consumption of servers in the data centers.

According to the above results, DVFS with cubic model consumes less energy in both allocation policies and thus it is selected to employ in energy-saving resource allocation algorithms.

6.3 Energy-Saving Resource Allocation Algorithms

Two energy-saving resource allocation algorithms: DFCFS and DSJF are implemented to address the resource allocation in multiple VMs and compared to maintain a minimized turnaround time and minimized energy consumption using CloudSim.

Algorithm 1 and 2 describe DVFS and DSJF Algorithms, respectively. The data center broker creates a list to receive the jobs (cloudlets). Virtual machine (VMs), where VM= {vm1, vm2, ..., vmn} are created, and the broker maps a job to a VM based on one-to-one mapping configuration. A job request contains the features which include the runtime (length), the number of CPU cores, the required memory size etc., as mentioned in Table 5.2. When a job request arrives, the broker

has (DC x Host) different VM placement options. It submits the VM to the chosen data center and host.

If the host has sufficient resources for the VM, it computes the average turnaround time, as shown in (6.2), and (6.3). The energy consumption is computed as a function of CPU usage and is regulated dynamically based on DVFS.

6.3.1 DVFS Enabled First Come First Serve (DFCFS) Algorithm

Figure 6.8 presents DFCFS algorithm that is based on FCFS policy combined with DVFS power management technique.

Algorithm 1: DFCFS Algorithm

Input:
Job_List *// the list of job requests*
Host_list *// the list of hosts in a data center*
DC_list *//the list of available data centers in geographic sites*
Output:
Destinations *[*$Host_k$ *in* DC_j*]*
ATT *// average turnaround time*
E_{total} *// total energy consumption of hosts in data centers*
1 **BEGIN**
2 *Create* VMs *as the number of job requests in* Job_List
3 **foreach** Job *in* Job_List **do**
4 *Assigned* job_i → VM_i
5 **end for**
6 **for** *all* VM *i* ← *1* **to** *n* **do**
7 **for** *all* DC *j* ← *1* **to** *m* **do**
8 **for** *all* Host *k* ← *1* **to** *l* **do**
9 **if** *CPU core of* VM_i <= *remaining CPU core of* $Host_k$ **then**
10 *Assigned* VM_i → $Host_k$ *in* DC_j
11 *Calculate the remaining CPU core of* $Host_k$
12 **else** *Start* $Host_{k+1}$ *in* DC_j
13 **end if**
14 **end for**
15 **end for**
16 **end for**
17 *Return* Destinations
18 *Calculate* ATT
19 *Calculate* E_{total} *by* DVFS with cubic model
20 **END**

Figure 6.8 DFCFS Algorithm

6.3.2 DVFS Enabled Shortest Job First (DSJF) Algorithm

To calculate the energy consumption of the servers, SJF allocation policy is applied and DVFS power management technique is used in DSJF algorithm presented in Figure 6.9.

Algorithm 2: DSJF Algorithm

Input:
Job_List *// the list of job requests*
Host_list *// the list of hosts in a data center*
DC_list *//the list of available data centers in geographic sites*
Output:
Destinations *[*$Host_k$ *in* DC_j*]*
ATT *// average turnaround time*
E_{total} *// total energy consumption of hosts in data centers*
1**BEGIN**
2 *Sort* Job_List *in ascending order based on their* Length
3 **if** $Length_i$ = $Length_{i+1}$ **then**
4 *Sort* Job_List *according to their* Arrival Time
5 **end if**
6 *Create* VMs *as the number of job requests in* sorted Job_List
7 **foreach** Job *in* sorted Job_List **do**
8 *Assigned* job_i → VM_i
9 **end for**
10 **for** *all* VM $i \leftarrow 1$ **to** *n* **do**
11 **for** *all* DC $j \leftarrow 1$ **to** *m* **do**
12 **for** *all* Host $k \leftarrow 1$ **to** *l* **do**
13 **if** *CPU core of* VM_i <= *remaining CPU core of* $Host_k$ **then**
14 *Assigned* VM_i → $Host_k$ *in* DC_j
15 *Calculate the remaining CPU core of* $Host_k$
16 **else** *Start* $Host_{k+1}$ *in* DC_j
17 **end if**
18 **end for**
19 **end for**
20 **end for**
21 *Return* Destinations
22 *Calculate* ATT
23 *Calculate* E_{total} *by* DVFS with cubic model
24 **END**

Figure 6.9 DSJF Algorithm

6.3.3 Comparison for Energy Consumption of DFCFS and DSJF

Figures: 6.10-6.12 show the comparisons of energy consumption for the different number of requests from three workload datasets under DFCFS and DSJF algorithms.

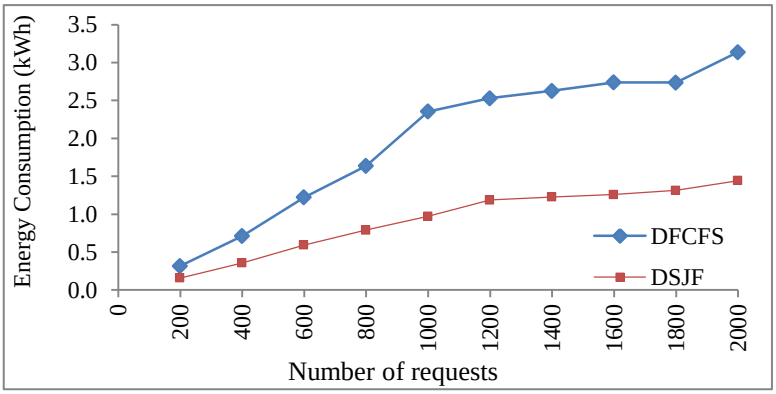


Figure 6.10 Comparison for Energy Consumption of DFCFS and DSJF (for DAS)

