Elasticity in Cloud Computing using Automatic Policy

A Project Report submitted in fulfillment of the requirement for the award of the degree of

Master of Technology

in

Computer Science & Engineering

under the Supervision of

Prof. Dr. S.P. Ghrera

by

Kshitiza Vasudeva

Enrollment No: 142209

Jaypee University of Information Technology,Waknaghat, Solan, Himachal Pradesh-India 173234

Certificate

This is to certify that thesis report entitled **"Elasticity in Cloud Computing using Automatic Policy"**, submitted by **Kshitiza Vasudeva** in fulfillment for the award of degree of Master of Technology in Computer Science & Engineering to Jaypee University of Information Technology,Waknaghat,Solan has been made under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Supervisor's Name – Prof. Dr. S.P. Ghrera

Dated:

Signature

Acknowledgement

I would like to take this opportunity to acknowledge all those who helped me during this report work. Compiling a year's work into this was an exhausting job, but writing this page of acknowledgement is a joyous task to cherish the memories of all those who helped me to enrich the newer experience of life.

At the very onset, I bow my head with reverence and dedicatedly accord my recondite and gratitude to "ALMIGHTY", the merciful and compassionate, whose grace, glory and blessings allowed me to complete this endeavor and without his encouragement and co-operation it would have never been possible for me to achieve this.

I owe my deep sense of respect and heart felt gratitude to my major supervisor ***Prof. Dr. S.P. Ghrera, Head of Department, Computer Science and Engineering Department, Jaypee University of Information Technology*** for his meticulous and sagacious guidance, sympathetic encouragement, precise and constructive criticism and ever willing help throughout the course of this investigation as well as in the preparation of manuscript. I will always remain indebted to him for his unending guidance and untiring efforts in successful completion of this work. I consider myself fortunate to have worked under his able guidance. I express my sincere and whole hearted thanks to him for rendering help and moral support.

I am thankful to office staff of the department for providing all the necessary and timely help. I am also thankful to respondents of my study for their co-operation who helped me to complete my study.

I wish to express my sincere thanks to all my friends for their support and guidance. There is paucity of words to express my heartiest thank to my friends ***Ms Aditi Zear*** and ***Ms. Swati Sharma and Mr. Abhishek jaswal*** for their timely help, best wishes and cheerful company remained a morale booster and made things smother throughout the course of this study.

I owe my achievements to the unconditional love and support of my parents whose sacrifice I can never repay. They inspired me at every step of my life and encouraged me to never give up even in the face of overwhelming odds. I grope for words to express my deep feelings, love and affection to my younger brother.

Last but not least I would like to express my gratitude to all those who have helped, guided and supported me in one way or the other but have been inadvertently left out because all may not have been mentioned but none have been forgotten.

Needless to say, omissions are mine.

Name of the student- Kshitiza Vasudeva

Dated:

Signature

Table of Content

List of Figures

List of Tables

List of Acronyms

Acronym	Description
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
NN	Neural Network
SVM	Support Vector Machine
RMSE	Root Mean Square Error
MAPE	Mean Absolute Percentage Error
SLA	Service Level Agreement
SLO	Service Level Objectives
IQR	Inter Quartile Range
MAD	Mean Absolute Deviation
THR	Threshold
MMT	Minimum Migration Time
RC	Random Choice
LR	Local Regression

ABSTRACT

Cloud computing is among the most fast growing and symbolic contemporary technologies that has revolutionized modern ICT. To deal with growing demand, cloud needs to establish large data centers consisting of optimized computing machines. Elasticity is an asset of cloud computing which makes it better or may be best over the other conventional grid and cluster computing. Data centers in cloud whether private, public or hybrid use elastic nature of cloud to provision Virtual Machines (VMs) with remarkable flexibility. However, operating and maintaining those underlying physical resources incurs large amount of energy consumption causing significant cost to the provider and CO_2 emission leading to environmental impact. Therefore, cloud providers must optimize the VM allocation to the physical resources, constantly balancing between the conflicting requirements on performance and operational costs. Consolidating VMs effectively on the physical machines utilizes the resources efficiently. Proper utilization of resources saves electrical energy, cost and reduces CO_2 emission too. Virtualization and VM migration are core part of this optimization process.

The goal is to reduce energy consumption by effectively utilizing the resources by following the objectives the Service Level Agreement (SLA). Dynamically consolidating VMs cause fluctuations in workload leading to live migration of VMs. VMs are migrated among the physical nodes to reduce the number of working physical hosts. This live VM migration causes delay and overburdens the network as well as the physical hosts involved. Hence, violates SLA and degrades the performance of the system. So it adds up to the goal to reduce energy consumption as well to minimize the number of VM migrations. The thesis is divided into six different chapters. Chapter wise description of the thesis report is described as follows:

Chapter 1 presents the detailed introduction to Cloud Computing technology with its evolution. It discussed the complete concept and all the predecessor technologies involved in its evolution. Definition for Cloud Computing provided by NIST is mentioned which further describes the service models, deployment models, five essential characteristics and the various challenges found while working with Cloud Computing. At last problem context is discussed describing the complete scenario of problems found in this context.

Chapter 2 presents the literature survey done in the context of problems mentioned in previous section. Firstly elasticity feature of cloud is discussed and classified on the basis of scope, policy, purpose and method. Many researchers have focused on just improving the elasticity of cloud. This work is compared and presented in the table 1. This survey raised the issue of VM allocation problem because of elastic nature of cloud and trying to consolidate the VMs so as to deal with dynamic workload. Complete problem of VM allocation has been discussed with parameters like Virtual Machines (VMs), Physical Machines (PMs), Service Level Agreement(SLA), Resources and Live Migration. Literature review for all of these is mentioned and at last compared and shown in table.

Chapter 3 describes the problem in detail after reviewing the complete literature in the previous chapter. Firstly problem description and then problem statement is given in short. Proposed solution is also mentioned in brief.

Chapter 4 is about evaluation of existing algorithms in the respective of the proposed solution mentioned in previous chapter. CloudSim framework is used to setup the system and compare the algorithms. The parameters used for comparison are power consumption in kWh, Number of SLA violations and Number of VM migrations. The policies compared are IQR-MMT, LR-MMT, MAD-MMT, THR-MMT and LR-RC. Test cases were setup and results were noted down for the parameters mentioned.

Chapter 5 is the proposed part of the thesis. In order to minimize the number of VM migrations and power consumption, Self-Healing is proposed. Concept says before migrating a VM while VM consolidation self-heal the host so as to reduce the power consumption. This will finally reduce the migrations. It consists of proposed model architecture, main contribution part to the system, block diagrams, various modules of the proposed system and the algorithm. CloudSim simulation is tool is used to implement the algorithm. The parameters used for comparison are Power consumption in kWh and number of VM migrations. The proposed algorithm and the existing system are compared on the basis of the parameters mentioned. At last results are noted and compared through graphs.

Finally, conclusion and findings of the thesis and future directions are presented in **Chapter 6**.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Among all the newfangled technologies, cloud computing has completely revolutionized IT industry. To cover all the features, cloud computing has acquired all limitations, checks and advancements of other computing research areas like virtualization, utility computing, service oriented architecture, autonomic computing, distributed and grid computing. Cluster and Grid Computing being the most obvious predecessor technologies that enabled the outlet of cloud computing. Thereupon, several business models expeditiously evolved to harness this profitable technology by providing software applications, programming platforms, data-storage, computing infrastructure and hardware as services. It follows 'pay-per-use' model, customer has to pay for only those resources which he/she has used. Developers these days need not to worry about the hardware to deploy their service or the problem of overprovisioning / under-provisioning of resources. Cloud computing on whole covers the applications delivered over the internet as a service and the Data Center hardware and software. The goal of this computing model is to make software even more attractive as a service, increase availability of resources and higher throughput.

US Government's National Institute of Standards and Technologies (NIST) [36] defines, "*cloud computing is 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*". This cloud model clearly describes five essential characteristics, four deployment models and three service models.

1.2 Service Models

All allow users to run applications and store data online however each offers a different level of user flexibility and control.

1. **SaaS**(Software as a Service)- Allows users to run existing online applications.
2. **PaaS**(Platform as a Service)- Allows users to create their own cloud applications using supplier specific tools and languages
3. **IaaS**(Infrastructure as a Service)- Allows users to run any application they please on cloud hardware of their own choice.

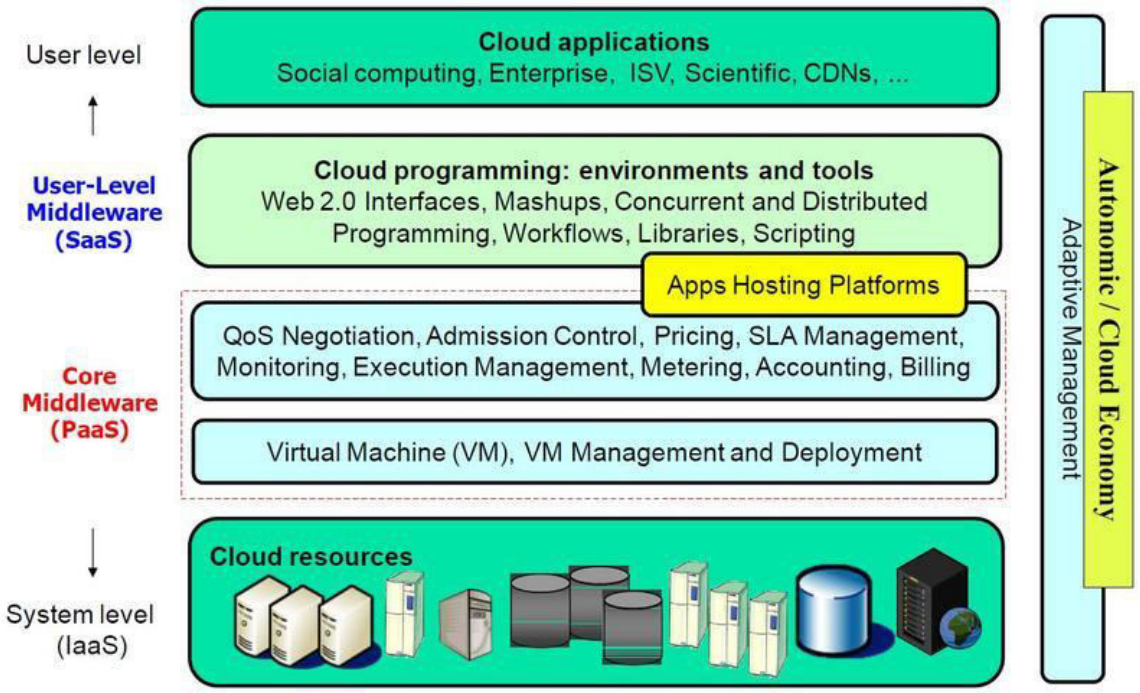

Figure 1: Layered architecture of Cloud

1.3 Characteristics

1. **On-demand capabilities:** One can access to their own services and also modify the cloud services with the help of online control panel. One can remove and add the users accordingly and also change the can also change the software as well as

networks as per the requirement. At the end, users are supposed to pay on the basis of pay-per-use model.

2. **Broad network access:** To promote heterogeneous platforms like PDAs, mobile and laptops, cloud services are available over the network and can be accessed through simple mechanisms following standards.
3. **Resource pooling:**The cloud providers' pool their resources together so as to serve various users by dynamically assigning them the physical machines and virtual resources depending upon the demand.
4. **Rapid elasticity**: elasticity is the ability of the system to rapidly and automatically scale in or scale out the resources depending upon the current demand. The services and resources appear unlimited to the users so that they can buy in any quantity.
5. **Measured service**: Resource usage by the customers can be monitored, measured, guarded and proclaimed which provides transparency in the use of service for both cloud provider as well as customer.

