Energy Efficient Extended Pool Management Scheme in Cloud Data Centers

Adnan Asif Chowdhury

ID: 2012-3-60-038

Md. Shahidul Hoque

ID: 2013-1-60-007

Nabil Shawkat

ID: 2013-1-60-023

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering
East West University
Dhaka-1212, Bangladesh

April, 2017

Declaration

We, hereby, declare that the work presented in this thesis is the outcome of the research performed by us under the supervision of Amit Kumar Das , Lecturer, Department of Computer Science and Engineering, East West University. We also declare that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Signature

. . . .

(Adnan Asif Chowdhury)
(ID: 2012-3-60-038)

Signature

. . . .

(Md. Shahidul Hoque)
(ID: 2013-1-60-007)

Signature

. . . .

(Nabil Shawkat)
(ID: 2013-1-60-023)

Letter of Acceptance

This Thesis Report entitled *"Energy Efficient Extended Pool Management Scheme in Cloud Data Centers"* submitted by Adnan Asif Chowdhury (ID: 2012-3-60-038), Md. Shahidul Hoque (ID: 2013-1-60-007) and Nabil Shawkat (ID: 2013-1-60-023), to the Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh is accepted by the department in partial fulfillment of requirements for the Award of the Degree of Bachelor of Science and Engineering on April, 2017.

Supervisor

. . . .
(Amit Kumar Das)
Lecturer, Department of Computer Science and Engineering
East West University, Dhaka, Bangladesh.

Chairperson

. . . .
(Dr. Ahmed Wasif Reza)
Chairperson (Acting) and Associate Professor,
Department of CSE, East West University.

Abstract

Cloud computing is a major shift from the conventional way of businesses considering *IT* assets. Cloud computing empowers organizations to expand a process asset, such as virtual machine, storage or an application, as a utility. As a result, the demand for energy is increasing day by day. In this report, we focus on improving an analytical performance model for improving the resource allocation process to reduce power consumption. In particular, our report proposes a resource allocation algorithm, namely *Triple-E*, for energy-efficient management of clouds. We are using the *CloudSim* toolkit to validate our approach by leading a set of accurate performance evaluation study. The results show that *Triple-E* has enormous potential as it offers critical execution gains as regards to response time under dynamic workload situations.

Acknowledgments

As it is true for everyone, we have also arrived at this point of achieving a goal in our life through various interactions with and help from other people. However, written words are often elusive and harbor diverse interpretations even in one's mother language. Therefore, we would not like to make efforts to find best words to express our thankfulness other than simply listing those people who have contributed to this thesis itself in an essential way. This work was carried out in the Department of Computer Science and Engineering at East West University, Bangladesh.

First of all, we would like to express our deepest gratitude to Allah for blessings on us. Next, our special thanks go to our supervisor,"Amit Kumar Das", who gave us this opportunity, initiated us into the field of **"Energy Efficient Extended Pool Management Scheme in Cloud Data Centers"**, and without whom this work would not have been possible. His encouragements, visionaries and thoughtful comments and suggestions, unforgettable support at every stage of our BS.c study were simply appreciating and essential. His guidance helped us in all the time of research and writing of this thesis report. The door to his assistance was always open whenever we had a trouble or had a question about our research or writing. His ability to muddle us enough to finally answer our own question correctly is something valuable what we have learned and we would try to emulate, if ever we get the opportunity.

Our deepest gratitude to all our faculty members who have supported us in East West University. Their devotion toward our improvement helped us to increase our knowledge

of the individual subjects and enabled us to complete our studies in time. Finally, we must express our very profound gratitude to our parents for providing us with unfailing support and continuous encouragement throughout our years of study and through the process of researching and writing this thesis report.

There are numerous other people too who have shown us their constant support and friendship in various ways, directly or indirectly related to our academic life. We will remember them in our heart and hope to find a more appropriate place to acknowledge them in the future.

Adnan Asif Chowdhury
April, 2017

Md. Shahidul Hoque
April, 2017

Nabil Shawkat
April, 2017

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Cloud Computing

Cloud computing is the act of putting away frequently utilized computer information on servers that can be accessed through the Internet. It gives shared *PC* handling assets and information to *PCs* and different devices on request. For accessing to a shared pool which can be easily supplied and released with negligible attempts is a model for common empowering. Cloud computing and storage solutions provide clients and undertakings with different abilities to store and process their information in either exclusive or outsider data centers that might be situated a long way from the user ranging in separation from over a city to over the world [1]. Cloud computing depends on sharing of assets to accomplish intelligence and economy of scale, like a utility over a power arrange. In cloud computing, response time and service availability are two major factors concerning user experience. Relocating super-tasks to cloud assets or growing new super-tasks in a cloud-local environment.

1.1.1 Objectives of Cloud Computing

The objective of cloud computing is to permit clients to take benefit from these advancements, without the requirement for profound learning about or ability with every one of them. The cloud intends to cut expenses, and helps the clients concentrate on their center business as opposed to being blocked by *IT* snags. The principle empowering innovation for cloud computing is virtualization. Virtualization programming isolates a physical registering gadget into at least one *"virtual"* gadgets, each of which can be effortlessly utilized and figured out how to perform processing undertakings. With working system-level virtualization basically making a versatile arrangement of numerous autonomous figuring gadgets, sit still processing assets can be designated and utilized all the more productively.

1.2 Green IT

Cloud computing is using Green Information Technology in short *Green IT* to improve energy-efficiency and sustainability goals [2]. The area of green computing is very much needed because of limited energy resources and the immense amount of increasing demand for more technological power [3]. *Green IT* means to minimize the adverse effect of *IT* operations on the earth by planning, fabricating, working and discarding PCs and PC-related items in an ecologically well-disposed manner. The intentions behind *Green IT* rehearses incorporate lessening the utilization of dangerous materials, amplifying vitality effectiveness amid the object's lifetime and advancing the biodegradability of unused and obsolete items. *Green IT* is not quite recently centered on diminishing the effect of the *ICT* business. Its main focus is to make cloud computing more profitable by reducing energy wastage and also keeping the service level at its best [4].

1.2.1 Importance of Green IT

Green computing is the act of utilizing figuring and *IT* assets capably. As an individual it is our prime obligation to secure the earth and spare vitality cost in today's undeniably computing necessities. Green computing or *Green IT*, is the examination and routine of earth economical figuring or *IT*. *Green IT* can be come to through decrease of energy utilization and waste. Vitality administration and discharges following programming are accessible. What the *IT* purchases - from *PC* gear to paper - specifically impacts how *Green IT* is and how green its providers are. In the event that an *IT* association just buys advances with Energy Star, Electronic Product Environmental Assessment Tool (*EPEAT*), and other energy effectiveness appraisals, it can altogether decrease its energy utilization and nursery gas impression, and it will help drive innovation makers to create items that procure vitality proficiency evaluations. Toward the finish of the chain, a *Green IT* work needs a waste administration program.

1.2.2 Advantages of Green IT

- Diminished imperativeness use from green computing methodologies changes over into lower carbon dioxide surges, originating from a reducing in the oil subsidiary used as a piece of vitality plants and transportation.

- Directing resources infers less essentialness is required to convey, use, and dispose of things.