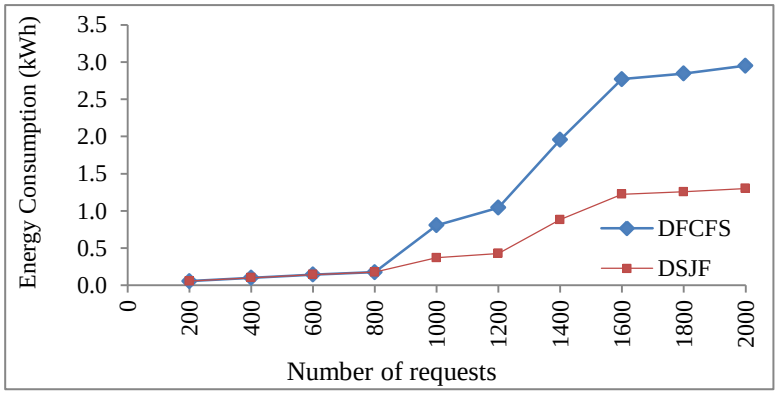


Figure 6.11 Comparison for Energy Consumption of DFCFS and DSJF (for RICC)

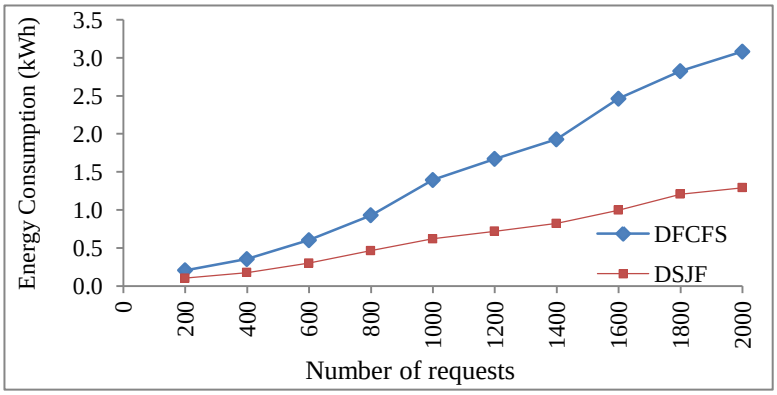


Figure 6.12 Comparison for Energy Consumption of DFCFS and DSJF (for MetaCentrum)

The energy consumed for 200 to 800 requests of RICC dataset under DFCFS and DSJF algorithms is almost nearly the same. The reason for this is that the

runtimes (lengths) of those requests are nearly equivalent. As mentioned above, DSJF sorts the runtimes of the requests in ascending order before execution while DFCFS does not. Although DSJF algorithm sorts the lengths of the requests having similar lengths with each other, there is no effect of sorting the requests based on lengths. For the requests over 800 number of RICC dataset and for all the requests of the other two datasets (DAS and MetaCentrum) having different lengths with each other, it is observed the effect of DSJF that can save energy consumption compared to DFCFS. DSJF algorithm gains higher energy efficiency for the incoming requests with different lengths. When compared to the DFCFS algorithm, DSJF algorithm can save up to 55% of the energy.

The shorter average turnaround time of the allocation method produces lower energy consumption. The turnaround time is the main factor influencing energy consumption. The average turnaround times (in seconds) for executing the requests of three datasets under DFCFS and DSJF algorithms are presented in the following Table 6.9.

Table 6.9 Average Turnaround Time of DFCFS And DSJF Algorithms

Datasets	DFCFS	DSJF
DAS	204	86
RICC	164	77
MetaCentrum	442	86

By comparing the average turnaround time obtained from DFCFS and DSJF as shown in Table 6.3, the turnaround time of DSJF is almost 50% less than DFCFS. DSJF helps to reduce the turnaround time of the tasks as it first fulfills the request with the shortest time and it takes the shortest possible time to finish. The VM can then move on to the next selected task. This will reduce the amount of time as well as the number of active VMs and also the number of active servers, resulting in lower energy consumption. In this way, DSJF algorithm that takes the minimum turnaround time shows its contribution to energy-saving.

The aim of this comparison is to minimize the turnaround time while reducing the energy consumption of the computing resources. DSJF algorithm provides enhanced performance in terms of minimum average turnaround time as well as low energy consumption.

6.4 Proposed Energy-Efficient and Cost-Effective Resource Allocation (EECERA) Algorithm

Lowering the high operating costs of data centers is one of the problems that the cloud providers face. Fortunately, the geographical distribution of data centers exposes the opportunity for cost-saving. The data centers are exposed to different electricity prices markets and they pay different electricity prices. To reduce operational cost of GDCs, both the energy efficiency of the servers inside data centers and the location-based electricity prices diversity need to be considered.

EECERA algorithm as described in Figure 6.13 is proposed for energy-efficient and cost-effective resource allocation. The key idea is to shift resource allocation to energy-efficient servers as well as to locations associated with comparatively lower electricity price. In order to lower energy cost exploiting the electricity price variations across regions under multiple electricity price markets environment, our policies attempt to submit each arriving job to the data center that leads to the lowest cost.

To be energy-efficient, EECERA deploys DSJF algorithm applied with SJF allocation policy and DVFS power management technique. Moreover, it attempts to save the energy cost based on price diversities of GDCs. Three GDCs in multi-regional electricity markets where electricity prices vary by location, as described in section 6.1.2 are considered. The data center j electricity price is denoted as Pr_j, and all servers in DC_j must share the Pr_j electricity price. It routes the incoming requests to the cloud data center with the lowest electricity price first by sorting the data center list in ascending order based on their electricity prices, as line number 10 of Algorithm 3 shown in Figure 6.13. It calculates the total energy consumption of the servers in data centers that support the VMs. The total energy cost is computed as (6.4) by summing the energy costs of all servers across all GDCs, i.e.,