The cloud providers provide the hardware and software resources which can be provisioned dynamically like utilities. Cloud provides elasticity as a feature which allows users to use resources as per the requirement at any time.**Elasticity** in the context of cloud computing means the capability of the system to expand or shrink the number of resources in an automatic manner so that SLA is not violated with minimum cost incurred. The main elasticity **parameters** or dimensions of any cloud application are *cost, resource* and *quality*. Each cloud application or process tries to increase or decrease the cost, maximize resource utilization and improve the quality so as to accommodate the specific requirements. Furthermore, elasticity captures two core aspects: *speed* and *precision*. Speed can be considered as the time taken by it to swap under-provisioned state and optimal state or vice versa. Precision is difference in the number of currently allocated resources and actual demand.

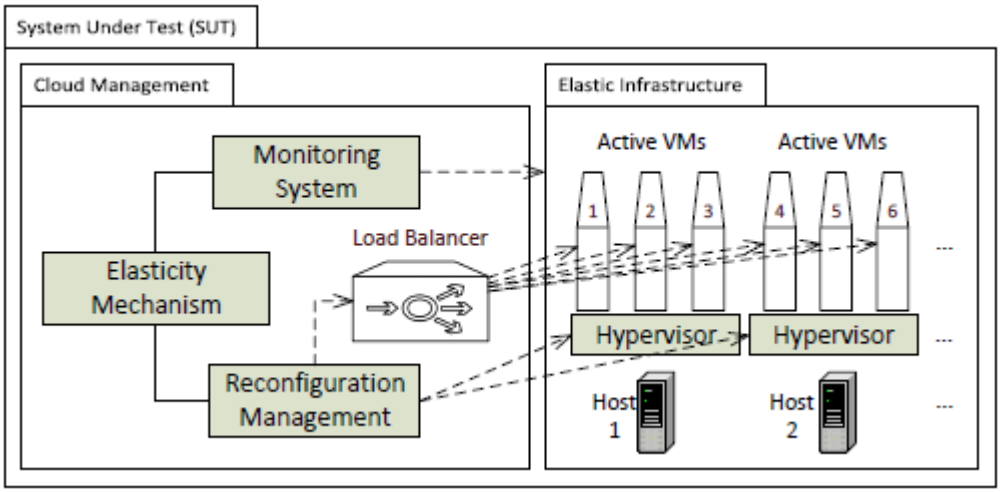

Figure2: Blueprint of Elastic Cloud System architecture

1.4. Challenges

1) Resource availability
2) Clouds interoperability
3) Resource granularity
4) Start-up time of VMs
5) Tools and platforms for elastic applications development

1.5. Problem Context

Focusing on purpose of elasticity and the various challenges faced while processing the requests in cloud data centers (DCs).From the perspective of the provider, the elasticity of cloud makes sure that computing resources are utilized in a better way, providing economies of scale and to allow the simultaneous use of resources by many users. From the user's perspective elasticity mostly avoids the inadequate provisioning of resources and degradation of system performance. Large, virtualized data centers (DCs) are serving the ever-growing demand for computation, storage, and networking. The efficient operation of DCs is increasingly important and complex [37].

To insure isolation of applications and robust utilization of physical resources, cloud DCs make use of virtualization technology. It enables multiple VMs to be placed on the same physical machine or host. Virtual machines (VMs) are available to users/customers directly as resources in case of Infrastructure as a Service (IaaS) or used to wrap the provisioned applications in case of Software as a Service (SaaS) and Platform-as-a-Service (PaaS) [38].

It is essential to note that for serving multiple customers to host their applications many large scale data centers are built containing thousands of computing nodes which consumes huge amount of energy and cost[39].Though there are other traditional cost factors like staff and equipment but energy consumption has become critical because of the environmental impact. According to recent study DC energy consumption is the fastest-growing part of the energy consumption of the ICT ecosystems; moreover, the initial cost of purchasing the equipment for a DC is already outweighed by the cost of its ongoing electricity consumption [40].Not just the increasing cost, high energy consumption causes low system reliability since failure rate of electronic devices is directly proportional to rise in temperature. Furthermore, it is estimated that each computer's usage contributes around 2 percent of anthropogenicCO_2 emission. Data center activities are estimated to release 62 million metric tons of CO_2 into the atmosphere [41]. Therefore, it is highly crucial to employ some techniques or a way out to reduce the energy consumption.

Virtualization being a central part of cloud computing can be used efficiently for saving energy in data centers. Concept used is to consolidate the VMs to the minimal number of physical hosts and turning off the unused hosts or changing their mode of operations like sleep mode. Live migrations of VM are used to reduce the energy consumption by migrating VMs from overloaded hosts to less utilized hosts. In underutilized hosts, all the VMs are migrated to other hosts and it is switched off to save energy.

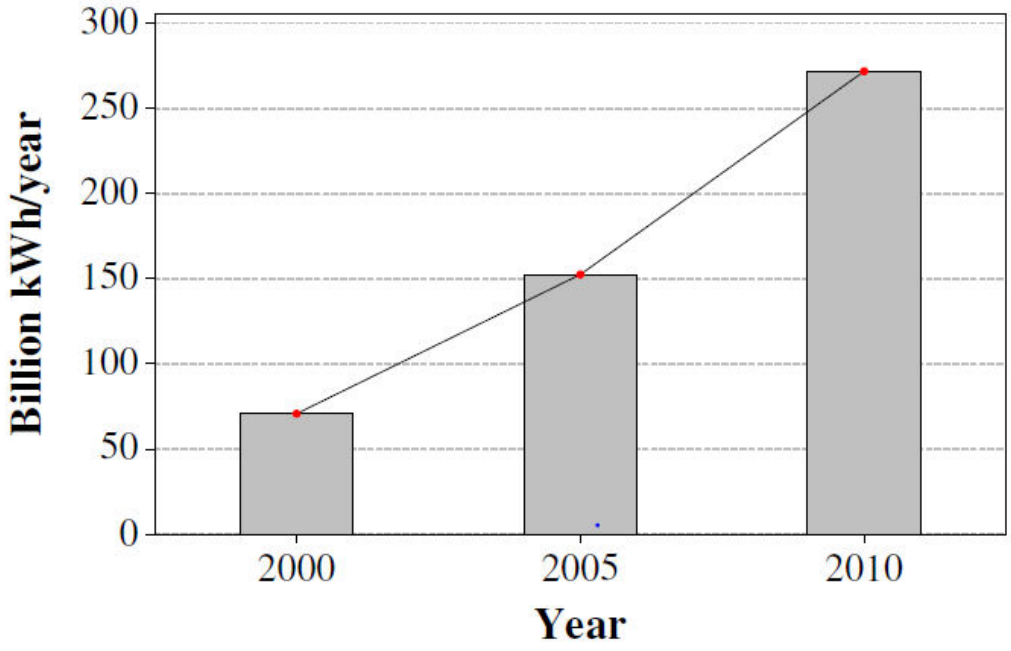

Figure 3: The Worldwide Data center energy Consumption 2000-2010[56]

However, VM consolidation that is too combative may lead to overloaded hosts and damage the Quality of Service (QoS). This may further contravene the service level objectives (SLOs) set in the agreement between the cloud provider and the customer. Some systems also need to pay the penalty cost for violating the SLOs.

Hence, there is a trade-off between QoS and energy consumption so VM allocation must find the optimal balance between them. [42-43].VM allocation problem can be characterized as follows:

- The Cloud Provider(CP) owns the data center (DC) consisting of physical machines PMs (hosts) or rents machines form eCPs (External CP)
- Users request for resources i.e. VMs with different specifications (e.g. memory, computational power in MIPS, storage, network communication)which can vary over time.
- The CP accommodates VMs on the available machines PMs. The number of these VMs changes over time due to upcoming requests to create additional VMs or to remove existing ones.

- The PMs also have a specified capacity in terms of above mentioned specifications like VMs.
- The resources are available to users on the basis of pay-per-use model. The use of resources incurs some cost and electrical power which may vary depending on the type and utilization percent of resources.
- Now to reduce power consumption some machines are turned off by migrating VMs from underutilized hosts to others. This live migration may also further cause delay and extra load on the PMs involved and network.
- Over-utilizing some hosts violates SLOs in the Service Level Agreement which may count as penalty and can lead to extra expenses too.

Considering the points mentioned above, one can conclude that VM placement is a multi-objective problem discussed in many research works [44-46].

Some of those objectives for the CP are:

- Minimize overall energy consumption
- Minimize carbon emission
- Minimize number of SLA violations
- Avoid penalties
- Minimize number of VM migrations
- Maximize utilization of resources
- Minimize operation costs

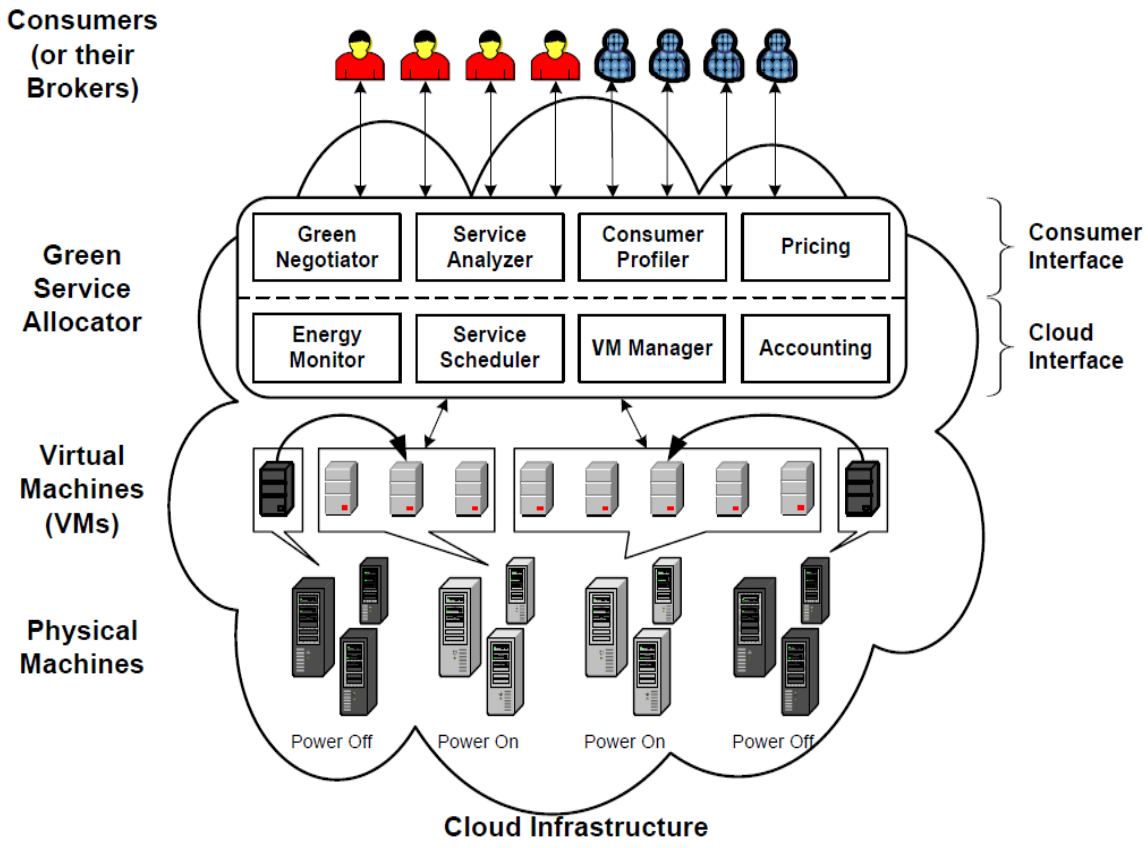

Figure 4: System Architecture

Chapter 2

Literature Survey

2.1 Elasticity

The elastic nature of cloud enables cloud providers to allocate resources as per the requirement of the user. Resources can be scaled up and down according to the need. In IaaS, by resources we mean Virtual Machines. The Virtual Machines are created, destroyed or migrated from one Physical Machine to other as per the situation. These decisions directly affect the operational cost, energy consumption and QoS. Hence, Vm allocation problem is one of the core challenges faced by Cloud providers and users too indirectly.