- Saving essentialness and resources saves money.

- Green computing even joins changing government way to deal with stimulate reusing and cutting down essentialness use by individuals and associations.

- Reduce the risk existing in the convenient workstations, for instance, creation known to achieve infection, nerve hurt and safe reactions in individuals.

1.2.3 Disadvantages of Green IT

- Green figuring could really be very exorbitant.

- A few *PCs* that are green might be extensively underpowered.

- Fast innovation change.

1.3 Contributions

Infrastructure as a Service (*IaaS*) is used to reallocate the resource in cloud computing [5]. Because *IaaS* is a method of cloud computing that provides virtualized computing resources and work on availability analysis of largescale cloud computing over the internet. An *IaaS* Cloud, for example, *Amazon EC2* [6] and *IBM* Smart Business Cloud [7], conveys, on-request, working framework cases provisioning computational assets as virtual machines sent in the cloud supplier's server farm. Huge cloud specialist co-ops, for example, *IBM* give Service Level Agreements (*SLAs*) managing the accessibility of the cloud benefit. Before conferring an *SLA* to the clients of a cloud, the specialist co-op necessities to do accessibility investigation of the framework on which the cloud service is facilitated. In [8], authors have shown how stochastic analytic models could be utilized for cloud service availability analysis for efficiency. In the beginning, they developed a one-level monolithic model and then use an interacting sub-models approach. But a monolithic model may suffer from intractability and poor scalability due to a large number of parameters. In [9], the authors developed and evaluated tractable functional sub-models and their interaction model and solve them iteratively. But the interaction among the sub model is not correctly performed to make it more efficient. Our proposed model grasps different perspectives and gives understanding into their cooperation's of today's cloud centers. The principle components of our expository model can be chronicled as:

- In this report, we are utilizing Multiple Linear Regression (*MLR*) for distributing computing assets to user applications in a way that enhances and improves the productivity of cloud computing.
- We have developed an energy-aware process using *MLR* that will self-manage changes in the state of resource allocation effectively and efficiently to fulfill benefit commitments and accomplish energy effectiveness.

- We are exploring various workloads of different sorts of cloud applications and create algorithms for energy-efficient Virtual Machine (*VM*) resource allocations.

Cloud computing is a vast area of doing research. We have discussed about cloud computing, about the importance of cloud computing. Again, *Green IT* also is an important factor. We have over-viewed about *Green IT*, about the importance of *Green IT*, about the advantages and disadvantages of *Green IT*. Lastly, we have represented our contributions while working on this paper.

1.4 Organization of the Report

We have structured our rest of the Thesis works as following: In Chapter 2, we survey related work in pool management performance analysis; In Chapter 3, model and assumptions are discussed meticulously; Chapter 4 introduces our proposed model with Markov Chain and their interactions as well as presents the numerical results obtained through Multiple Linear Regression (*MLR*) from our proposed model; Chapter 5 presents the result of performance evaluation and simulation; In Chapter 6, the conclusion and future research directions are highlighted ; Lastly, proper references of our thesis works, a list of acronyms and notations as well as our list of publications has been illustrated respectively.

Chapter 2

Related Work

2.1 Introduction

At business as well as technological level, cloud computing can be considered as a new topic. Such computing can be considered as a distributed computing atmosphere as well as a adaptable computing atmosphere. Cloud computing *IT* comprises of a gathering of interconnected and virtualized *PCs* that are powerfully provisioned and exhibited as at least one brought together figuring assets to buyers. Buyers can get from anywhere and anytime via internet the various level of services such as, *SaaS*, *PaaS*, *IaaS* which are delivered by cloud computing. So, cloud computing has made our life easy. With the help of such thing we can send data to cloud data centers from anywhere of the world through devices. But much energy is consumed while allocating the resources. Different researches has been done to reduce the consumption of energy for such issue. Our main motivation is to establish such a model which will bring energy efficiency in cloud data centers while allocating resources and which will be better then the models that has been established. So, we have gone through some research papers to know about their models, works and techniques so that we can make a better one.

2.2 Related Work

Various authors tried to represent their thoughts by presenting their own techniques to establish a model to have energy efficiency by reducing power consumption in cloud data centers which are as follows:

2.2.1 Exploitation of energy

A significant issue for cloud access provider is energy exploitation by cause of commercial along with universal interest. For that reason, decreasing the energy consumption of data centers, cloud service organizations are searching for new ideas. So that, they can implement more efficient techniques for their data centers. Service providers differ from the way of giving classical services which creates various opportunities along with new design challenges when they are trying to implement energy-aware resource allocation techniques for cloud data centers. Various research has been done on reducing the consumption of energy of data centers in cloud computing [10] [11] [12].

2.2.2 Challenges in resource allocation

In [13], authors indicated the resource allocation challenges and some possible solving ways for reducing energy consumption in cloud centers where virtualization innovation to save energy was the focusing point on power management technique. Reliability and energy efficiency are two significant challenges in cloud computing. In [13], the authors have given much preference on energy consumption efficiency in cloud data centers since cloud data centers are instances of such far reaching server ranches whose offered organizations are expanding perpetually lifted universality, especially with the starting late observed growing reliance of Personal Digital Assistants ($PDAs$) on cloud organizations. Then introduced the cloud paradigm that they observed. Then explained the new challenges and opportunities that arise when trying to save energy in cloud centers. After that they tried to describe the most popular techniques and solutions that can be adopted by cloud data centers to save energy.

2.2.3 Reliability and Energy efficiency

Reliability also, energy efficiency are two major difficulties in cloud computing frameworks that need watchful consideration and examination since with the prevalence of cloud computing, it moves toward becoming urgent to give on-request benefits progressively as per the client's necessities and for such reason, in [14], the authors tried to explain the existing techniques for safety and energy efficiency and then identified the research gaps to combining these two metrics for resource provisioning in cloud computing environments. Also outlined about the current strategies for giving resource immovable quality and restricting energy usage in cloud structures, independently. They also depicted investigation challenges and gaps recognized from the outlines.

2.2.4 Use of assets

Again, power awareness for decreasing energy consumption in cloud data center, server control exchanging and assets controlling has been used in [15]. Server power switching is a process where idle servers are turned off to reduce energy consumption. Assets controlling is a process where energy consumption need to minimize by hardware or software level to meet execution necessities for cloud computing [15]. Utility oriented *IT* services are offered by cloud computing to users globally. In pay-as-you-go model, business domains are hosted in cloud nowadays [16][17]. For that reason, the huge amount of energy is consumed by the cloud data centers which cause an enormous amount of carbon emission in our nature. Therefore, green cloud computing solution is necessary to improve the performance along with reducing the energy consumption in nature.

2.2.5 Visions

In [18], the authors tried to present such visions, challenges, and architectural elements for energy-efficient management of cloud computing environments. They tried to lead an outline of research in imperativeness capable enlisting in addition, propose compositional norms for energy gainful organization of Clouds, energy capable resource task methodologies and booking counts considering *QoS* cravings and power utilize properties of the devices and different open research challenges, watching out for which can bring extensive points of interest to both resource providers and consumers. In [18] the authors also have endorsed their approach by driving an execution appraisal think about using the *CloudSim* tool box. The results display that circulated registering model has gigantic potential as it offers significant cost subsidizes and shows high potential for the change of essentialness viability under component workload circumstances.