$$C_{total} = \sum_{j=1}^{m} (\sum_{k=1}^{l} E_k) \cdot Pr_j \qquad (6.4)$$

Algorithm 3: EECERA Algorithm

Input:
#Job_List *// the list of job requests*
Host_list *// the list of hosts in a data center*
DC_list *//the list of available data centers in geographic sites*
Output:
Destinations *[*$Host_k$ *in* DC_j*]*
E_{total} *// total energy consumption of the hosts in the data centers*
C_{total} *// total energy cost of hosts in data centers*
1 **BEGIN**
2 *Sort* Job_List *in ascending order based on their* Length
3 **if** $Length_i = Length_{i+1}$ **then**
4 *Sort* Job_List *according to their* Arrival Time
5 **end if**
6 *Create* VMs *as the number of job requests in* sorted Job_List
7 **foreach** Job *in* sorted Job_List **do**
8 *Assigned* $job_i \rightarrow VM_i$ *// Add user job request to* VM *based on one-to-one mapping configuration*
9 **end for**
10 *Sort* DC_list *in ascending order based on their* Pr_j
11 **for** *all* VM $i \leftarrow 1$ **to** *n* **do**
12 **for** *all* DC $j \leftarrow 1$ **to** *m* **do**
13 **for** *all* Host $k \leftarrow 1$ **to** *l* **do**
14 **if** *CPU core of* VM_i <= *remaining CPU core of* $Host_k$ **then**
15 *Assigned* $VM_i \rightarrow Host_k$ *in* DC_j
16 *Calculate the remaining CPU core of* $Host_k$
17 **else** *Start* $Host_{k+1}$ *in* DC_j
18 **end if**
19 **end for**
20 **end for**
21 **end for**
22 *Return* Destinations
23 *Calculate* E_{total} *by* DVFS with cubic model
24 *Calculate* C_{total}
25 **END**

Figure 6.13 Energy-Efficient and Cost-Effective Resource Allocation (EECERA) Algorithm

Three workload traces are used to compare the proposed EECERA algorithm with respect to cost minimization with energy-efficient resource allocation (EERA) algorithm that considers only energy efficiency factors as DSJF but does not consider the order of GDCs' electricity price diversities as line number 10 of Algorithm 3.

Figures 6.15 to 6.17 show energy cost comparisons of EECERA and EERA that allocates 2000 VM requests from each of three workload datasets knowing

hourly electricity prices of three electricity markets: ISONE, CAISO and ERCOT shown in Figure 6.14.

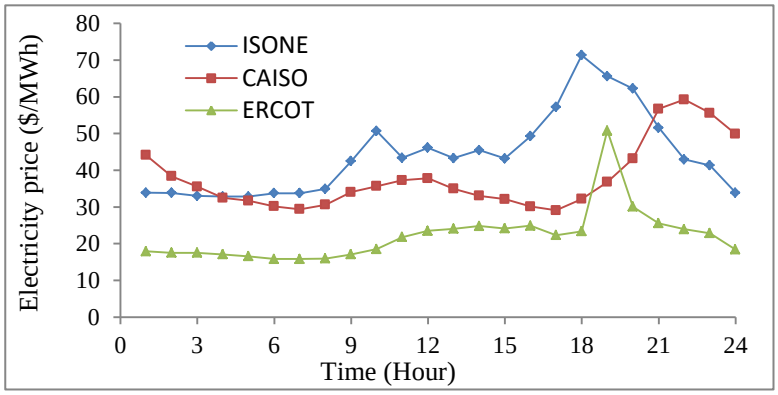


Figure 6.14 Hourly Prices of Data Centers in Multi-region Electricity Markets

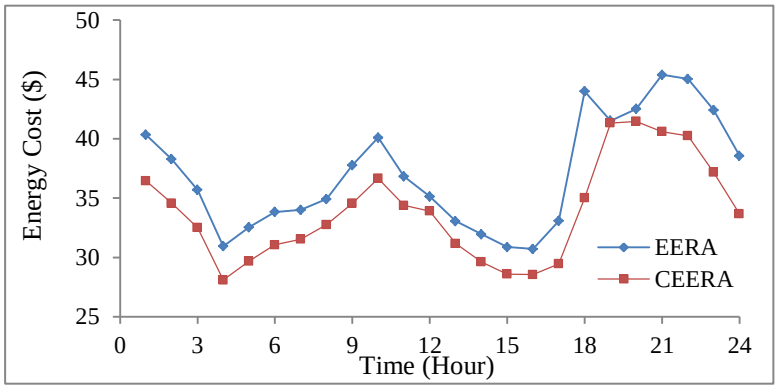


Figure 6.15 Total Energy Cost (for DAS)

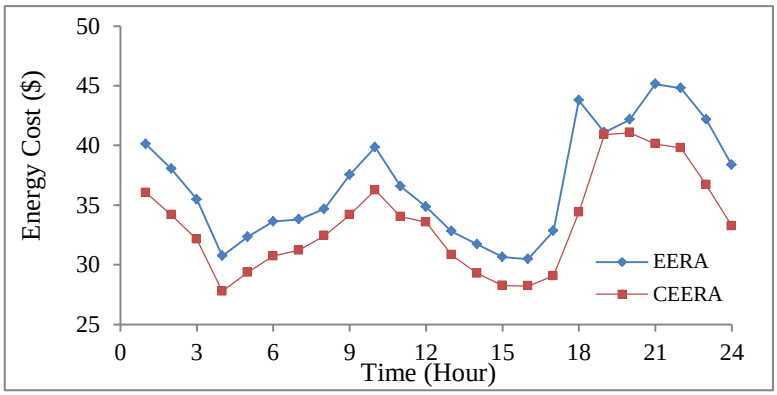


Figure 6.16 Total Energy Cost (for RICC)

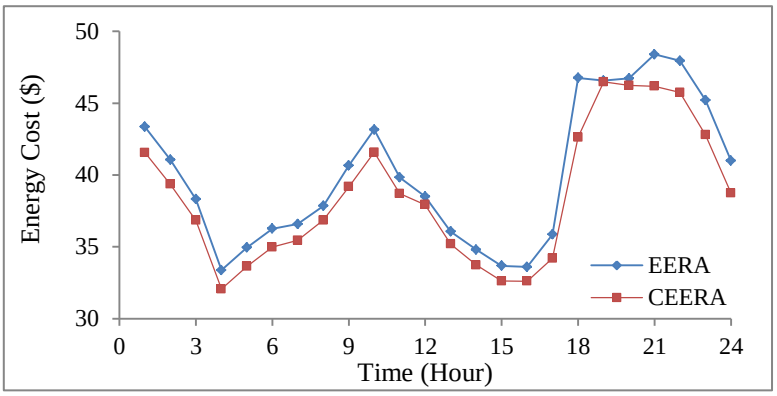


Figure 6.17 Total Energy Cost (for MetaCentrum)