Firstly, Elasticity as an important feature of cloud computing is studied. Elastic resource provisioning [11] is somewhat like winsome feature provided by Infrastructure as a Service (IaaS) clouds but to decide how many resources to get and when to get make it bit complicated while changing application workload dynamically.

Title: A survey on cloud computing elasticity

Authors: G. Galante and L. C. E. d. Bona

They proposed a solution for complex and vast elasticity mechanisms by classifying these mechanisms on the basis of features found in studied academic & commercial solutions. They classified elasticity on basis of scope, policy, purpose and method.

The two most common policies of an elastic cloud application are: **manual and automatic**. The term *policy* as a characteristic is related to those interactions which are needed while executing elasticity actions. The manual policy and automatic are different on the basis of who is responsible for monitoring his/her virtual environment, applications and then taking an action to perform elasticity. In *manual policy* user is responsible for all this work and in *automatic policy* the control and actions are taken by the cloud system or the application itself without the intervention of users. Some public providers which manage resources manually are: GoGrid, Rackspace and Microsoft Azure, and the frameworks Elastin and Work Queue.

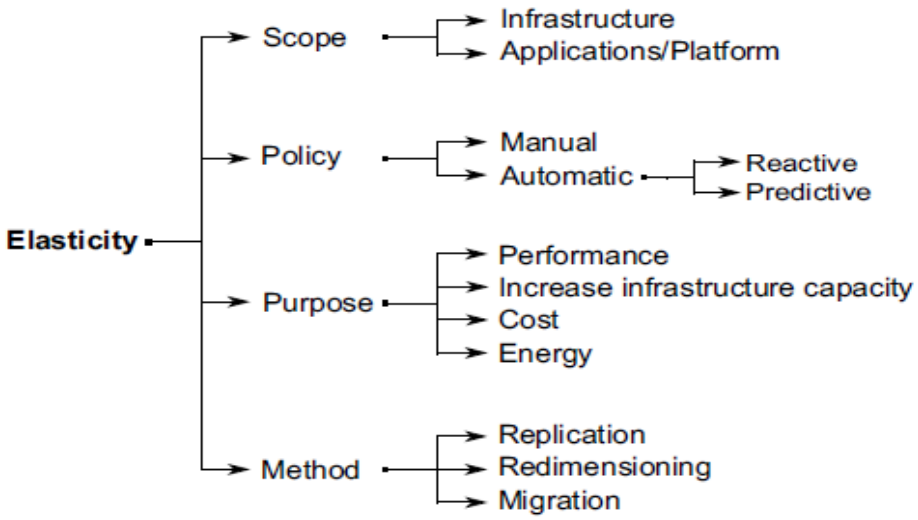

Figure 5: Classification of elasticity

Automatic / auto-scaling techniques can be further classified into reactive, proactive and hybrid.

Reactive techniques are also known as rule based methods. The system reacts to changes but doesn't anticipate them. Each rule has some conditions which when satisfied some action is triggered. These conditions are based on the threshold values which vary according to the system. Reactive methods are popular in research and practice [17-19]. For instance, reactive methods are used by many public cloud providers (like Amazon, Microsoft), cloud platforms (likeOpenNabula), and third party tools (like RightScale). Threshold-based rules are explicitly mentioned and popular among current research work [20-23]. RightScale's auto-scaling algorithm [24] uses a voting process, that is, all nodes vote for scaling up or down and if majority of the nodes agree then that particular action is performed. This RightScale's auto-scaling algorithm is a complement to reactive rules.

Title: Autonomic resource provisioning for cloud-based software.

Authors: Jamshidi, Pooyan, Aakash Ahmad, and Claus Pahl

Reactive approach has some shortcomings like the parameters and threshold values which are keys in rules require deep knowledge, an extra effort and expertise. Furthermore, all the existing approaches don't deal with the uncertainty caused by noise and unexpected events in cloud based software which is very common if we think out of theoretical

concepts. So**,** they proposed a solution to this problem by developing RobustT2Scale, an elasticity controller which enables quantitative specifications of elasticity rules by utilizing fuzzy logic. These Fuzzy logic systems can manipulate linguistic rules so that conflicting rules can be handled. It is robust to noisy data too [2].

The virtual resources that cloud computing uses while scaling dynamically don't have negligible setup time. Reactive approach can't solve this problem and also incurs huge cost. So, **Proactive techniques** are used. These techniques try to predict future resource demand in order to ensure that sufficient resources are available before time. For this prediction some heuristics and analytical methods are used so as to conjecture the systems load behavior and then to decide to scale in/out resources on the basis of the results. In this context, Caron et al. [3] was the one who initiated groundwork for this new approach by developing resource usage prediction algorithm. Some references of the works done by authors using predictive techniques to scale resources are Gong et al., Vasi´c et al. , Shen et al., Sharma et al., Roy et al., Dawoud et al..

Time series analysis, machine learning, queuing models and control theory are some popular techniques used in predictive approach. Time series analysis use historical data usage to predict the future resource demand and work on particular domain. It performs well and better if provided with large historical data and interval size is optimum. Reinforcement learning enables the policies to learn from observations. But it is suitable for only stable workloads because prolonged learning is required. Queuing theory sets many restrictive assumptions. And due to this restriction only stationary scenarios fulfill these assumptions so whenever conditions change it needs to calculate the values again. Last, the controllers take some input and give an output which should be maintained at some desired level. Outputs change as the values of the input parameters change.

Title: An adaptive hybrid elasticity controller for cloud infrastructures

Authors: Ali-Eldin, Ahmed, Johan Tordsson, and Erik Elmroth

Hybrid auto-scaling, the third category of automatic policy combines the other two reactive and proactive. Reactive is considered when working on short time scale and proactive when time scale is long. In this category they introduced two adaptive hybrid controllers Pc1 and Pc2 that use hybrid approach to know the current and predict future

demand. Then on the basis of this prediction it dynamically scales the VM resources in a cloud. Results proved that after using reactive technique for scaling up and predictive for scaling down SLA violations rate improved 2-10 times when compared to only reactive approach[4] .

Title: Optimizing the Cost for Resource Subscription Policy in IaaS Cloud

Authors: Ms.M.UthayaBanu, Mr.K.Saravanan

They proposed a solution to minimize the service provision cost in both reservation and on-demand plan using hybrid approach. They divided the resource subscription problem into two sub-problems: how many long term resources to be reserved and how many on-demand resources to be acquired. They proposed a two-phase algorithm. In the first phase, a mathematical formula is used to reserve correct and optimal amount of resources during reservation and in second phase, Kalman Filter is used to predict resource demand. The results showed that it significantly reduced provision cost and prediction is of reasonable accuracy [5].

Title: Lightweight resource scaling for cloud applications

Author: Rui Han, Li Guo, Moustafa M. Ghanem, and YikeGuo

Many cloud services using VM level scaling may overuse resources increasing the operating cost of the cloud provider. They gave a solution for this extra cost as well as overuse of resources in cloud services. They proposed a lightweight scaling (LS) algorithm to enable fine grained scaling of an application at the level of underlying resources namely CPU, memory and input/output. The algorithm tries to use the idle resources at the max to release overload resources before scaling in other nodes which increases resource utilization of PMs. This approach efficiently meets the QoS requirements and also reduces the cost by scaling resources in/out [6].

Cloud applications handle dynamic scalability through virtualization. The drawback of virtualization is that the setup time of virtual machines is non-zero. This drawback can't be neglected when considering efficiency and performance.

Title: Forecasting for Grid and Cloud Computing On Demand Resources Based on Pattern Matching

Authors: E. Caron, F. Desprez, and A. Muresan

They proposed a new resource usage prediction algorithm for solving this problem of non-zero setup time. The algorithm uses historic data saved in the past to match similar usage patterns of the current window of records. Then after matching, algorithm predicts the system usage by interpolating what follows after those matched patterns from the historic data. Algorithm proves better when provided with input data of same application domain and improving on the data size plus interval size [3].

Title: AGILE: Elastic Distributed Resource Scaling for Infrastructure-as-a-Service

Author: Hiep Nguyen, Zhiming Shen, Xiaohui (Helen) Gu, Sethuraman Subbiah ,John Wilkes

They proposed a system AGILE, a practical elastic distributed resource scaling system for IAAS cloud infrastructures. Along with dynamic work load changes AGILE also considers the interference from other users at runtime. AGILE provides medium-term resource predictions so that there is enough time for scaling up the resources of the server and application's SLO is not affected by workload increase. AGILE implements live cloning to scale up the performance by replicating running VMs ahead of time. In contrast to previous resource demand prediction schemes, AGILE achieves enough lead time for setting up the VMs with good prediction accuracy. By combining this medium term resource demand predictions and online profiling AGILE can very well predict whether an application will face an extra workload. And if it happens then how many new servers should be added to avoid that situation [9].

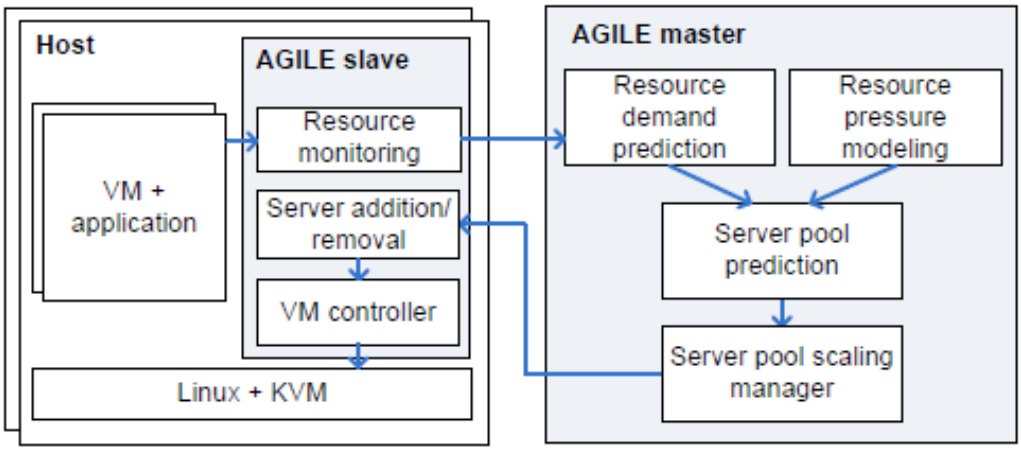

Figure 6: Overall structure of AGILE

Table 1: Literature review(Elasticity)

S.No	Paper	Cost	Resource	Quality	SLA	Elasticity policy
1.	Nguyen et al.[2013]	Yes	no	no	yes	Automatic(hybrid)
2	Eldin et al.[2012]	No	no	no	yes	Automatic(hybrid)
3.	Jamshidiet al.[2014]	Yes	yes	no	yes	Automatic(Reactive)
4.	Caron et al.[2010]	Yes	no	no	no	Automatic(proactive)
5.	Rui Han et al.[2012]	Yes	yes	no	no	Automatic(reactive)
6.	M.UthayaBanu, K.Saravanan [2013]	Yes	yes	no	no	Automatic(Hybrid)
7.	Samuel et al.[2013]	No	no	no	yes	Automatic(proactive)
8.	Jara et al.[2009]	Yes	no	no	yes	Automatic(hybrid)
9.	Copil et al.[2013]	Yes	yes	yes	yes	Automatic
10.	Islam et al. [2012]	Yes	no	no	yes	Automatic(proactive)

2.2 VM allocation problem

VM allocation problem has been considered by many research scholars and have proposed solutions for it. All have taken up the different version of it. This problem of VM allocation can be classified on the basis of the parameters considered while finding a solution. The entities involved in cloud computing like Resources, SLOs(System Level Objective), Virtual Machines (VMs), Physical Machines (PMs), VM migration factors, etc. all have different parameters to be specified [47].