2.2.6 Fixed Number of *PMs*

In [19] the model proposed by the authors expected a fixed number of *PMs* in every pool and does not consider the effective conduct of server pools. In [8], the authors said that once a failed *PM* is repaired, it has to be returned to the original pool where it belonged before failure and if a *PM* was borrowed from other pool to replace the failed *PM*; such *PM* also has to be returned to the original pool instantly. For which much energy will be consumed if the above procedure is followed and much time is needed to complete the full process. In [9], in pool management sub model section, the authors proposed that if all the *PMs* in hot pools become idle (*i.e. no running PMs*), then all the empty *PMs* are transferred to cold pool keeping a minimum predefined number of *PMs* in the warm pool.

2.3 Conclusion

According to this process, energy savings might not be efficient. So in this report we proposed such type of approach which provides better performance in case of energy efficiency in cloud data centers. It will possess important aspects of cloud centers such as resource prediction, resource allocation, super task requests (*super task is a set of* *requests send by a user*), resource virtualization. By this, it will be easier to serve more user requests in cloud computing. As an outcome, it will generate a new dimension of using cloud computing among users.

Chapter 3

Model and Assumption

3.1 Introduction

In case of technological and commercial level, cloud computing is a top ranking topic. At present energy consumption in cloud computing is a very important factor. Various authors tried to represent their own established model to reduce the consumption of energy in cloud computing. We have also tried to represent our model and assumptions.

3.2 Some Factor Learnings

IaaS (Infrastructure as a Service) is a type of distributed computing that gives virtualized computing assets over the Internet. *IaaS* is one of the three principle classes of cloud computing administrations, nearby Software as a Service (*SaaS*) and Platform as a Service (*PaaS*). In *IaaS* cloud, when a request is prepared, a pre-constructed or tweaked plate picture is utilized to make at least one Virtual Machine (*VM*) occurrences. In computing, a virtual machine is an imitating of a *PC* framework. *VMs* depend on *PC* models and give usefulness of a physical *PC.* Their executions may include appropriate equipment, programming, or a blend. A physical *PC* (*sometimes called a physical machine or a physical box*) is an equipment based gadget. The term is secondhand to separate equipment based *PCs* from programming based *VMs.*

A Virtual Machine Monitor (*VMM*) is a product program that empowers the creation, administration, and management of *VM* and deals with the operation of a virtualized domain on top of a physical host machine. Every *PMs* and *VMs* are alike, and there exist two Virtual Machine Monitors (*VMM*) in each *PM* by which *VMs* are constructed and dispatched on the *PM.* There is a global queue which works as a queue where super-tasks are kept before assigning in *PMs* in any of the three pools. *PMs* can be busy for other super-tasks so success probabilities are used to define how many possibilities of another super-task may have to find a *PM*in the pools. Mean look up delay is a kind of measurement for finding the appropriate *PM* by the super-tasks through searching the pools. If no *PM* is free to deal with any arrived super-task, then the super-tasks in the (*global queue*) is blocked.

3.3 Assumption

In this work, we accept that pre-built models fulfill all client demands which is shown in Figure 3.1. we expect that Physical Machines (*PMs*) are ordered into three Server Pools: Hot (*i.e., with running VMs*), Warm (*i.e., turned on yet without running VM*) and Cold (*i.e., shut down*). The pool is a set of several virtualization hosts that have access to the similar virtual and physical systems, and storage resources. Pools give load balancing, high accessibility abilities, and sharing of few assets for all members. They can be designed in changing sizes and give various advantages, including execution, administration and information assurance enhancements. Also, they can be provisioned to incorporate any measure of limit and utilize any blend of physical storage room in a Storage Area Network (*SAN*). In virtual server situations, *VMs* can be put away on devoted pools, guaranteeing basic *VMs* have entry to the best possible measure of capacity. Creation of an instance in *VM* and provisioning it on hot *PM* has minimum lag correlated to other *PMs.* For provisioning, warm *PMs* needed more time to prepare, but cold *PMs* require extra time to be ready for creating instances. In this book, we allow super-tasks from user where every super-task needs single instances.

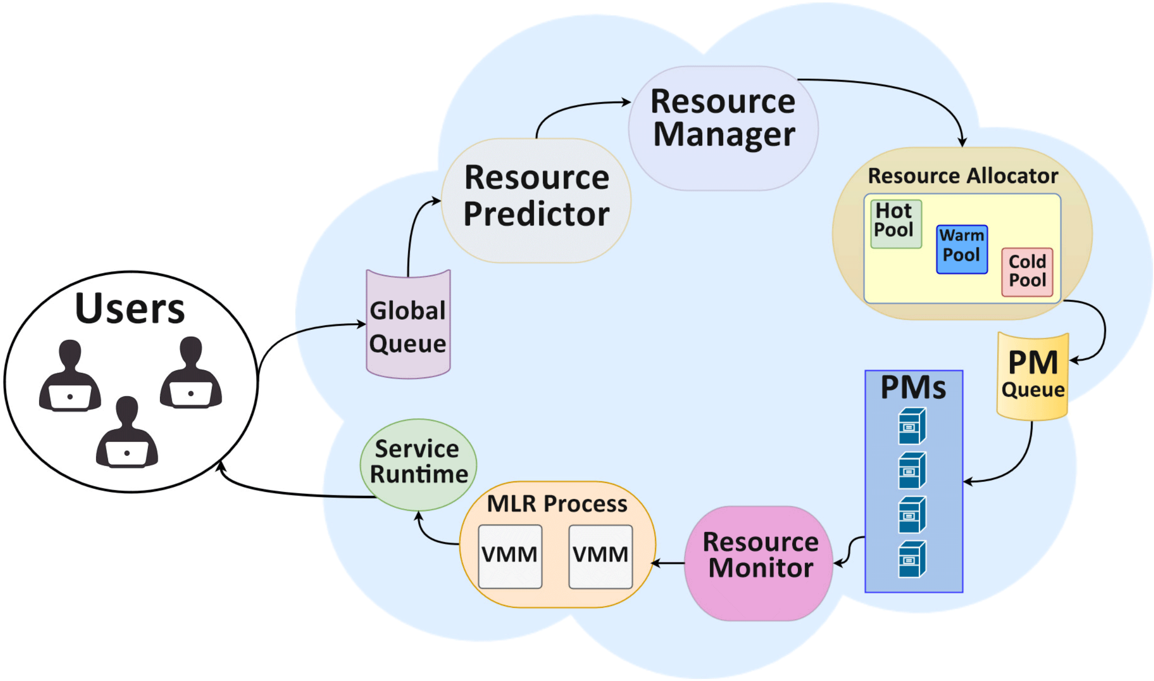

Figure 3.1: The Steps of Servicing.

3.4 Model

We exhibit a motivating example in this paper in order to clarify our work further. The scenario is that users request super-tasks (*using any media device*), which are submitted to a global finite number of the queue and then processed on a Priority based First-in, First-out basis (*PFIFO*). In *PFIFO*, the task which will come first in the queue will get high priority to accomplish the task than other tasks. In this way, the priority of the next consecutive tasks will get priority to accomplish those tasks one after another. If a user has a dedicated server in our system then they will have better chance to accept their requests earlier but without violating our terms on frequent users. The frequent users will get their service within the possible time limit. Then the super-task is then sent to Resource Predictor (*RP*) where with the help of *MLR*, prediction of data is made that how much space is initially required. Then it is sent to the Resource Manager (*RM*) so that the data are precisely managed.