EECERA Algorithm directs more workload requests to the cloud data center with the cheapest electricity price first, knowing the future electricity prices of GDCs so that the energy cost can be saved. According to the results experimented on three real workload traces shown in Figures 6.15-6.17, the total energy cost of EECERA is significantly lower than EERA in every hour. With extensive simulations based on the electricity prices of GDCs shown in Figure 6.14, it is shown that EECERA reduce the electricity cost by 14%, 13% and 6% for each of three -workload traces respectively. This further proves that EECERA algorithm provides a practical approach to lowering electricity costs for multiple GDCs in multi-regional electricity markets with price diversities.

6.5 Chapter Summary

By using CloudSim simulator, the researcher investigates the promising energy efficiency of cubic power models, DVFS in three power management techniques and SJF in two resource allocation policies. After analyzing two -energy efficient resource allocation algorithms, it can be found that the energy consumption of resource allocation with DSJF algorithm is lower about 55% than DFCFS algorithm. GDCs' electricity prices in multi-regional markets are predicted to deliver the incoming requests to data centers at cheaper electricity prices firstly. The proposed EECERA algorithm is compared with EERA algorithm for energy cost minimization. Evaluation results show that we achieve both minimized energy consumption and cost reduction in our resource allocation techniques.

CHAPTER 7

CONCLUSION AND FUTURE DIRECTIONS

Multi-tenant environments, large scale environments, and heterogeneous resources in geographically dispersed data centers present management and monitoring challenges. The resource management problems range from managing the efficient allocation of heterogeneous resources and jobs scheduling that are also mapped for a given resource to dealing with both system and workload uncertainties.

The overall workload of data centers is constantly changing, as the resource demand to cloud data centers is often dynamic. Providing resources with the right amount of dynamic resource demand while satisfying SLOs in cloud data centers becomes a critical issue. A tremendous amount of energy consuming which causes to high operating costs of cloud computing has been recognized as an emerging issue.

This chapter describes this study work on energy-efficient and cost-effective management of resources in GDCs and highlights the research findings. It also describes avenues for future research directions in this studying area.

7.1 Thesis Summary

This section summarized the previous chapters' research work in order to gain a better understanding of the proposed system.

The background theory related to cloud computing is listed first in Chapter 1 which is the research work's preamble. The thesis motivations that lead to the necessity of the research work are then explained. Finally, the main part of the thesis: the objectives and contributions of the proposed system is presented. It also includes a detailed explanation of the proposed framework for energy-efficient and cost-effective resource management. The chapter includes a conceptual design of the proposed system as well as an explanation of the design requirements. Then comprehensive descriptions of the proposed system and framework and its components are discussed.

Chapter 2 outlines some fundamental research outcomes in the major areas of electricity price prediction, SLO guaranteed dynamic resource demand prediction as well as energy-efficient and cost-effective resource allocation for geographically distributed cloud data centers.

The next chapter is about the theory background for the proposed resource management system. It initially describes the cloud computing and its backend technology. The objective is to integrate the theory findings across ML techniques for prediction perception and comprehensive energy and cost aware resource allocation approaches.

In Chapter 4, the implementation of system for electricity price prediction is presented. Then, the evaluation results of prediction system are illustrated by the real- world electricity price traces.

In Chapter 5, the implementation of SLO guaranteed resource demand prediction system is presented. Then, the evaluation results of prediction systems are illustrated by the real -world data center workload traces.

Chapter 6 presents the energy-efficient and cost -effective resource allocation strategies. Then analysis of the proposed allocation algorithms and evaluation results in terms of energy-efficient and cost-effective manner are discussed.

7.2 Scope and Limitations

This thesis illustrates predicting electricity prices for GDCs in US multi-regional markets.

This research area covers the resource prediction for High Performance Computing, and Grid Workloads. Although there is no historical workload for Cloud Computing environments publicly for the model development, the current research trend to build the prediction model is well suited for resource prediction of any infrastructure environment. There are many resource parameters to predict in the Cloud infrastructure such as memory, disk, bandwidth, etc. This research mainly emphasizes on CPU resource because of the rarity of the workloads containing the perfect parameters.

Another factor is the number of configuration parameters that needs to be manually specified in Decision Trees to develop resource demand prediction model.

Currently, the parameter Maximum Tree Depth and Minimum Information Gain only adjusted in Chapter 5.

SLO considered in this thesis is based on the criteria to meet the requested CPU cores of the submitted jobs.

This research implements an energy efficiency solution in the soft computing area, but it excludes the type of energy sources, as well as the energy consumption and cost of other IT equipments other than the CPU of servers in the data centers.

This research implements and evaluates an energy-efficient and cost-effective resource management framework for three GDCs (each configured with 45 heterogeneous physical servers) in US multi-regional electricity markets. The proposed resource allocation algorithms promote energy efficiency and minimize the energy cost of Cloud Data Centers based on a thorough analysis of the evaluation results. These findings indicate that the research was successful in meeting its objectives.

7.3 Conclusion

In cloud environments, energy-efficient and cost-effective resource management strategy is needed to achieve users' satisfaction as well as to maximize cloud service providers' profit. Resource management is becoming popularity as it is paying attention for managing resources of GDCs to maximize revenue of the cloud providers. To provide for managing the cloud resources in GDCs cost efficiently through exploring electricity price diversities and energy-efficiently while satisfying certain level of SLO, electricity price prediction for GDCs to incorporate the price diversity, SLO guaranteed resource demand prediction for handling dynamic resource while meeting SLO and resource allocation in energy-efficient and cost-effective manner are required to develop.