1. Virtual Machines(VMs)

A virtual machine has following characteristics:

- The number of cores in CPU
- The CPU capacity per core(in MIPS)
- Size of RAM (in gigabytes)
- disk Size
- Bandwidth
- Latency

2. Resources

The cloud provider can avail resources either directly in the form of Physical Machines owned by CP or through external Cloud providers (eCPs) in the form VMs on lease. When PMs are owned by the CP, it has complete information about the state of PMs which includes workload, power consumption, etc so it is Cloud Provider's duty to optimize the resource utilization. On the other hand, CP has no information about the infrastructure when VMs are taken on lease from eCPs. They just request the VMs and manage them without any burden of optimization. Resources from eCPs are expensive as they add their own profit too. A CP may own more than one DC which may add to communication latency. Live migrations are performed within one DC.

3. Physical Machines (PM) characteristics

Physical machines or hosts also have same resource characteristics as VMs like processing capacity, storage, CPU cores, etc but in large amount which are divided among the VMs created on physical machine. The PM utilization is measured on the basis of utilization of the VMs hosted on the machine. The resources considered while

measuring utilization adds to the version of VM allocation problem. The most demanding resource considered by most of the researchers is CPU. High load on CPU may degrade the performance and violate SLA whereas less utilization of CPU can leave resources under-utilized. Therefore, utilization of CPU is an important factor in VM allocation optimization. Power consumption of a PM is a monotonously increasing function of the CPU load [48]. However, other resources like disk storage and memory can also affect the performance of the system and taken into account while measuring the utilization of PM[49-52].

Title: Power-aware virtual machine scheduling on clouds using active cooling control and DVFS
Author: Daniel Guimaraes do Lago, Edmundo R. M. Madeira, and Luiz Fernando Bittencourt.
According to recent studies, it has been shown that idle resources also consume some amount of energy in fact 70 percent of the peak energy. By idle resources, we mean no VM is accommodated on it. Therefore, turning off or changing the mode of operation to low-energy state of PMs becomes essential to save more energy.Cloud data centers usually face variations in number requests which increases or decreases the load on PMs. Dynamic Voltage and Frequency Scaling(DVFS) technique is inherent in microprocessor technology to maintain the performance of PMs. DVFS is used to scale up the frequency of the CPU in times of heavy load to increase performance at the cost of higher power consumption and scale down in times of low load to decrease the consumption of energy [53].

4. Service Level Agreement(SLA)

SLA is an agreement between a cloud provider and a customer mentioning the expected Quality of Service(QoS). This agreement sets some objectives (SLOs) for Cloud Provider while servicing the requests of customers. Mostly SLA is a formal document with all customer and CP belong to the same organization. SLOs can be categorized as hard and soft SLOs. A hard SLO is the one which has to be accomplished definitely whereas soft SLO may be violated in some worst case scenarios. Another categorization is on the basis of level of abstraction. User-level SLOs are quality metrics with respect to users (e.g.

response time, throughput) and System-level SLOs define objectives for performance of the system as a whole(e.g. system availability, reliability).An SLA violation occurs when some of the objectives are not fulfilled. Usually this happens due to improper allocation of resources(e.g., less processing capacity to a VM). Sometimes inefficient scaling decisions and inappropriate sizing of VMs can lead to violation of SLOs [55].

5. Live Migration

Title: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers

Author: Anton Beloglazov and Rajkumar Buyya

Migration of Virtual Machine from one host to another is an important way out to solve many issues in VM allocation problem. Energy consumption can be controlled by migrating VMs from underutilized hosts to others so that less utilized ones can be turned off. Migration of some VMs from overloaded hosts is also required for avoiding SLA violations. Migrations are time consuming and create overhead which may be harmful for SLOs. It increases the load on both the PMs involved in migrating a VM and an extra burden on the network. Moreover, VM becomes less responsive during the migration process. Therefore, number of migrations should be of reasonable amount and avoided as much as possible [56].

Title: Energy efficient allocation of virtual machines in cloud data centers

Author: Anton Beloglazov and RajkumarBuyya

Beloglazov et al. gave two step proposal to efficiently allocate the VMs. In the first step, new requests for VM provisioning are allowed and VMs placed on hosts, and in next step current allocation setup of VMs is optimized. The first part is somewhat like a bin packing problem with variable bin sizes and prices. As a solution, modification of the Best Fit Decreasing (BFD) algorithm is applied. In MBFD, VMs are sorted in decreasing order of CPU utilization and then VM is allocated to a host which shows minimum increase in power consumption after allocation. This gives a chance to choose the most efficient one with respect to power. The complexity of the algorithm is $n \cdot m$, where n is

the number of VMs. Optimization part of VM consolidation is further carried out in two steps: in first part VMs are selected to migrate and then chosen VMs are allocated to the host based on MBFD algorithm. The energy consumption is less with respect to the reliable QoS. The proposed approach does not follow the strict SLAs between the service provider and user under the dynamic workload[57].

Title: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing

Author: Anton Beloglazov, Jema lAbawajy, and Rajkumar Buyya

In this research work, Beloglazov et al. proposed a complete solution for reducing power consumption. The main concept was to remove VMs from the less utilized Physical Machines so that these hosts can be turned off and also select some VMs from overloaded hosts to be removed which will reduce the energy consumption. Secondly, a physical machine is to be chosen for accommodating the migrated VM.Selecting a PM is a version of the bin-packing problem in which size of bin is compared with capacity of a PM and also the prices in same way. For bin-packing problem, the Modified Best Fit Decreasing (MBFD) heuristic was used in which VMs were sorted in decreasing order of load and then allocated to the PMs. Several heuristics were used by them to select VMs for migration like Minimization of Migrations (MM) policy, Highest Potential growth(HPG) policy and the Random Choice (RC) policy. To evaluate the presented heuristics, they simulated 100 PMs in CloudSim platform and compared with the Non-Power Aware (NPA) method, DVFS, and a Single-Threshold (ST) VM selection algorithm. The policies were compared with respect to power consumption, number of VM migrations and SLA violations. MM proved to be the best among all presented [59].

Title: Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints

Author: Anton Beloglazov and RajkumarBuyya

Performance of the system is degraded if the server machines are overloaded because the resources which are available are not enough. Mostly, all the researchers have proposed heuristic based solutions which rely on historical data for detecting overloaded hosts.

Heuristic based methods have a disadvantage that they give sub-optimal results. Beloglazov et al have given a solution to host overload detection problem by maximizing the mean inter-migration time keeping in mind the specified SLA based on a Markov chain model. Multi size Sliding Window workload estimation technique is also proposed to heuristically adapt the algorithm to unknown non-stationary workloads. It is quite probable that when the resources are utilized at max, the applications are more prone to lack of resources and performance degradation. To address this problem, most of the schemes for dynamic VM consolidation apply either heuristic-based technique, such as static utilization thresholds. The mean inter-migration time is reduced of the VM migration but the number of migration of VM is high which violates the SLAs [60].

Title: Dynamic placement of virtual machines for managing SLA violations

Author: Norman Bobroff, Andrzej Kochut, and Kirk Beaty

Bobroff et al. aimed at reducing the number of SLA violations and also the energy consumption by reducing active PMs. Authors proposed the Measure-Forecast-Remap cycle which means firstly we measure the consumption of VMs, then future demand is predicted on the basis of this measurement and finally a new mapping is found for a VM to corresponding PM. This cycle of three phases is repeated at regular interval of τ (length). The Remapping part is very much similar to bin-packing problem mentioned before. The best part of proposal is the forecast phase which uses time-series analysis [61].

Title: Energy aware consolidation for cloud computing

Authors: ShekharSrikantaiah, AmanKansal, and Feng Zhao

Srikantaiah et al. also focused on reducing power consumption through VM consolidation without violating SLOs. Unlike the other works explained before, they didn't only consider CPU but one more resource i.e. disk. The very important observation by them was that consolidation impacts performance and energy consumption in a highly nontrivial manner. Up to some point, increasing the utilization leads to higher energy efficiency as expected, however, at some point, some resource of the PM saturates, and thus further increase in the utilization leads to performance degradation; since jobs take

longer to complete, the energy consumption per job increases. As a result, energy consumption per job is a U-shaped function of utilization, yielding an optimal level of utilization. The authors propose to aim for this optimal utilization, which should be determined in an offline profiling phase. Afterward, a two-dimensional packing heuristic is used, where the bin sizes correspond to the optimal utilization of the PMs. The heuristic is a variation of WF, aiming at maximizing the remaining free capacity of PMs. This heuristic can be used both for accommodating new VMs and for optimizing the current placement of the VMs [43].

Title: Dynamic resource allocation using virtual machines for cloud computing environment

Authors: Zhen Xiao, Weijia Song, and Qi Chen

Multidimensional optimization of VM placement was also the subject of the work of Xiao et al. from Peking University. Their approach works in four steps: load prediction, hot spot elimination, cold spot elimination, and execution of migrations. For load prediction, an exponentially weighted average of past observations is used; however, weights are different for increasing and decreasing values so that the method reacts quickly if the load is increasing. Hot spots (PMs with load above some threshold in at least one dimension) are handled by greedily choosing VMs to migrate away from them. Cold spots (PMs with load below some threshold in each dimension) are handled only if the average load of all PMs is below some given threshold; in that case, the algorithm tries to find a new host for the VMs on cold spot PMs; if a PM thus becomes empty, it can be switched off. In both hot spot and cold spot elimination, the Skewnessof the PMs is considered: this metric captures how unbalanced the resource load of the PM in the different dimensions is; the algorithm tries to minimize the skewness of the PMs. The proposed algorithm has been tested using both trace-based simulation and real servers. The results demonstrate that the algorithm is very fast and—if the parameters are configured properly—effective in eliminating overloads and consolidating servers[63].

Title: Self adaptive particle swarm optimization for efficient virtual machine provisioning in cloud

Author: R. Jeyarani, N. Nagaveni, R. Vasanth Ram

They proposed a novel Self Adaptive Particle Swarm Optimization(SAPSO) algorithm for solving the VM placement problem. They tried to map a set of VM instances onto a set of physical nodes or servers so that to minimize power consumption and satisfy SLOs. SAPSO is compared with Multi-Strategy Ensemble Particle Swarm Optimization (MEPSO). Results demonstrate that SAPSO is better than MEPSO in heterogeneous and dynamic environment [64].