When a user requests for a super-task, the request is sorted by priority of the users depending on them being premium user or not. If they are the premium user, the system will try to admit their super-task to the queue first but not hampering the Quality Of Service (*QoS*) of the system. So the new users will not be getting their services delay. The super-task need to hold till the Resource Allocator (*RA*) operates it (*first delay*) after entering into the queue. Accordingly, ram checks up the availability of required space in *PM*. *RA* shifts to the warm pool if required space is not available. In this way, *RA* shifts to the cold pool to find required space to execute the process. Ultimately, a super-task is either rejected due to deficiency or appointed to a *PM* (*second delay*). After the beginning of actual service, super-task need to wait in the *PM's* input queue when it is assigned to *PM's* (*third delay*), till the creation and uses of needed *VMs* is provided by the *MLR* Processing Module (*fourth delay*). Then the service will be sent to the user.

3.5 Conclusion

So, it is very important to give much preference to energy consumption consuming much energy in cloud computing. We have discussed some terms related to our model so that it can be easily understandable. Then we have focused on our model and assumptions for describing the model which we think is very helpful to make energy efficiency in cloud computing.

Chapter 4

Proposed Section

4.1 Introduction

In the previous chapter, we tried to represent about the structure of our model. In this chapter we will try to explain about how our model works and gives much better performance by giving almost error free result which will be very helpful for energy efficiency in cloud data centers.

4.2 Markov Chain

A Markov chain is a stochastic procedure with the Markov property. The expression "*Markov chain*" includes to the arrangement of irregular factors such a procedure travels through, with the Markov property characterizing serial reliance just between contiguous periods (*as in a chain*). It can accordingly be utilized for portraying frameworks that take after a chain of connected occasions, where what occurs next depends just on the present condition of the framework.

4.2.1 Continuous Time Markov Chain

A Continuous time Markov chain is characterized by a limited or countable state space S, a move rate grid Q with measurements equivalent to that of the state space and starting likelihood circulation characterized on the state space. For $i \neq j$, the components q_{ij} are non-negative and portray the rate of the procedure moves from state i to state j.

The components q_{ii} are picked with the end goal that each line of the move rate network totals to zero.

4.3 Proposed Model Working Function

In *Triple-E*, we are using interactive Continuous Time Markov Chain (*CTMC*) which is shown in Figure 4.1, that records the quantity of super-tasks in the comprehensive line and also the present pool on which provisioning is occurring. The global queue size is n; Thus the capacity of the system is *n+1* because deployment unit may have one or more super-tasks. Each condition of *Markov Chain* is named as (*i,j*), where i demonstrates some super-tasks in line and j shows the pool on which the basic super-task is under-provisioning. State (*0,0*) shows there is no demand under provisioning or in the line. Let π_h, π_w and π_c the achievement probabilities of finding a *PM* that can acknowledge the present errand in hot, warm and cold pool individually. Also, $1/\beta_h$, $1/\beta_w$, and $1/\beta_c$ are the mean look into deferrals to locate the proper PM. After the landing of first super-tasks, the system moves to state (*0,h*) which implies the super-task will be provisioned quickly in the hot pool. Eventually, contingent upon approaching occasion, there can be the occurrence of three conceivable steps:

- After arrival of another super-task, the system transits to state (*1,h*) with arrival rate γ_{st}.
- After accepting the super-task in hot pool by a *PM*, so that system shifts back to state (*0,0*) with rate π_h β_h.
- After failing to allocate super task in the hot pool due to deficiency of space to approve the super-task, the system examines the warm pool [*transit to state (0,w)*] with rate (1-π_h) β_h. Resource allocator tries to equip the super-task on the warm pool.

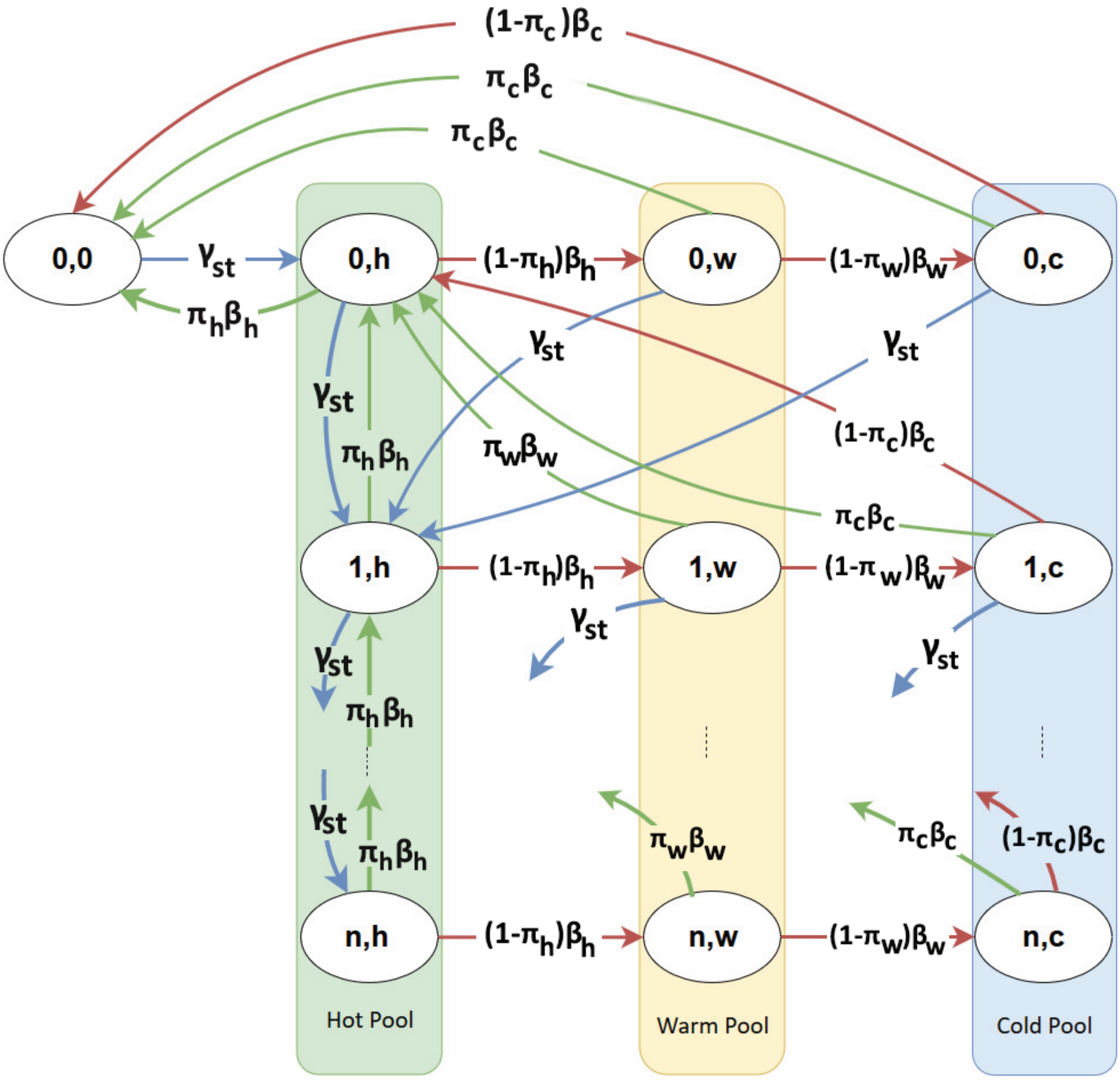

Figure 4.1: Continuous Time Markov Chain (*CTMC*)