In the thesis, energy-efficient and cost-effective resource management framework is presented which firstly predicts the electricity prices of GDCs, secondly predicts the resource demand and ensures SLO. Then, it allocates the incoming requests to the GDCs resources in energy-efficient and cost-effective manner.

Predicting electricity prices of GDCs in multi-regional electricity markets is performed using Machine Learning techniques and comparing the prediction error

of M5P, Linear Regression and Decision Table algorithms. After investigating the prediction accuracies of three ML algorithms for the experimented workload traces, resource demand prediction model with high accuracy is developed by Decision Tree algorithm with hyper-parameter optimization. It is clear that the predicted CPU demand by our proposed models is nearly identical to the actual CPU demand. According to the prediction results, the proposed predictor is under provisioning predictor. SLO analysis is performed over the prediction results system to avoid SLO violations. Analysis results show the SLOs meet by increasing a small percentage (3% in our experiment) to the prediction result which almost eliminates the under provisioning of the prediction system.

The promising energy efficiency was investigated by exploring the advantage of DVFS with cubic power model and shortest job first allocation policy after investigating three power management techniques as well as four power models in each of two resource allocation policies. It is shown that the energy consumption of resource allocation with DSJF algorithm is lower than with DFCFS algorithm by implementing in CloudSim simulator. The electricity prices of GDCs in multi-regional markets are predicted in order to deliver the workloads to the data center with cheaper electricity price. The evaluation for proposed EECERA algorithm is compared with EERA algorithm using three real workload traces of virtualized environment for energy cost minimization.

7.4 Further Research Direction

Despite the contributions of this thesis to energy-efficient and cost-effective resource management in GDCs, there are many open research challenges that need to be addressed in order to further advance the research area: VM migration across GDCs transmission networks, routing workloads to GDCs renewable energy availability, and minimizing GDCs brown energy consumption.

There are still some shortcomings in this research work. First, we just take saving energy consumption and energy-related cost as our objective. Next, there are other targets, such as QoS guaranteeing and network optimization. Second, to evaluate the proposed resource allocation methods, we only use the CloudSim simulator, and we do not test them in a real large-scale cloud platform. In the future,

the proposed allocation algorithms can be evaluated by implementing them in large scale cloud infrastructure platform again.

AUTHOR'S PUBLICATIONS

[P1] Moh Moh Than, Thandar Thein, "*Electricity Price Prediction for Geographically Distributed Data Centers in Multi-Region Electricity Markets*," in Proceedings of the IEEE 3rd International Conference on Computer and Communication Systems (ICCCS), Japan on 27th-30th April, 2018, pp. 89-93

[P2] Moh Moh Than, Thandar Thein, "*Cloud Data Center Resource Demand Prediction Model Development on Apache Spark*," in Proceedings of the 17th International Conference on Computer Applications (ICCA), Myanmar on 27th-28th February, 2019, pp. 30-36

[P3] Moh Moh Than, Thandar Thein, "*Energy-Saving Resource Allocation in Cloud Data Centers*," in Proceedings of the IEEE 18th International Conference on Computer Applications (ICCA), Myanmar on 27th-28th February, 2020, pp.45-51

[P4] Moh Moh Than, "*Resource Management for Minimizing Energy and Cost of Geo-Distributed Data Centers*," Walailak Journal of Science and Technology, Volume 18, Number 13, 1 July 2021 (SCImago index –Q3)

BIBLIOGRAPHY

[1] M. A. Adnan, R. Sugihara and R. K. Gupta, "Energy Efficient Geographical Load Balancing via Dynamic Deferral of Workload," 5th International Conference Cloud Computing, 2012.

[2] B. Ahmad, Z. Maroof, S. McClean, D. Charles, G. Parr, "Economic Impact of Energy Saving Techniques in Cloud Server," Cluster Computing, 23(2), pp. 611–621, 2020.

[3] A. Y. Alanis, "Electricity Prices Forecasting using Artificial Neural Networks," in IEEE Latin America Transactions, vol. 16, no. 1, pp. 105-111, Jan 2018.

[4] M. Alhamad, T. Dillon, E. Chang, “Conceptual SLA Framework for Cloud Computing,” Proceedings of the 2010 4th IEEE International Conference on Digital Ecosystems and Technologies (DEST), pp. 606 -610, 2010.

[5] A. Ali, L. Lu, Y. Zhu, J. Yu, “An Energy Efficient Algorithm for Virtual Machine Allocation in Cloud Data Centers”, Springer Science and Business Media Singapore, J. Wu and L. Li (Eds.): pp. 61–72, ACA 2016.

[6] M. A. Alworafi, A. Dhari, A. A. Al-Hashmi, A. B. Darem, Suresha, "An Improved SJF Scheduling Algorithm in Cloud Computing Environment," International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), pp. 208-212, 2016.

[7] Assefi, M., Behravesh, E., Liu, G., Tafti, A., P., "Big Data Machine Learning using Apache Spark MLlib," in Proceedings of IEEE International Conference on Big Data, Boston, MA, USA, pp. 3492–3498, Dec. 11-14, 2017.

[8] Avinash K.P., Dr. Minavathi., “Effective Use of Resources for Cost Minimization on Big Data in Distributed Data Centers,” International Journal Of Engineering Research & Technology (IJERT) NCRTS –Volume 3 – Issue 27, 2015.

[9] Bahwaireth, K., Tawalbeh, L., Benkhelifa, E. et al. “Experimental Comparison of Simulation Tools for Efficient Cloud and Mobile Cloud Computing Applications,” EURASIP J. on Info. Security 2016, 15 (2016).

[10] L. A. Barroso, “The Price of Performance,” In Magazine of Queue - Multiprocessors, Volume 3, Issue 7, New York, NY, USA, pp. 48-53, September 2005.

[11] M. Belkhouraf, A. Kartit, H. Ouahmane, H. K. Idrissi, Z. Kartit and M. E. Marraki, "A Secured Load Balancing Architecture for Cloud Computing Based on Multiple Clusters," IEEE, 2015.