Title: Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence

Authors: Jeyarani, Rajarathinam, N. Nagaveni, and R. Vasanth Ram

Jeyarani et al. minimized the total power consumption by the data center by proposing a meta-scheduler called Adaptive Power-Aware Virtual Machine Provisioner (APA-VMP). The scheduler works efficiently by scheduling the workload which minimizes energy consumption without violating the SLOs. The scheduler uses swarm intelligence strategy for detecting the need of optimization of VM placement at the right time. The results evidently show the better performance by great amount of reduction in energy consumption [65].

Title: A multi-objective ant colony system algorithm for virtual machine placement in cloud computing

Authors: Yong qiangGao, Haibing Guan, Zhengwei Qi, Yang Hou, and Liang Liu

Gao et al. proposed a solution for virtual machine placement problem by giving a multi-objective algorithm based on ant colony system.The main focus was to minimize utilization of resources as well as power consumption by simultaneously obtaining the Pareto set of non-dominated solutions. The proposed systems performance is evaluated by comparing with existing genetic algorithm, bin packing algorithm and a max–min ant system (MMAS) algorithm. The results prove the effective and efficiency of the algorithm as compared to others [44].

Title: Energy-aware Hierarchical Scheduling of Applications in Large Scale Data Centers

Authors: Gaojin Wen and Jue Hong

Energy consumed by distributed systems has brought up many issues and has become an outstanding question and requires consideration. Among all the existing methods to save energy, energy consumption can be reduced by proper scheduling of the applications and consolidating them to reduce the running servers. However, many scheduling approaches yet did not acknowledge the cost of energy consumption on network devices, which is also contributes to power consumption in data centers. Hierarchical Scheduling Algorithm (HSA) was proposed to curtail the energy consumed by both servers and network devices. In HSA, a Dynamic Maximum Node Sorting (DMNS) method dealt with optimizing the placement of applications on servers. To further lessen the number of working servers, Hierarchical crossing-switch adjustment is used. Results showed that the number of working servers as well as data transfer speed reduced to good extent. The HSA is simple and robust to minimize the energy consumption by effectively scheduling the applications but HSA and DMNS are not suitable for dynamic workload [69].

Title: Energy-aware scheduling for infrastructure clouds

Authors: Knauth, Thomas, and ChristofFetzer

An immediate fix to curtail the power consumption in data centers is to utilize the modes with lower power. To measure the variation in energy consumption due to virtual machine scheduler's simulation was conducted and besides also demonstrated the inability of default schedulers, using optimized scheduler. The customized scheduler has reduced the complete machine uptime by up to 60.1% after using many real simulation scenarios. OptSched optimizes the virtual machine to physical host mapping by utilizing the reservation length. The parameters covered in this study were heterogeneity of data centers and VMs, the long effect of run time distributions and sensitivity to batch requests. The cumulative machine uptime is balanced for heterogeneity of virtual machines but energy consumption is not efficient if the work load is highly dynamic [70].

Title: A New Approach for Dynamic Virtual Machine Consolidation in Cloud Data Centers

Authors: Asyabi, Esmail, and Mohsen Sharifi

The core technology used by cloud is virtualization so as to adequately consolidate the VMs into physical host for better utilization of resources and to save power. A survey done by many show that the average utilization of servers is still less than expected. A new concept was proposed for this dynamic consolidation of VMs by a dynamic programming algorithm which chooses the VMs from an over utilized host taking into consideration the overhead caused by migrating a VM. Since, all VMs are attached to a storage area network (SAN), the cost of live migration of a VM is decided by its memory imprint. Therefore, time taken by a VM to migrate is calculated by dividing the memory size of VM by network bandwidth. As a result, cost of migration is measured by memory size of the VM. Thus, while selecting the one with less memory size is the best. The cost based approach of VM migration minimizes the power consumption cost of the service provider but when the workload is variable with respect the application, the approach is failed to meet the SLAs [71].

Title: Dynamic Consolidation of Virtual Machines with Multi-Agent System

Authors: EshaBarlaskar and Y. Jayanta Singh

The large-scale data centers contain thousands of servers which consume large amount of electrical power leading to high operating costs. Therefore, to curtail this cost of power the cloud providers need to optimize resource usage effectively by consolidating VMs efficiently in order to improve energy efficiency. The problem of VM consolidation is divided into four sub-problems: physical host overload detection; host under-load detection; VM selection and VM placement. Each of the sub parts work together to optimize the trade off between energy and QoS. For dynamic consolidation of VMs, a new multi-agent system (MAS) was proposed to make the cloud system smarter by blending the five traits of multi agent systems which are ubiquity, intelligence, delegation, interconnection, and human orientation. MASs provide the cloud systems intelligent and insightful based software which can help in effective and better system. The proposed method has significantly reduced energy consumption and also kept constant with the objectives of the Service Level Agreements (SLA).The number of VM

migration and energy consumption is effectively minimized but when the workload is variable with respect the application, the approach is failed to meet the SLAs [72].

Table 2: Literature survey(VM Allocation)

SN	Paper	Resources	Energy	Placement	SLA	Other
1.	Beloglazov et al.[2012]	CPU	Switch off and Dynamic Power	Initial and re-optimization both	Soft	Different PM capacity and Migration
2.	Beloglazov and Buyya [2012]	CPU	Switch off and Dynamic Power	Initial and re-optimization both	Soft	Different PM Capacity, migration with cost
3.	Beloglazov and Buyya [2013]	CPU	-		Soft	Migration and Load prediction
4.	Biran et al. [2012]	CPU and others	-	Initial Placement	-	Different PM capacity and Data transfer
5.	Shi et al.[2013]	CPU and others	Switch off	Only Re-optimization	-	Different PM Capacity, migration with cost
6.	Song et al.[2014]	CPU and others	Switch off	Only Re-optimization	Soft	Migration with cost and load prediction
7.	Srikantaiah et al. [2009]	CPU and others	Switch off and Dynamic power	Initial and Reoptimization	User-level	Migration
8.	Tomas and Tordsson [2014]	CPU and others	-	Initial placement	Soft	Different PMs and load Prediction
9.	Xiao et al. [2013]	CPU and others	Switch off	Reoptimization	Soft	Migration with cost and Load Prediction
10.	He et al.[2012]	CPU cores and other resources	-	Reoptimization	-	Migration

CHAPTER 3

Problem description

3.1. Problem Description

Cloud computing service is a collection of virtual data centers with high optimization which consist of not only software but also hardware and other resources. Companies and other organizations in need of any resource can use resources through pay-per-use model by simply connecting to the cloud. This curtails down their capital expenditure on extra resources at premises. To cope up with growing demand of computational power for high performance applications, companies need large scale data centers which are the core part of system. However, these data centers devour colossal amount of electrical power which has exceeded the cost of actual infrastructure. To make maximum profit, saving operational cost is preferred over performance. People have begun to pay more attention to energy consumption rather than only considering performance.

Many different applications are run at the same data center which contains many heterogeneous servers and network devices. To keep these applications isolated and exploit features of cloud like elasticity, flexibility and reliability, cloud uses virtualization technology. Virtual machines(VMs) are the basic blocks of resources which are provided to customers either directly or indirectly through the provisioned applications.[8]

To save energy in data center best way is to efficiently utilize the resources i.e the VMs. VMs are consolidated to minimum number of hosts and idle hosts are switched off or put in other mode of operation like sleep mode. However, sometimes VM consolidation becomes too combative which may overload hosts and violate SLOs fixed in Service Level Agreement (SLA). Hence, there is a trade off between the QOS and energy consumption which is optimized by allocating VMs efficiently.

3.2. Problem statement

To improve power efficiency and performance of cloud using elastic VM allocation algorithm in IaaS cloud.

3.3. Proposed Solution idea

Cloud data centers can be power efficient and cost effective by minimizing the resource utilization. Allocation of VMs to physical resources i.e. hosts should be optimized so as to balance between the performance requirements and power efficiency. Virtualization and VM migration are the core part of this optimization process. So, proposed solution is to design a VM allocation algorithm for minimizing the resource utilization and reducing the number of VM migrations.

3.4. Methodology

Automatic policy will be used for designing an efficient algorithm. IaaS clouds have an elasticity controller, which is responsible for converting the user requirements to actions provided by IaaS clouds. The controllers use monitoring data from applications and make decisions on whether or not there sources must be scaled. Automatic policy of elasticity i.e. auto-scaling will be used for making these decisions. Optimal online deterministic algorithms will be used for dynamic VM consolidation problems.

CHAPTER 4

Analysis of the Existing Algorithms

In reference to the work shown by Beloglazov et al for the problem of dynamic VM allocation some algorithms were presented in [56].

4.1 Methodology

To deal with dynamic VM consolidation they gave some heuristics for analyzing the old pattern of usage of resources by VMs. The complete problem of dynamic VM consolidation was resolved into these:

1. Firstly to find out when the host is overloaded and needs VMs to be migrated from it.
2. Next all the under utilized hosts are found so as to migrate all the VMs from it.
3. From overloaded hosts, VMs are selected for migration.
4. Last is to determine the new VM-to-PM mapping for all the VMs selected for migrations in above steps.

In the starting, algorithm 1(VM Placement Optimization) was run to find out the overloaded hosts, underutilized hosts and the VMs selected for migration. Initially, it uses heuristics to check whether the host is overloaded or not. The policies used for this are discussed in section 3.5.1. Once the host is detected as overloaded, some VM selection policy is used for selecting VMs from that particular host for migration. The VM selection policies are discussed in section 3.5.2. After selecting the VMs, VM placement algorithm is invoked to return a new migration map i.e a new VM-to- PM mapping. Now, the underutilized hosts are checked in second part of the algorithm. In these, all the VMs are selected for migration and new placement is determined for these. At last, the algorithm returns new mapping for the VMs with complexity 2n, where n is the number of hosts.

Algorithm 1: VM placement Optimization

- **Input:** hostList**Output:** migrationMap
- foreach host in hostList do
- if isHostOverloaded (host) then
- vmsToMigrate.add(getVmsToMigrateFromOverloadedHost(host))
- migrationMap.add(getNewVmPlacement(vmsToMigrate))
- vmsToMigrate.clear()
- foreach host in hostList do
- if isHostUnderloaded (host) then
- vmsToMigrate.add(host.getVmList())
- migrationMap.add(getNewVmPlacement(vmsToMigrate))
- return migrationMap

4.1.1. Host Overloading Detection

To detect overloaded hosts we need some criteria on the basis of which hosts can be considered overloaded. Many try to use static threshold values of utilization for deciding it. Some lower and upper threshold values are set, if the CPU utilization of a particular host falls below the lower value then it is underloaded whereas if the CPU utilization is above the upper threshold then overloaded. For comparison purposes, static value for threshold is also set through **Threshold (THR)** algorithm.

However, in cloud like dynamic environment where workloads are so unpredictable using such fixed values of threshold is not suitable. So the system should be capable enough to adjust the threshold values automatically so as to perform better. To auto-adjust the threshold values some techniques have been proposed which use heuristics based on the statistical analysis of the old patterns of resource usage by VMs.

The main focus is on upper value of threshold which changes with deviation in CPU utilization. Higher is the deviation in CPU utilization, more the chances of SLA violation as utilization will reach 100%. So, lower should be the value of upper threshold when higher is the deviation in CPU utilization. Some novel techniques used are Mean Absolute Deviation (MAD), Interquartile Range(IQR) and Local Regression(LR).

4.1.2 VM Selection

After determining the overloaded hosts, next is to choose VMs for migration from this particular host. It is an iterative process. After applying selection policy host is again verified for overloaded situation. If found overloaded, VM selection policy is again applied to select another VM for migration. This process continues till the host under detection is not overloaded. For VM selection, policies used are Minimum Migration time policy (MMT), Random Choice (RC) and Maximum Correlation Policy (MC).