If one of the *PMs* in the warm pool can oblige the super-tasks, the system will return to (*0,0*). Generally Resource Allocator looks at the cold pool (*transition from state (0,w) to (0,c)*). If none of the *PMs* in cold pool can arrangement the super-tasks, the system moves back from (*0,c*) to (*0,0*). In the mean while, the provisioning in cold pool, another super-tasks may arrive and takes the structure to state (*1,h*) which include there is one super-task in arrangement unit and one assume in the global line. Finally, the super-task under provisioning choice leaves the sending unit, once it gets a choice from asset allocator and the accompanying super-task will go under the arrangement. In this model, arrival rate γ_{st}, look up delays ($1/\beta_h$, $1/\beta_w$, $1/\beta_c$) and success probabilities (π_h, π_w, π_c) are exogenous parameters.

Super-tasks may face two types of blockings which can be calculated by using the steady-state probabilities $\mu(i,j)$:

- Blocking due to a full global queue occurs with the probability of,

$$B_q = \mu_{(n,h)} + \mu_{(n,w)} + \mu_{(n,c)}, \tag{4.1}$$

- Blocking due to insufficient resources (PMs) at pools occurs with the probability of,

$$B_r = \sum_{i=0}^{n} (\frac{(1-\pi_c)\beta_c}{\beta_c + \gamma_{st}})\mu_{(i,c)}, \tag{4.2}$$

The probability of reject is then, $P_{reject} = B_q + B_r$.

To calculate the mean holding up time in line, we build up the Probability Generating Function (*PGF*) for the number of super-tasks in the queue [8], as

$$Q(z) = \mu_{(0,0)} + \sum_{i=0}^{n} (\mu_{(n,h)} + \mu_{(n,w)} + \mu_{(n,c)})z^i, \tag{4.3}$$

The mean number of super-tasks in queue [8] is

$$\overline{q} = \overline{Q}[1], \tag{4.4}$$

Applying *Little's Law* [8], the mean waiting time in the global queue is given by (*first delay*):

$$\overline{wt} = \frac{\overline{q}}{\gamma_{st}(1 - B_q)}, \tag{4.5}$$

4.3.1 *MLR* Approach

For the betterment of this approach, here we have considered Multiple Linear Regression (*MLR*). *MLR* endeavors to demonstrate the relationship between at least two relevant factors and a reaction variable by fitting a straight condition to observed information. Each evaluation of the independent variable is related to an estimation of the dependent variable. By using *MLR*, we can predict resource allocation of user requests. So, it would be less of a concern of the blocking due to inefficient resources. The chances of (*i,h*) state moving to (*i, w*) or (*i, c*) would be much less than before. So no user request (*super-task*) will get rejected. Multiple Linear Regression endeavors to demonstrate the relationship between at least two informative factors and a reaction variable by fitting a straight condition to watch information. Each estimation of the independent variable z is connected with an estimation of the needy variable y.

The Multiple Linear Regression for p logical factors Z_p= $\{z_1, z_2, ..., z_p\}$ is defined to be,

$$y_i = \gamma_0 + \gamma_1 z_{i1} + \cdots + \gamma_j z_{ij} + \cdots + \gamma_p z_{ip} + \epsilon_i, \tag{4.6}$$

Where, for the i^{th} observation, y_i=Y and $z_{ij} = Z_j$,$\forall Z_j \in Z_p$. The *MLR* coefficient

γ_0, is the population block and γ_j, is the change in Y for 1 unit change in Z_j;$\forall Z_j \in Z$ and $1 \leq j \leq p$; holding other indicator factors consistent, ϵ_i is arbitrary or unexplained mistake connected with the i^{th} perception. Estimating $\gamma = \{\gamma_0, \gamma_1, \cdots, \gamma_p\}$, $\forall \gamma_j \in \gamma$ and σ_ϵ^2(variance of error), the fitted regression line that predicts Y and an obscure perception can be communicated as takes after,

$$\hat{y}_i = c_0 + c_1 z_{i1} + \cdots + c_j z_{ij} + \cdots + c_p z_{ip} \tag{4.7}$$

Where, $\hat{y}_i$ is the predicted value of y_i- any unknown observation , c_j is the sample estimate of γ_j, $\forall \gamma_j \in \gamma$. Instead of performing regression on every indicator variable independently, *MLR* utilizes the data from all indicator factors all the while to foresee the model one. Therefore, it is intrinsically quicker than other multivariate investigation techniques.

4.3.2 Resource Allocation Prediction Algorithm

Algorithm 1 finds the yield for the estimation of an arrangement of qualities of a specific task. It creates the assessed result for another demand T_{in}, in light of the past outcomes in Z_n. T_{in} contains a set of attributes $z_1, z_2, ..., z_n$.

On the off chance that the number is more noteworthy than three, the outcome is anticipated given the beforehand put away information. If the quantity of record is under three, it will anticipate the outcome utilizing [20].

Algorithm 1 *Optimal Resource Allocation Prediction for a request*

```
INPUT: $T_{in}$=$Z_1, Z_2, ..., Z_n$ , $T = t_1, t_2, ..., t_m$;
OUTPUT: $\hat{Y_i}$

if | $Z_n$ | $\geq 5$ then
    Calculate $\hat{Y_i}$ using the equation (4.7)
    Perform operation on input
    Measure $z$
else
    Predict allocation size $\hat{Y_i}$ by using [20]
    Perform operation on input
    Measure $z$
end if
```

4.3.3 EXAMPLE of Results using *MLR*

Table 4.1: Resource prediction using MLR

Users	File Size MB	Super-Task to Cloud Server	QoS	Actual CPU% Needed
User 1	23	28	0.78	12.2280148
User 2	24	28	0.60	11.00017549
User 3	25	27	0.79	12.10670967
User 4	28	30	0.77	12.25930796
User 5	29	32	0.81	12.74656659
User 6	12	18	0.56	09.843935194

In Table 4.1, we assume that the value generated during a specific time slot (i.e., 10-11am). In this way, our algorithm will generate value for the different time slot to cover the whole day. In this table, file rendering inputs (*File Size, Super-task to Cloud*

Server, QoS Time) and their outputs (*Actual CPU % needed*) appear. The estimations of (*User 1 - User 5*) are already put away in the table given the genuine forecast. After that when *User 6* asks for a similar task to the procedure, based upon its input value, the procedure will foresee the output values utilizing Multiple Regression for serving the task. Subsequent completion of the task it will investigate the genuine interest for that assignment and store both info and real yield values in the table. Presently if another demand goes to the server with a similar undertaking to the procedure, then based upon past six values new expectation will be resolved.