[12] A. Beloglazov, J. Abawajy, R. Buyya, “Energy-aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing,” Future generation computer systems, 28(5), pp. 755-768, 2012.

[13] N. Bobroff, A. Kochut, and K. Beaty, “Dynamic Placement of Virtual Machines for Managing SLA Violations,” in Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM), pp.119-128, 2007.

[14] L. Breiman, “Random Forests,” Journal of Machine Learning, Volume 45, Issue 1, pp. 5-32, 2001.

[15] L. Breiman, “Bagging Predictors,” Journal of Machine Learning, Volume 24, Issue 2, pp. 123-140, 1996.

[16] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as The 5th Utility”, Journal of Future Generation Computer Systems, vol. 25, issue 6, pp. 599–616, 2009.

[17] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose and R. Buyya, “Cloudsim: a Toolkit for Modeling and Simulaion of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,” Software: Practice and Experience, vol. 41, no. 1, pp.23-50, 2011.

[18] M. Camilleri, F. Neri, "Parameter Optimization in Decision Tree Learning by using Simple Genetic Algorithms", WSEAS TRANSACTIONS on COMPUTERS, Vol. 13, pp. 582–591, 2014.

[19] J. Chen, Tang, K., Li, Bilal, Yu, Z.K.S., Weng, C., Li, K., "A Parallel Random Forest Algorithm for Big Data in Spark Cloud Computing Environment," IEEE Transactions on Parallel Distributed System, vol. 28, issue 4, pp. 919–933, 2017.

[20] “Data Center Map”, [Online]. Available at: “http://www. datacentermap.com/”

[21] “Decision Table” [Online]. Available at “https://www.geeksforgeeks.org/software-engineering-decision-table/”

[22] “Decision Tree” [Online]. Available at “https://spark.apache.org/docs/2.2.0/mllib-decision-tree.html”

[23] "Decision Trees - RDD-based API", [Online]. Available at “https://spark.apache.org/docs/2.2.0/mllib-decision-tree.html”

[24] V. C. Emeakaroha, M. A.S. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. F. De Rose, “Towards Autonomic Detection of SLA Violations in Cloud Infrastructures,” Journal of Future Generation Computer Systems, Volume 28, Issue 7, pp. 1017-1029, July 2012.

[25] “ENERGY ONLINE”, [Online]. Available at “http://www.energyonline. com/Data/Default.aspx”

[26] U.S. Environmental Protection Agency, “Report to Congress on Server and Data Center Energy Efficiency: Public Law 109-431,” Lawrence Berkeley National Laboratory, CA, USA, 2008.

[27] X. Fan, W. D. Weber, L. A. Barroso, “Power Provisioning for A Warehouse-Sized Computer”, Proceedings of 34th Annual Int. Sympo. Comput. Archit. (ISCA), ACM New York, USA, pp. 13–23, 2007.

[28] Federal Energy Regulatory Commission, “U.S Electric Power Markets,” [Online]. Available at “http://www.ferc.gov/market-oversight/mktelectric/overview.asp”

[29] M. Feurer, T. Springenberg, and F. Hutter, “Initializing Bayesian Hyper-Parameter Optimization via Meta-learning,” in Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp. 1128-1135, Jan. 2015.

[30] M. Feurer and F. Hutter, “Hyper-parameter optimization,” in: Hutter F., Kotthoff L., Vanschoren J. (Eds.), Automated Machine Learning, The Springer Series on Challenges in Machine Learning, Springer, Cham, pp. 3-33, 2019.

[31] J. C. R Filho and C. M. Affonsoe R. C. L. Oliveirae, "Pricing Analysis in the Brazilian Energy Market: A Decision Tree Approach,” presented at the Conference IEEE Bucharst Power Tech, Bucharest, Romania, 2009.

[32] J. Fu, J. Sun, K. Wang, “SPARK—A Big Data Processing Platform for Machine Learning,” in Proceedings of IEEE International Conference on

Industrial Informatics- Computing Technology, Intelligent Technology, Industrial Information Integration, Wuhan, China, pp. 48-51, Dec. 3-4, 2016.

[33] R.C. Garcia, J. Contreras, M. Van Akkeren, and J.B.C. Garcia. “A GARCH Forecasting Model to Predict Day-Ahead Electricity Prices,” IEEE Transactions on Power Systems, 20(2), pp. 867–874, 2005.

[34] K. C. Gouda, T. V. Radhika, and M. Akshatha, “Priority based Resource Allocation Model for Cloud Computing,” International Journal of Science, Engineering and Technology Research (IJSETR), Volume 2, Issue 1, pp. 215-219, January 2013.

[35] Lin Gu, "Cost Efficient Resource Management for Geo-distributed Data Centers," Ph.D Thesis Book, Department of Computer and Information Systems, The University of Aizu, February 2015.

[36] J. Han and M. Kamber, “Data Mining: Concepts and Techniques,” Second Edition, Elsevier Inc., ISBN 13: 978-1-55860-901-3, 2006.

[37] S. Harizopoulos, M. A. Shah, J. Meza, and P. Ranganathan, “Energy Efficiency: The New Holy Grail of Data Management Systems Research,” Proceedings of the 4th Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar, CA, USA, January 2009.

[38] T. K. Ho, “The Random Subspace Method for Constructing Decision Forests,” Journal of Pattern Analysis and Machine Intelligence, IEEE Transactions, Volume 20, Issue 8, pp. 832-844, 1998.

[39] Huurman, C., Ravazzolo, F., & Zhou, C. “The power of Weather,” Computational Statistics and Data Analysis, 56(11), pp. 3793–3807, 2012.

[40] Hwang, K., Fox, G., and Dongarra, J, “Distributed and Cloud Computing: From Parallel Processing to the Internet of Things,” Morgan Kaufmann, 2012.