4.1.3. VM Placement

Finding a new placement for VMs is similar to bin-packing problem which has different sizes and prices of bin. Physical hosts are represented as bins with CPU capacity as bin size and power consumption of hosts as bin prices. VMs are the items to be allocated to the bins.A modified version of BFD algorithm is proposed for effective power consumption i.e. Power Aware Best Fit Decreasing (PABFD. All the VMs selected for migration are sorted in decreasing order of their CPU utilization and then VM is allocated to that host which gives minimum increase in power consumption after the allocation of VM. Algorithm 2 shows the modified version of BFD with complexity nm, where n is the number of nodes and m is the number of VMs that have to be allocated.

Algorithm 2: Power Aware Best Fit Decreasing (PABFD)

- **Input:** hostList, vmList**Output:** allocation of VMs
- vmList.sortDecreasingUtilization()
- **foreach**vm in vmList do
- minPower MAX
- allocatedHost NULL
- **foreach** host in hostList do
- **if** host has enough resources for vm then
- power estimatePower(host, vm)
- **if** power <minPower then
- allocatedHost host
- minPower power
- **if**allocatedHost 6= NULL then
- allocation.add(vm, allocatedHost)
- **return** allocation

4.1.4. Host Underloading Detection

In this section, underloaded hosts are detected by simple way. Firstly, the overloaded hosts are determined and also the VMs for migration are allocated to the final hosts. Next is to find a host with less utilization in comparison with other hosts so that all the VMs from this host can be migrated to other hosts without overloading the other. If successful in finding such one, then VMs are migrated to other host and it is turned off or switched to sleep mode saving the power. If not able to migrate all the VMs, then host is kept active and not turned off. This is again an iterative process.

4.2 Experimental Setup

The CloudSim toolkit [67] has been selected as platform for simulation. CloudSim 3.0.3 version is used. Data centers as physical nodes have been simulated half of which are HP ProLiant ML110 G4 servers, and the other half consists of HP ProLiant ML110 G5 servers.Test cases have been created changing the number of Hosts and VMs. For each particular test case Power Consumption, Number of VMmigrations and Average SLA Violations have been calculated by using combinations of host overloaded detection algorithms and VM Selection Algorithms.

Combinations used are:

- **IQR-MMT**(Interquartile Rangeas host overloading detection algorithm and Minimum Migration Time Policy as VM selection policy)
- **LR-MMT**(Local Regression and Minimum migration Time policy)
- **MAD-MMT**(Mean Absolute Deviation and Minimum migration Time policy)
- **THR-MMT**(static Threshold and Minimum migration Time policy)
- **LR-RC**(Local Regression and Random Choice Policy)

4.3. Performance Metrics

The efficiency of the algorithms are compared using some metrics so as to evaluate their performance. Metrics used are total Energy Consumption, SLA violations and Number of VM migrations. Energy Consumption in kWh shows the consumption of energy by the server when applications are run on it. It is calculated according to the values assumed and mentioned in table 3. SLA violations and Vm migrations are managed by the VMM while consolidating VMs.

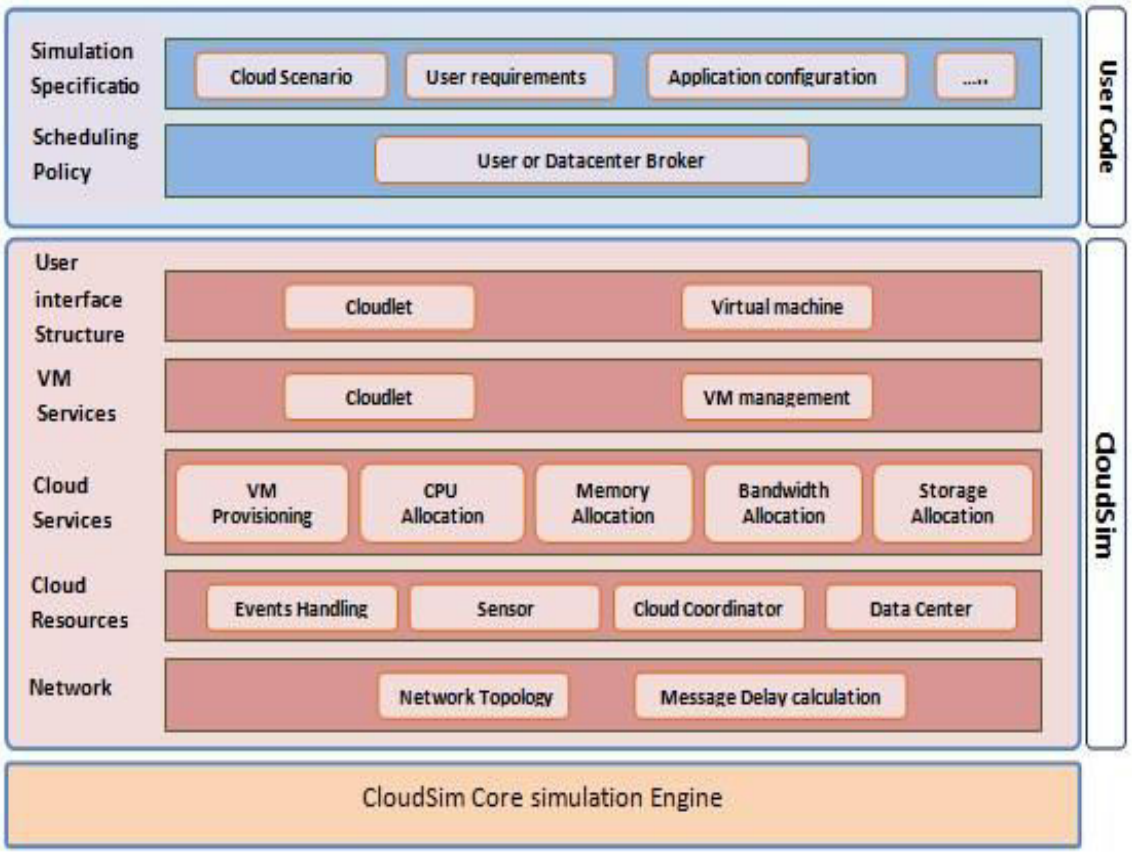

Figure 7: CloudSim Architecture

Table 3: Power consumption by the selected servers at different load levels in Watts

CPU UTILIZATION IN PERCENTAGE (%)

SERVER	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiantG4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

4.4. Test Cases for Simulation and Results

Table 4: Test Cases

Test Case	Number of Hosts	Number of Virtual machines(VMs)
1	50	50
2	40	40
3	40	50
4	40	60
5	60	40
6	50	100
7	100	50

In table 1 tests cases are built for evaluating the algorithms. These are built by varying the number of hosts and number of Virtual Machines (VMs). For each test case, each algorithm is run and values are recorded for power consumption, SLA violations and Number of VM Migrations. The simulation results are evaluated with the help of graphs and conclusions are given.

Table 5: Results of power consumption by the policies

Test case	IQR-MMT	LR-MMT	MAD-MMT	THR-MMT	LR-RC
1	47.85	35.37	45.61	41.81	34.41
2	37.66	27.98	35.60	33.29	27.14
3	45.84	34.85	43.31	40.73	33.94
4	52.62	40.69	49.80	45.86	39.75
5	38.23	27.79	36.11	33.31	26.83
6	84.54	75.62	80.34	74.04	63.01
7	48.68	41.76	46.42	42.12	34.24

Table 6: Results showing Number of VM migrations by the policies

Test case	IQR-MMT	LR-MMT	MAD-MMT	THR-MMT	LR-RC
1	5502	2872	5265	4839	2434
2	4274	2344	4086	3912	1819
3	5155	3000	5120	4743	2191
4	6457	3496	6534	6258	2817
5	4403	2235	4252	3889	1708
6	10109	5625	10435	11168	4670
7	5492	3363	5289	4778	2417

Table 7: Results showing SLA Violation rate by the policies

Test case	IQR-MMT	LR-MMT	MAD-MMT	THR-MMT	LR-RC
1	10.44%	12.89%	10.91%	12.81%	13.26%
2	10.67%	12.98%	10.93%	12.59%	12.75%
3	10.62%	13.30%	11.24%	12.59%	12.99%
4	10.95%	12.99%	11.54%	13.4%	13.49%
5	10.48%	12.85%	10.74%	12.66%	12.71%
6	11.57%	11.26%	12.36%	14.72%	13.68%
7	10.54%	10.73%	11.12%	12.81%	13.00%

4.5. Analysis of Results

Figure 8 represent Algorithm combinations showing comparison on the basis of Performance metrics

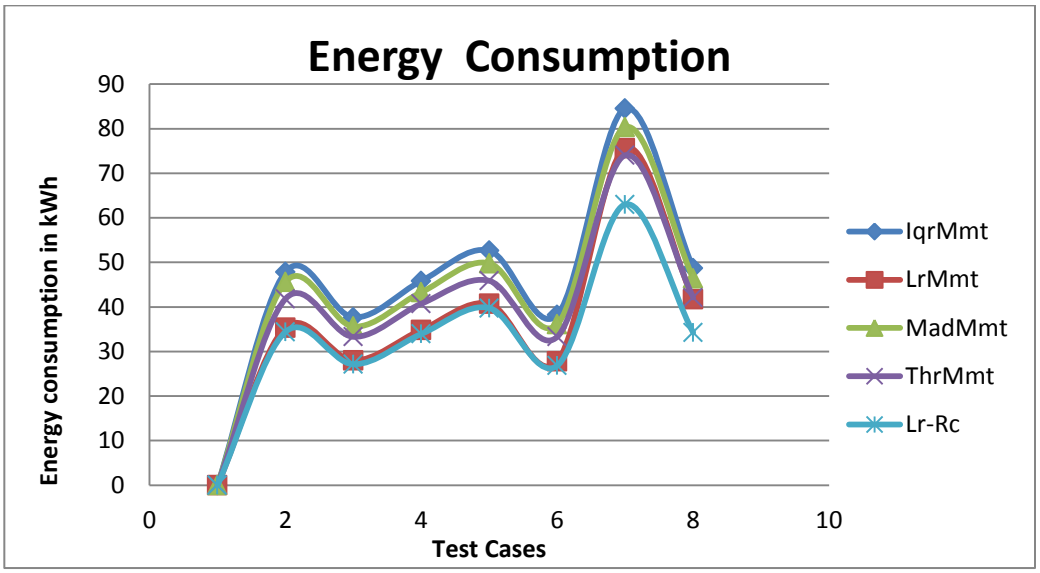

Figure 8(a) : Energy Consumption

In figure 8(a) the simulation results are projected of the algorithms for energy consumption metric. For each test case, algorithms are run on CloudSim and values are recorded for power consumption in kWh. Local Regression algorithms show better results by consuming less power than others. There is not much difference in results of LR-RC and LR-MMT. IQR-MMT consumes the maximum power.