For finding the 6^{th} *user (Actual CPU % needed):*

$$y = -3.086838718 10^{-2} x_1 + 1.260667402 10^{-1} x_2 + 6.64983846 x_3 + 4.221244979 \quad (4.8)$$

4.4 Conclusion

So, in this chapter we have tried to explain the working function of our model by extending the *CTMC* as well as using some equations with the help of a well defined figure which will be very helpful to understand. Then, we have used Multiple Linear Regression to show that how our model gives almost error free results by which it will be easy to understand that our model gives much better performance to have energy efficiency in cloud computing.

Chapter 5

Performance Evaluation

5.1 Introduction

We are using *CloudSim* [21] to figure out the efficiency of our proposed model. The execution of our approach which we have denoted as *Triple-E* has been analyzed with *TEC*[13], *FCFS*[22] and *EMCO* [23] concerning energy utilization, execution time, the number of active physical machine, average delay and number of users.

5.2 Simulation Environment

We have organized an environment with necessary requirements to simulate which are as follows:

- First of all, eight (8) *PCs* of the same configuration are used as cloud data center where each *PC* has one *corei7* processor, *8 GB RAM*, and *1 TB* storage
- Another five (5) *PCs* with same configurations are used as users.
- The request arrival rate at our cloud data center is randomly taken from the range of *40-120* requests per minute.
- The serving time of each request is taken from the range of *5-60* minutes.

5.3 Experiment Results

5.3.1 Number of Task VS. Execution time

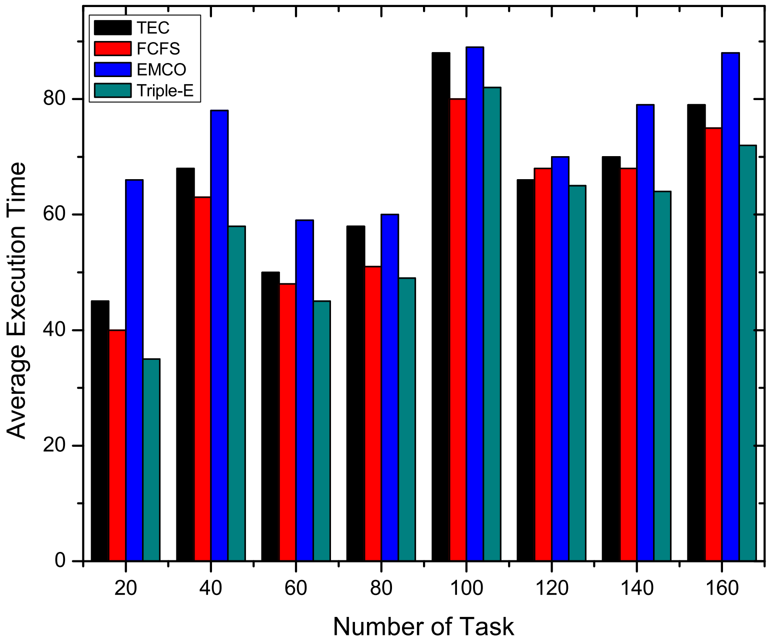

Figure 5.1: Execution time for varying no. of Task

In Figure 5.1, the execution time for performance analysis is shown for growing number of tasks. Here, execution time is taken in the account after executing a task in the *PMs*. The *MLR* process can calculate the execution time of different tasks. The results show the quality of services depends on the execution time which is proportional to the number of tasks. In the graph, considering the quality of service into the scheduling of *TEC*, *FCFS*, *EMCO* and *Triple–E* show that *FCFS* has competitive output according to the algorithm. However, our proposed approach is better at executing the task because of the *MLR* process.

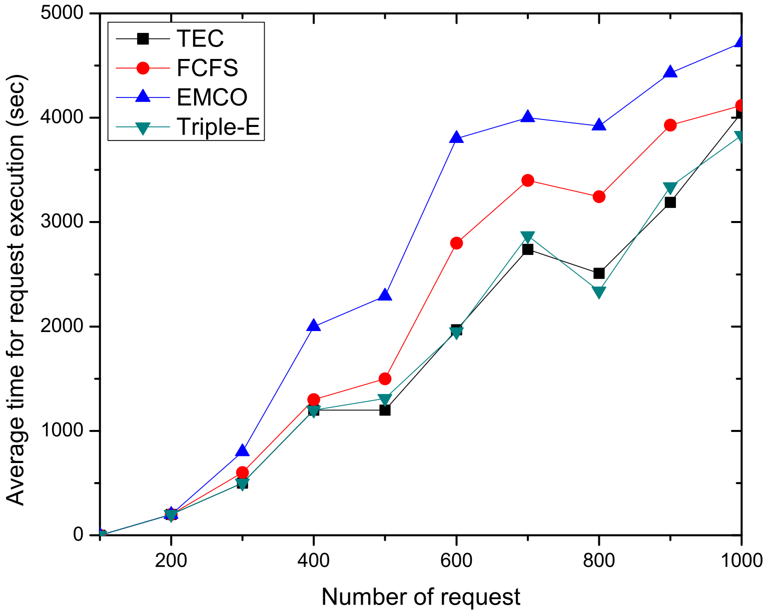

Figure 5.2: Execution time for varying no. of request

5.3.2 Average time for request execution(sec) VS. Number of Requests

In Figure 5.2, the relation between execution time(*sec*) and the number of requests (*super-tasks*) is shown. Here, the number of requests is taken from users throughout a day. The pool of requests to be gotten in every level is then anticipated in different times of the day, and every demand is accepted to have requested asset sums equivalent to those comparing to the class focus it has a place with. The predictions can be a little bit off at the initiate stage of the system. Our proposed work show optimal output during the prediction. *TEC* has exceeding status in predicting the requests from the server, but *Triple–E* also shows preferable output.

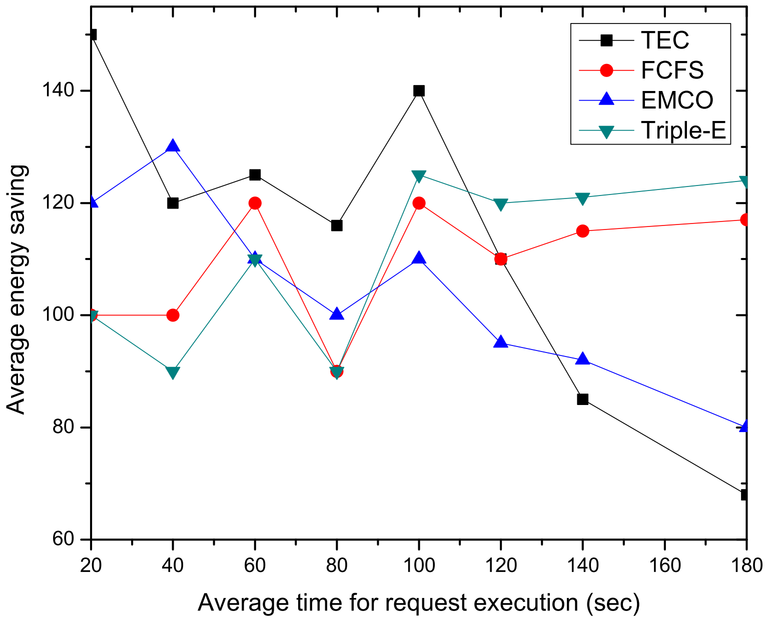

Figure 5.3: Energy saving for varying request execution

5.3.3 Average Time for request execution (sec) VS. Average energy saving

In Figure 5.3, it is shown how much energy our *MLR* process save; we plot the part of energy saved and estimated the amount while utilizing our proposed work compared with *TEC*, *FCFS*, and *EMCO*. The figure demonstrates that the *MLR*-based power management earned significant energy reserve funds and the measure of spared energy is near the ideal one, *FCFS*.