[41] Janczura, J., Trueck, S., Weron, R., & Wolff, R. “Identifying Spikes and Seasonal Components in Electricity Spot Price Data: A Guide to Robust Modeling,” Energy Economics, 38, pp. 96–110. Janczura, J., & Weron, R. 2013.

[42] S. R. Jena, V. Vijayaraja, A. K. Sahu, "Performance Evaluation of Energy Efficient Power Models for Digital Cloud," Indian Journal of Science and Technology, 9(48), pp. 1-7, 2016.

[43] S. Y. Jing, S. Ali, K. She and Y. Zhong, “State-of-the-art Research Study for Green Cloud Computing,” The Journal of Supercomputing ,Volume 65, Issue 1, pp. 445-468, 2013.

[44] Jonsson, T., Pinson, P., Nielsen, H. A., Madsen, H., & Nielsen, T. S. “Forecasting Electricity Spot Prices Accounting for Wind Power Predictions,” IEEE Transactions on Sustainable Energy, 4 (1), pp. 210–218, 2013.

[45] A. V. Karthick, E. Ramaraj and R. G. Subramanian, "An Efficient Multi Queue Job Scheduling for Cloud Computing," World Congress on Computing and Communication Technologies, Trichirappalli, pp. 164-166, 2014.

[46] M. G. Khajeh, A. Maleki, M. A. Rosen, M. H. Ahmadi, "Electricity Price Forecasting using Neural Networks with an Improved Iterative Training Algorithm," International Journal of Ambient Energy, pp. 147-158, 2016.

[47] A. Khan, X. Yan, S. Tao, and N. Anerousis, "Workload Characterization and Prediction in the Cloud: A Multiple Time Series Approach," in Proceedings of the IEEE Network Operations and Management Symposium, Maui, HI, pp. 1287–1294, April 2012.

[48] A. Khosravi, L. H. Andrew, R. Buyya. “Dynamic VM Placement Method for Minimizing Energy and Carbon Cost in Geographically Distributed Cloud Data Centers,” IEEE Transactions on Sustainable Computing (TSUSC), 2 (2), IEEE Computer Society Press, USA, pp. 183-196, 2017.

[49] Kliazovich, D. Bouvry, P. Audzevich and Y. Khan, “Greencloud: A packet-Level Simulator of Energy-aware Cloud Computing Data Centers,” IEEE Int. Conf. Global Telecommunications, pp.1–5, 2010.

[50] P. Kumar, A. K. Rai, "An Overview and Survey of Various Cloud Simulation Tools," Journal of Global Research in Computer Science, Volume 5, No. 1, January 2014.

[51] D. Laganà, C. Mastroianni, M. Meo and D. Renga, "Reducing the Operational Cost of Cloud Data Centers through Renewable Energy," Algorithms 2018, 11 (10), pp. 145.

[52] J. Lagarto, De Sousa, J., Martins, A., and Ferrão, P. “Price Forecasting in the Day-Ahead Iberian Electricity Market using A Conjectural Variations

ARIMA Model," IEEE Conference Proceedings — EEM12, art. no. 6254734, 2012.

[53] L. Lee, K. Liu, H. Huang, C. Tseng, "A Dynamic Resource Management with Energy Saving Mechanism for Supporting Cloud Computing", Int. J. Grid Distrib. Comput. 6 (1), pp. 67–76, 2013.

[54] Y. Liet al., "Operating Cost Reduction for Distributed Data Centers," in CCGRID '13, pp. 589–596, May 2013.

[55] W. Y. Lin, G. Y. Lin, and H. Y. Wei, "Dynamic Auction Mechanism for Cloud Resource Allocation," Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 591-592, May 2010.

[56] "M5P" [Online]. Available at "http://old.opentox.org/dev/documentation/components/m5p"

[57] A. T. Makaratzis, K. M. Giannoutakis, D. Tzovaras, "Energy Modeling in Cloud Simulation Frameworks", J. Futu. Gen. Comput. Syst., 79, pp. 715–725, 2017.

[58] S. Malik, P. Saini and S. Rani, "Energy Efficient Resource Allocation for Heterogeneous Workload in Cloud Computing," S.C. Satapathy et al. (eds.), Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Advances in Intelligent Systems and Computing, Springer Nature Singapore Pte Ltd. pp. 89-97, 2017.

[59] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft)," Special Publication 800-145 (Draft), National Institute of Standards and Technology, Gaithersburg, Maryland, September 2011.

[60] A. Mohsenian-Rad and A. Leon-Garcia, "Energy-information Transmission Tradeoff in Green Cloud Computing," Proceedings of IEEE GLOBECOM, pp. 1-6, 2010.

[61] Moreno-Vozmediano, R., Montero, R.S., Huedo, Llorente, I.M. "Efficient Resource Provisioning for Elastic Cloud Services based on Machine Learning Techniques," Journal of Cloud Computing, 8(5), 2019.

[62] NRDC and Anthesis, "Scaling Up Energy Efficiency Across The Datacenter Industry: Evaluating Key Drivers and Barriers," Natural Resources Defense Council, Tech. Rep., 2014.

[63] "Parallel Workloads Archive", [Online]. Available at "http://www.cs.huji.ac.il/labs/parallel/workload/"

[64] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs, “Cutting the Electric Bill for Internet-scale Systems,” in Proceedings of SIGCOMM, pp. 123–134, ACM, 2009.

[65] L. Rao, X. Liu, L. Xie and W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment," Proceedings IEEE INFOCOM, San Diego, CA, pp. 1-9, 2010.

[66] S. Rawas, A. Zekri. “Location-Aware Energy-Efficient Workload Allocation in Geo Distributed Cloud Environment,” Journal of Computer Science, 14(3), pp. 334-350, 2018.

[67] S. Rawas, A. Zekri, A. E. Zaart, "Power and Cost-aware Virtual Machine Placement in Geo-distributed Data Centers," in Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018), pp. 112-123, 2018.

[68] A. Rayan, Y. Nah, “Resource Prediction for Big Data Processing in a Cloud Data Center: A Machine Learning Approach,” IEIE Transactions on Smart Processing and Computing, vol. 7, no. 6, 2018.