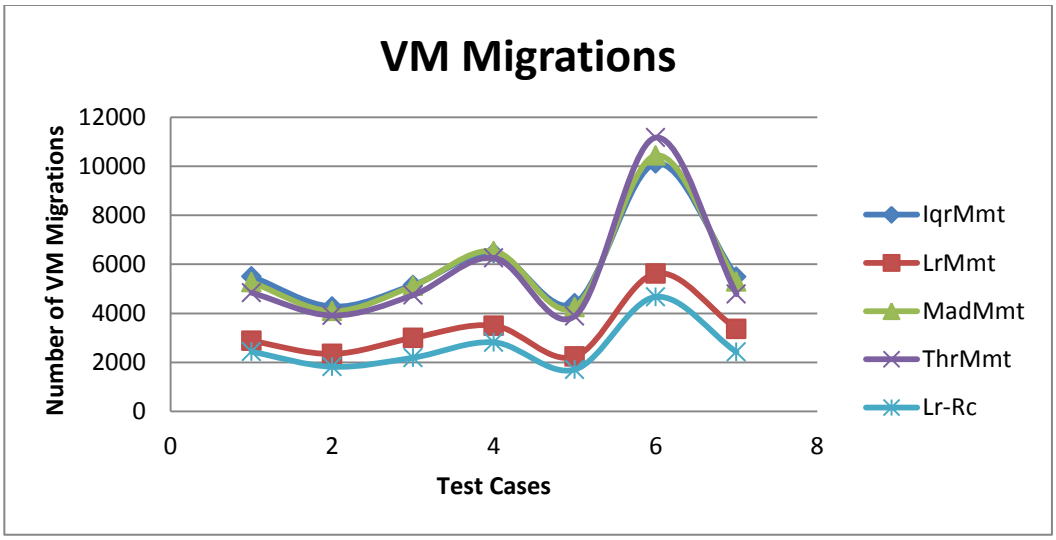

Figure 8(b): VM Migrations

In figure 8(b) simulation results of the algorithms are projected for VM migrations performance metric. For each test case, algorithms are run on CloudSim and values are recorded for number of VM migrations. Local Regression algorithms show better results by less VM migrations. LR-RC slightly gave lesser number of migrations than LR-MMT. There is not much difference in results of IQR-MMT, MAD-MMT and THR-MMT.

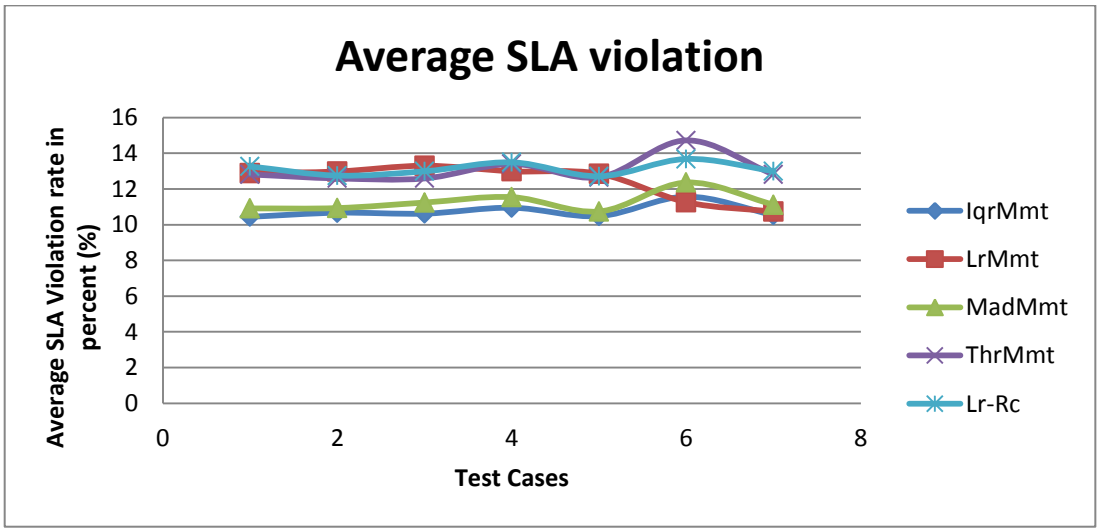

Figure 8(c): SLA Violation Metric

In figure 8(c) simulation results of algorithms are projected of average SLA violation metric. For each test case, algorithms are run on CloudSim and values are recorded for

average SLA violations. There is no statistically significant difference between the results of MAD-MMT and IQR-MMT. LR-MMT gives more SLA violations than others for some test cases. THR-MMT was constant in bad results.

According to the results of pairwise combination of all policies, it can be concluded that there is statistically no significant difference between the LR-RC and LR-MMT values. However, there is a statistically significant difference between Local regression algorithms and the other algorithms.LR-MMT and LR-RC outperform the other policies in power consumption and Number of VM migrations by a good mark. But in Average SLA violations IQR-MMT gives the best results.

CHAPTER 5

PROPOSED MODEL

In the proposed work, the main framework considered is an Infrastructure as a Service (IaaS) environment. It represents N number of different physical hosts. Each host is portrayed by the CPU performance defined in MIPS, RAM and transfer speed. Virtual machines (VMs) are hosted on the physical machines on-demand of the users. Customers submit their requirements specifying MIPS, bandwidth, number of processors, etc. Different users hosting various types of applications use resources simultaneously. For this transparency, cloud system uses virtualization technology.

The system model shows software layer of the framework which consists of global resource manager, local managers and VMs hosted on physical machines. Each node has one local manager as a part of VMM which keeps a continuous check on resource requirements of VMs and also makes decision regarding selection and migration of VMs at time of overload. The global manager is a part of master node which is in contact with all the local managers and collects information so as to optimize the resource utilization and Service level objectives. It gives the orders regarding VM placement. Actual migration and resizing of VMs is performed by VMM.

5.1 Proposed System

An adaptive heuristics is used for dynamic consolidation of VMs based on an analysis of previous data from the resource usage by VMs. The proposed algorithm significantly reduces energy consumption, while ensuring a high level of adherence to the Service Level Agreement (SLAs). The proposed algorithm performs dynamic consolidation of VMs at run-time on the basis of current utilization of resources which may involve live VM migration, changing the mode of unused host to lower power mode so that power can be saved. The system efficiently handles firm SLA and multi-core CPU architectures. The algorithm adapts the behaviour with respect to observations and characteristics of VMs.

5.1.1 Advantages

- The energy consumption is effectively minimized by better VM placement.
- Live migration scheme used in the approach strictly follows the SLAs between the service provider and user. Migrations are minimized using self-healing.

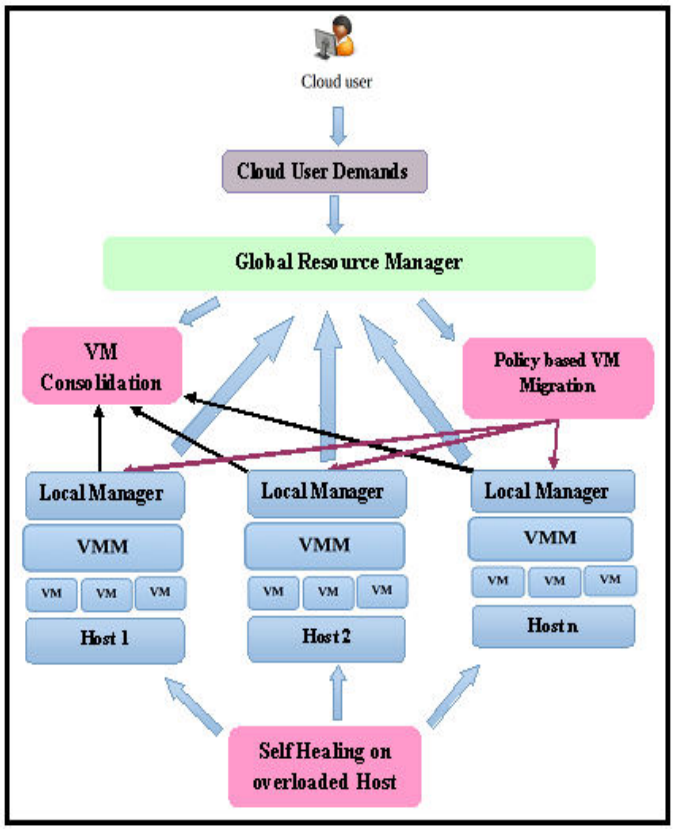

Figure 9: Proposed system model

5.1.2 Main Contribution

Self Healing in Overloaded Host

With respect to the proposed work mentioned above, the live migration of virtual machine from the overloaded host is not performed at the first instance. Instead of that, for each of the over utilized host self healing is performed. All virtual machines utilization is analyzed in each overloaded host, then add the MIPS to the more utilized VM and remove MIPS from less utilized VM or if the host has some free PE or MIPS that can be added to more utilized VM. So, overloaded host adjusts VM parameters using self healing and balance the utilization without violating the SLAs. If self healing is not

possible, then proposed approach performs the normal VM live migration algorithm where the migrating VMs are selected from the overloaded host and these VMs are migrated to the other host using some policy. For all underutilized hosts, all the VMs are migrated to the safe host.

5.2 BLOCK DIAGRAM

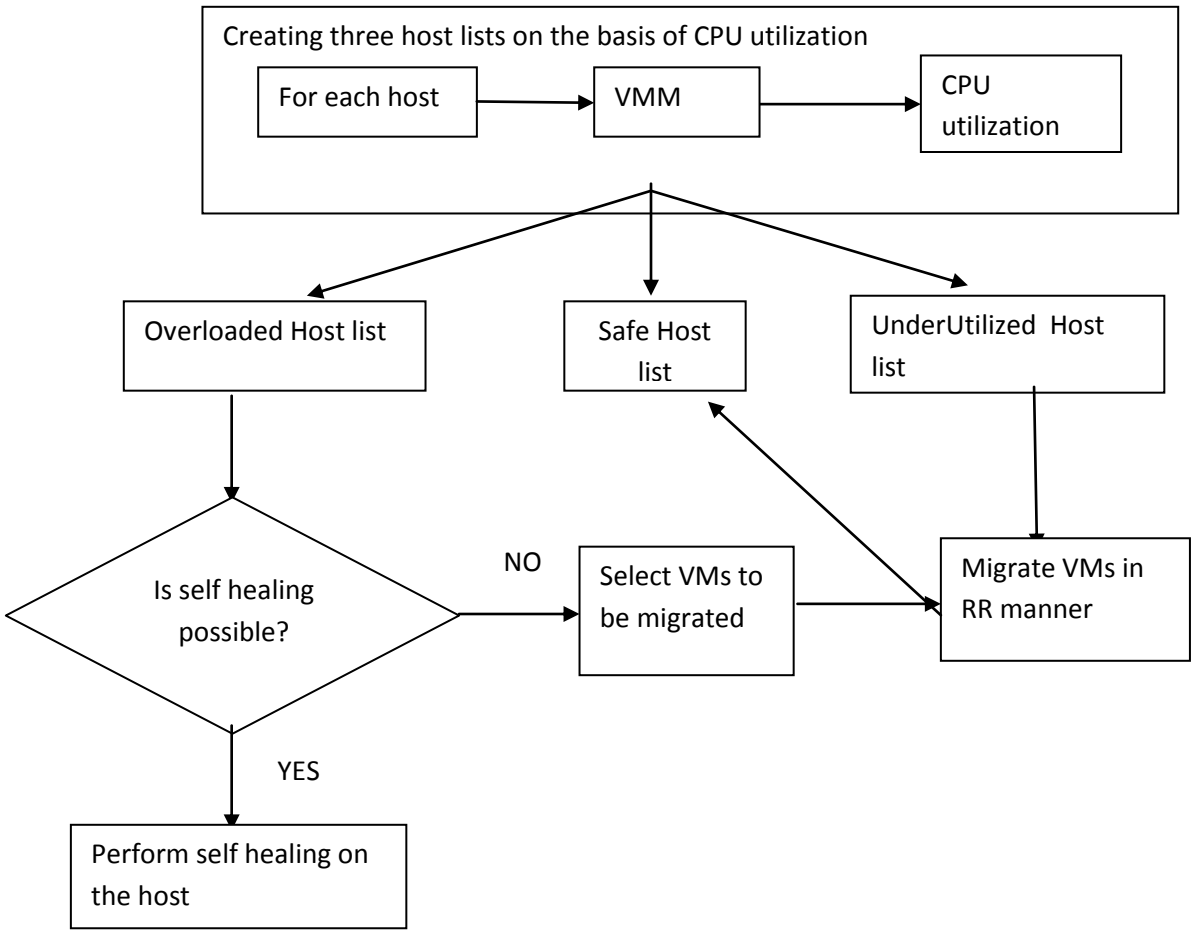

Figure 10: Block diagram of proposed system

5.3 Modules:

1. Cloud user requirements
2. Cloudlet Execution
3. Under Utilized and Over Utilized Host Detection
4. Self Healing on Over Utilized Host
5. VM consolidation from Under Utilized Host