5.3.4 Average time for request execution(sec) VS. No. of active Physical Machine

In Figure 5.4, we evaluate time of the *PMs* that will be in the hot pool. The figure shows that over time the number of *PMs* needed to be active is increasingly high for most of the systems. As our proposed work is to predict the number of *PMs*, this section

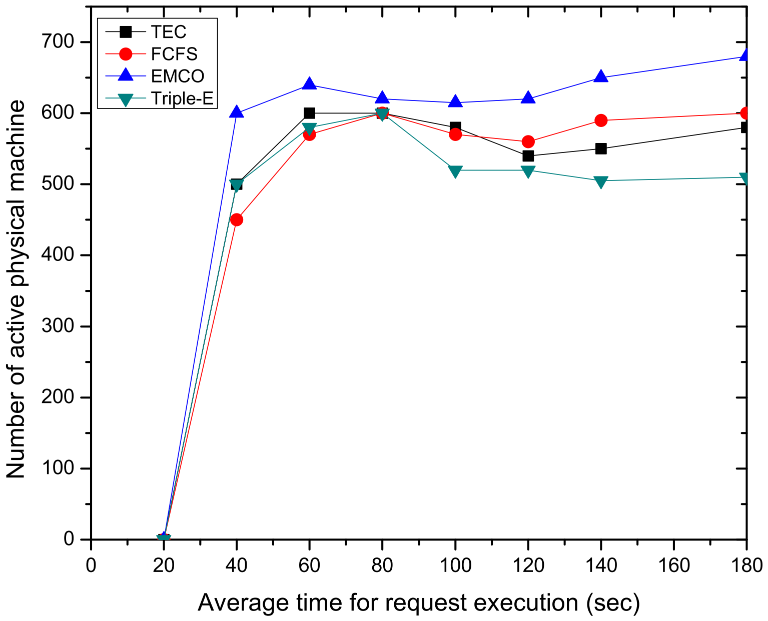

Figure 5.4: No. of acitve *PMs* for varying request execution.

is more important while executing for us. The calculation clearly shows that *Triple–E* have better output. The more PMs being active will drain more energy, so predicting precisely helps us to keep less *PMs* in active mode and save energy.

5.3.5 Average Delay VS. Average super-task size

In Figure 5.5, for the total task delay which is shown in our figure and describes that increase in super–tasks size leads to delay in executing the system. In the graph, we can see the almost flat peak and then quick fall off curves. This is because of the aggregate task arrival rate as the mean super-task estimate increments, and the actual super-task entry rate will diminish. However, our proposed work has a much favorable output for having less delay in service runtime.

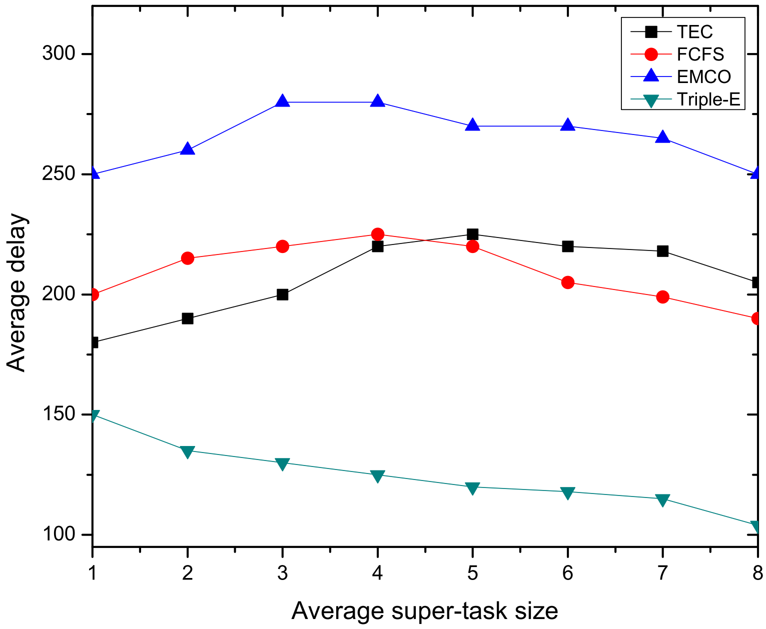

Figure 5.5: Delay for varying super-task size

5.3.6 Number of users VS. Average execution time

Figure 5.6, shows the evolution of service time that is dependent on the number of users. The result was obtained from simulation environment by applying the four different *VM* scheduling model. The solid lines show that as the user's increases, service time increases proportionally. It can be seen that the service time is greater in the technique we proposed. In our proposed work, reducing the information transmitted by the server is significant. It will help to decrease the complexity, the information which we receive will be easier to analyze, and the network is less collapsed.

5.4 Conclusion

So, comparing with *TEC*, *FCFS*, and *EMCO*, our model has given has given satisfactory results while evaluating the performance of our model. Our model has shown better

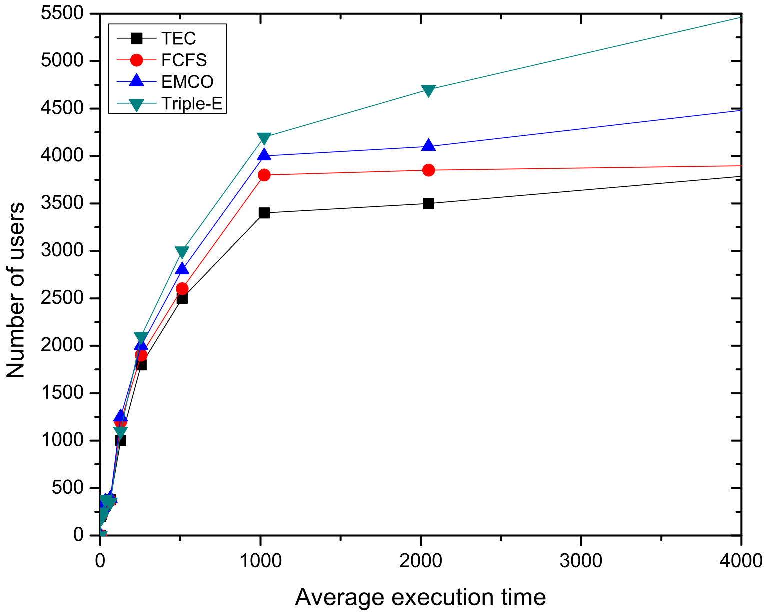

Figure 5.6: No. of users for varying execution time

energy efficiency in The graphs and discussions than rest of the three models. The performance metrics that we have considered has given satisfactory result to have a energy efficient cloud data center. We are very hopeful that in real life scenario, our model will be able to retain its better efficiency.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this report, we have built up an associating systematic model that catches vital perspectives including resource prediction, resource allocation, super–task requests (*super–task is a set of requests send by a user*), resource virtualization. Our proposed model can reduce the consumption of energy of today's cloud centers. Utilizing the proposed model, the most appropriate arrangement of tasks can be helpful to identify the characteristics of the arrived super-task for allocating resources to reduce the energy consumption of cloud center in advance. Then we have presented the working function of our proposed model by extending Continuous Time Markov Chain and also shown how our proposed model can process almost error free result by using the Multiple Linear Regression (*MLR*) approach. Finally, we have tried to evaluate the performance of our model by comparing with three other models *TEC*, *FCFS* and *EMCO* for which we have to consider some specific parameters to establish the performance evaluation of our model. By the help of such evaluation, we have tried to show that our model gives much better performance by which much energy efficiency can be ensured for allocating resources in cloud data centers.