[69] T. Renugadevi, K. Geetha, N. Prabaharan and P. Siano, "Carbon-Efficient Virtual Machine Placement Based on Dynamic Voltage Frequency Scaling in Geo-Distributed Cloud Data Centers," Applied Science Journal, 10 (8), 2020.

[70] Rivera, Wilson. Sustainable Cloud and Energy Services. "Short-Term Prediction Model to Maximize Renewable Energy Usage in Cloud Data Centers," 10.1007/978-3-319-62238-5(Chapter 8), pp. 203–218, 2018.

[71] N. Roy, A. Dubey, and A. Gokhale, "Efficient Auto Scaling in the Cloud using Predictive Models for Workload Forecasting," in Proceedings of the IEEE 4th International Conference on Cloud Computing, Washington, DC, pp. 500-507, July 2011.

[72] Shafie-Khah, M., Moghaddam, M. P., & Sheikh-El-Eslami, M. K. “Price Forecasting of Day-Ahead Electricity Markets using A Hybrid Forecast Method,” Energy Conversion and Management, 52(5), pp. 2165–2169, 2011.

[73] U. Sinha, M. Shekhar, "Comparison of Various Cloud Simulation tools available in Cloud Computing," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 3, March 2015.

[74] A. Smola and S.V.N. Vishwanathan ,“Introduction to Machine Learning,” Departments of Statistics and Computer Science, Purdue University, Cambridge University Press 2008.

[75] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, “Virtual Infrastructure Management in Private and Hybrid Clouds,” IEEE Internet Computing, Vol. 13, No. 5, pp.14_22, September, 2009.

[76] “Spark MLib”, [Online]. Available at “https:// spark.apache.org/mllib/”

[77] U. Ugurlu, I. Oksuz and O. Tas, "Electricity Price Forecasting using Recurrent Neural Networks," Energies, 11(5), 2018.

[78] M. J. Usman, A. Samad, H. Chizari, & A. Aliyu, “Energy-Efficient Virtual Machine Allocation Technique using Interior Search Algorithm for Cloud Datacenter,” 6th IEEE ICT International Student Project Conference (ICT-ISPC), pp. 1-4, 2017.

[79] M. Verma, G.R. Gangadharan, V. Ravi, V. Inamdar, L. Ramachandran, R. N. Calheiros, R. Buyya, “Dynamic Resource Demand Prediction and Allocation in Multi-Tenant Service Clouds,” 2016.

[80] X. Wang, Z. Sheng, S. Yang, and V. C. M. Leung, "Tag-assisted Social-Aware Opportunistic Device-to-device Sharing for Traffic Offloading in Mobile Social Networks," IEEE Wireless Communications, vol. 23, no. 4, pp. 60–67, 2016.

[81] A. Weiss, “Computing in the Clouds,” In Magazine netWorker - Cloud Computing: PC functions move onto the web, Volume 11, Issue 4, pp. 16-25, 2007.

[82] “Welcome to Apache Hadoop!”, [Online]. Available at “http://hadoop.apache.org/”

[83] I. H. Witten, E. Frank, and M. A. Hall, “Data Mining Practical Machine Learning Tools and Techniques,” ISBN 978-0-12-374856-0, Third Edition, Morgan Kaufmann.

[84] Ø. Wormstrand, “Electricity Price Prediction, A Comparison of Machine Learning Algorithms,” M.S. thesis, Østfold University College, Halden, Norway, 2011.

[85] J. Wu, X-Y. Chen, H. Zhang, L-D. Xiong, H. Lei, S-H. Deng, "Hyper-parameter Optimization for Machine Learning Models based on Bayesian Optimization," Journal of Electronic Science and Technology, vol. 17, issue 1, pp. 26-40, 2019.

[86] H. Xu, C. Feng, Baochun, "Temperature Aware Workload Management in Geo-distributed Datacenters," 10th International Conference on Autonomic Computing (ICAC '13), pp. 303-314, 2013.

[87] H. Yuan, J. Bi and M. Zhou, "Spatial Task Scheduling for Cost Minimization in Distributed Green Cloud Data Centers," in IEEE Transactions on Automation Science and Engineering, vol. 16, no. 2, pp. 729-740, April 2019.

[88] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster Computing with Working Sets," in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, 2010.

[89] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing," in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, pp. 2–2, 2012.

[90] Zhu. Timothy, Kozuch, Michael A., Harchol-Balter. Mor, "WorkloadCompactor: Reducing Datacenter Cost While Providing Tail Latency SLO Guarantees," Proceedings of ACM Press the 2017 Symposium on Cloud Computing, pp. 598–610, Santa Clara, California, 2017.

LISTS OF ACRONYMS

API	Application Program Interface
AR	Auto Regressive
ARIMA	Auto Regressive Integrated Moving Average
BDAS	Berkeley Data Analysis Stack
CPU	Central Processing Unit
DVFS	Dynamic Voltage Frequency Scaling
GARCH	Generalized Autoregressive Conditional Heterokedasticity
GDC	Geographically Distributed Data Centers
GRU	Gated Recurrent Unit
HDFS	Hadoop Distributed File System
HTTP	Hyper Text Transfer Protocol
IaaS	Infrastructure as a Service
IT	Information Technology
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MapR-FS	MapR File System
ML	Machine Learning
MlLib	Apache Spark's Scalable Machine Learning Library
MSE	Mean Squared Error
NPA	Non Power Aware
PA	Power Aware
PaaS	Platform as a Service
PC	Power Consumption
PM	Physical Machine
QoS	Quality of Service

RAM	Random Access Memory
RBF	Radial Basis Function
RDDs	Resilient Distributed Datasets
REPTree	Reduced Error Pruning Tree
RFR	Random Forest Regression
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SaaS	Software as a Service
SLA	Service Level Agreement
SLO	Service Level Objective
SQL	Structured Query Language
SVM	Support Vector Machines
VM	Virtual Machine
YARN	Yet Another Resource Negotiator