6.2 Future Works

In future, we are aiming to work on the following issues:

- Super-tasks have not been distributed in multiple *VMs* in cloud data centers. So, we want to establish such a process so that the super-tasks can be distributed among Multiple *VMs* in Cloud data centers.
- Again, we want to establish such kind of mechanism which can be able to reduce the average execution time of Super Task.
- We have not given much preference on the trade-offs between costs and benefits in case of real world scenario while doing research to establish such model. Therefore, we can work on the costs and benefits of such issue.

So, we think that, if we can achieve these goals in our future works then cloud computing will be more energy efficient and also we will be benefited in other aspects.

Bibliography

[1] A. K. Das, T. Adhikary, M. A. Razzaque, and C. S. Hong, “An intelligent approach for virtual machine and qos provisioning in cloud computing,” pp. 462–467, Jan 2013.

[2] T. Adhikary, A. K. Das, M. A. Razzaque, and A. M. J. Sarkar, “Energy-efficient scheduling algorithms for data center resources in cloud computing,” pp. 1715–1720, Nov 2013.

[3] M. Barbulescu, R. O. Grigoriu, G. Neculoiu, I. Halcu, V. C. Sandulescu, O. Niculescu-Faida, M. Marinescu, and V. Marinescu, “Energy efficiency in cloud computing and distributed systems,” pp. 1–5, Sept 2013.

[4] C. Aschberger and F. Halbrainer, “Energy efficiency in cloud computing,” July 2013.

[5] A. K. D. M. Akter, F. T. Zohora, “Q-mac: Qos and mobility aware optimal resource allocation for dynamic application offloading in mobile cloud computing,” pp. 1–5, 2017.

[6] A. amazon.com company, “Amazon elastic compute cloud, amazonec2,” Website, Last accessed Feb. 2017, http://aws.amazon.com/ec2.

[7] IBM, “Ibm cloud computing,” Website, Last accessed Feb.2017,http://www.ibm.com/ibm/cloud/.

[8] F. Longo, R. Ghosh, V. K. Naik, and K. S. Trivedi, “A scalable availability model for infrastructure-as-a-service cloud,” pp. 335–346, June 2011.

[9] H. Khazaei, J. Miic, V. B. Miic, and S. Rashwand, “Analysis of a pool management scheme for cloud computing centers,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 849–861, May 2013.

[10] T. Adhikary, A. K. Das, M. A. Razzaque, M. O. Rahman, and C. S. Hong, “A distributed wake-up scheduling algorithm for base stations in green cellular networks,” pp. 120:1–120:7, 2012. [Online]. Available: http://doi.acm.org/10.1145/2184751.2184888

[11] A. K. Das, T. Adhikary, M. A. Razzaque, M. Alrubaian, M. M. Hassan, M. Z. Uddin, and B. Song, “Big media healthcare data processing in cloud: a collaborative resource management perspective,” *Cluster Computing*, pp. 1–16, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10586-017-0785-8

[12] T. Adhikary, A. K. Das, M. A. Razzaque, M. Alrubaian, M. M. Hassan, and A. Alamri, “Quality of service aware cloud resource provisioning for social multimedia services and applications,” *Multimedia Tools and Applications*, pp. 1–25, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11042-016-3852-x

[13] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, “Toward energy-efficient cloud computing: Prediction, consolidation, and overcommitment,” *IEEE Network*, vol. 29, no. 2, pp. 56–61, March 2015.

[14] Y. Sharma, B. Javadi, W. Si, and D. Sun, “Reliability and energy efficiency in cloud computing systems: Survey and taxonomy,” *Journal of Network and Computer Applications*, vol. 74, pp. 66 – 85, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516301746

[15] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center," 2008.

[16] A. K. Das, T. Adhikary, M. A. Razzaque, and C. S. Hong, "A qos and profit aware cloud confederation model for iaas service providers," pp. 1–7, 2013.

[17] T. Adhikary, A. K. Das, M. A. Razzaque, A. Almogren, M. A. Alrubaian, and M. M. Hassan, "Quality of service aware reliable task scheduling in vehicular cloud computing," *MONET*, vol. 21, pp. 482–493, 2016.

[18] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755 – 768, 2012, special Section: Energy efficiency in large-scale distributed systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X11000689

[19] H. Khazaei, J. Misic, and V. B. Misic, "A fine-grained performance model of cloud computing centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2138–2147, Nov 2013.

[20] Y. Jiang, C. S. Perng, T. Li, and R. N. Chang, "Cloud analytics for capacity planning and instant vm provisioning," *IEEE Transactions on Network and Service Management*, vol. 10, no. 3, pp. 312–325, September 2013.

[21] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1002/spe.995

[22] D. S. C. N. Susila, "Energy efficient extended fcfs load balancing in data centers of cloud," pp. 599–605, 2016.

[23] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, October 2016.

Appendix A

List of Acronyms

3D	Three Dimensional Space
VM	Virtual Machine
PM	Physical Machine
PC	Personal Computer
IT	Information Technology
ICT	Information and Communication Technology
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
SLA	Service Level Agreement
MLR	Multiple Linear Regression
PDA	Personal Digital Assistant
EPEAT	Electronic Product Environmental Assessment Tool
VMM	Virtual Machine Monitor
SAN	Storage Area Network
PFIFO	Priority based First-in, First-out
RP	Resource Predictor
RA	Resource Allocator
QoS	Quality of Service
CTMC	Continuous Time Markov Chain

Appendix B

List of Notations

$\neq$	This is not equal to sign
$\leq$	This is less than equal to sign
μ	This is mu
π	This is Pi
β	This is Beta
γ	This is Gamma
$\sum$	This is sign of summation
ϵ	This is epsilon
$\forall$	This is for-all sign
$\cdots$	This represents the dotted line

Appendix C

List of Publications

International Conference Paper

1. Md. Shahidul Hoque, Nabil Shawkat, Adnan Asif Chowdhury, Amit Kumar Das, (“Energy Efficient Extended Pool Management Scheme in Cloud Data Centers”), *IEEE* International Conference on Innovations in Power and Advanced Computing Technologies, (*i-PACT-2017*), Chennai, India, 2017.