Modules Description

1. Cloud user requirements

Input : The cloud user demands and their credential

Output: Application and VM configuration

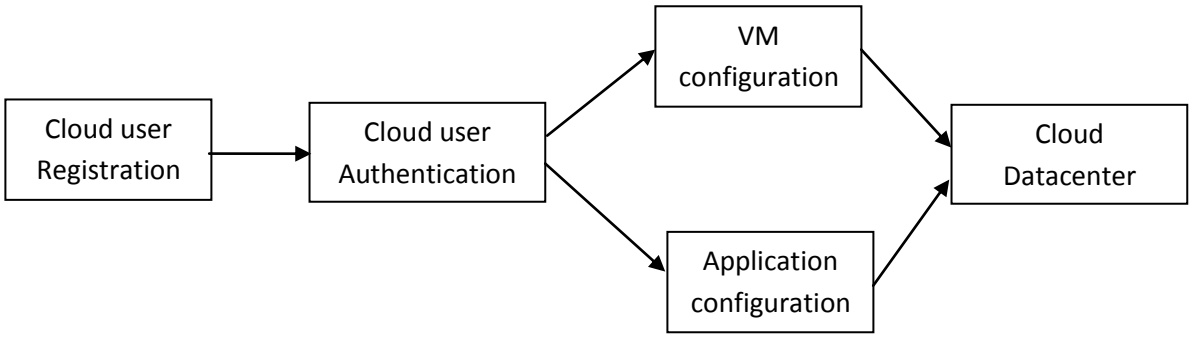

Figure 11(a): Cloud User Requirements

The cloud user sends the requirements, VM configuration and Application configuration. VM configuration such as MIPS, number of PEs, RAM, bandwidth and number of VMs. Application configuration such as number of cloudlets, application length. These requirements are sent to the cloud service provider such that global manager who allocates the resource to the cloud user.

2. Cloudlet Execution

Input : User Application and VM configuration

Output: Created cloudlet in Datacenter and cloudlet execution

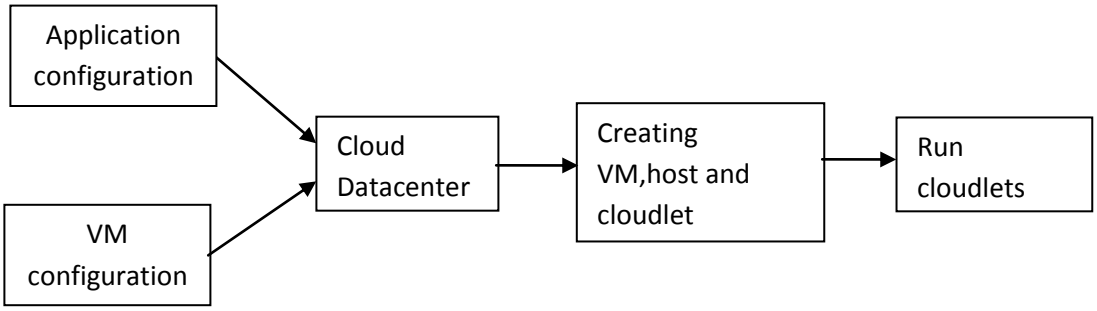

Figure 11(b): Cloudlet Execution

As per the user requirements, the VM instances are created in the hosts according to the VM configurations mentioned. After that, task or cloudlets are created in the cloud datacenter as per application configuration and these cloudlets are scheduled to the virtual machines by broker. After the cloudlet submission, the user tasks are executed on the datacenter.

3. Underutilized and Over Utilized Host Detection

Input: CPU utilization for every host

Output: Under and over utilized host list

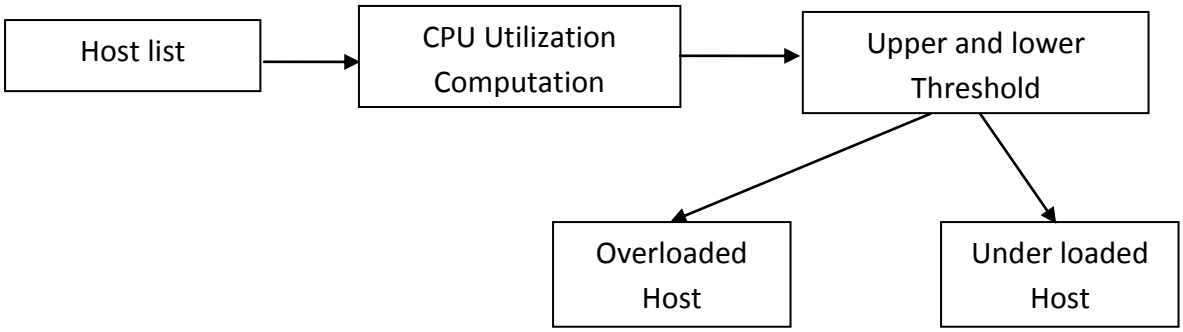

Figure 11(c): Underutilized and Overutilized Host Detection

During the run time, each host's CPU utilization is monitored by the VMM(virtual machine Manager) and these utilization details are sent to local manager present in every host responsible for resizing the virtual machines(VMs) according to the resources needs. The global manager resides on the master node and collects information from the local managers to maintain the overall view of the utilization of resource.

4. Self Healing on Over Utilized Host

Input: Over utilized host list

Output: Self healed host or VM migration list

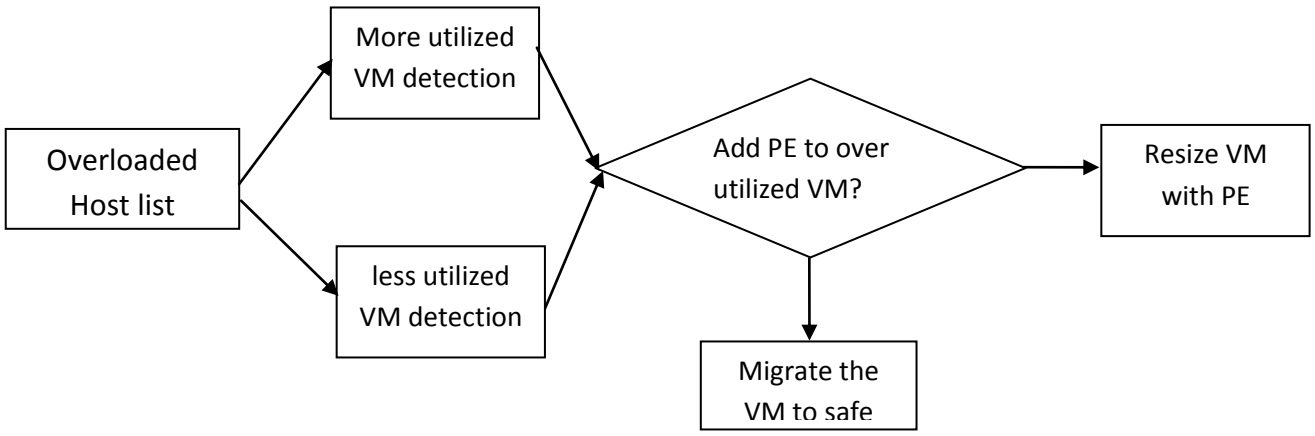

Figure 11(d): Self-Healing on over utilized Host

In this module, each overloaded host is subjected to self-healing. Firstly, VMs are analyzed of that particular host and resized with MIPS or PE. MIPS is transferred from less utilized VM to more utilized VM. This will balance the load, reduce power consumption without violating the SLA and without migrating any VM. If self healing option is not possible, then over utilized VMs are migrated to the safe host using round robin algorithm. Hence the number of VM migration will be reduced if self healing is performed on the host.

5. VM consolidation on Under Utilized Host

Input : Underutilized host list

Output: VM consolidation and live migration of VM

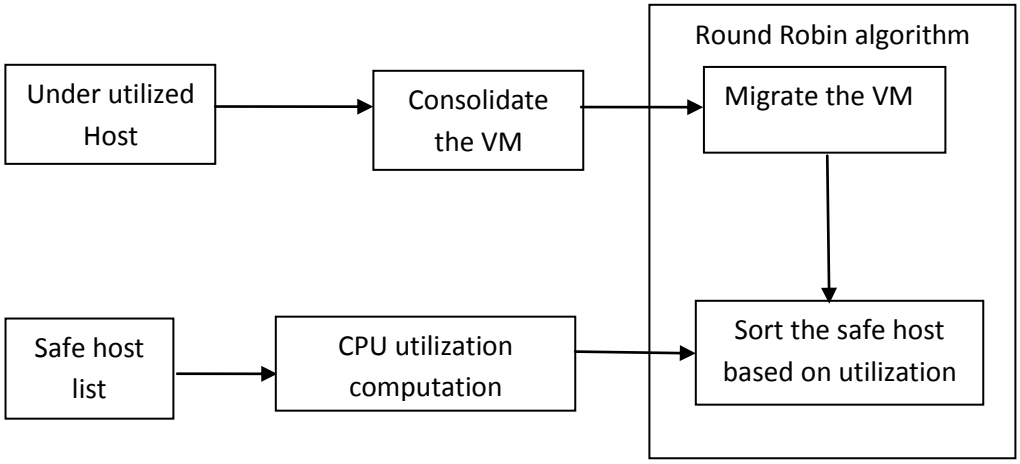

Figure 11(e): VM consolidation on Underutilized host

Each underutilized host undergoes VM consolidation. Here all VMs are migrated to the safe host. This migration is performed using Round Robin Algorithm, where the all safe host are sorted in increasing order according to the current CPU utilization of the host, and the host with less utilization is selected for VM allocation

5.4 Proposed Algorithm

Algorithm 3

STEP 1: Create the user demands such as application configuration and VM configuration
STEP 2: Initialize the CloudSim and create the datacenter, broker, hosts, and VM based on the user demands
STEP 3: Create cloudlets (jobs or applications) for user requirements
STEP 4: Schedule the task on the VM based on VM allocation policy
STEP 5: Start simulation
STEP 6: Calculate the CPU utilization on the every host
STEP 7: Iteration
7a: get the first host in the list
7b: if host CPU utilization is lower than 0.2 then move the host to underutilized host list
7c: if host CPU utilization is greater than 0.8 then move the host to over utilized host list
7d: else mover the host to safe host list
Close the for loop
STEP 8: Iteration over utilized host list get the VM with maximum utilization
8a: get the available MIPS from the host of maximum utilized VM
8b: if MIPS is available then add available MIPS to over utilized VM
8c: else migrate the VM to safe host based on some policy
STEP 9: Iteration for each underutilized host
9a: Consolidate the every VM on overloaded host and move those
9b: VMs to migration list
Close the for loop
STEP 10: Sort the safe host in increasing order based on CPU utilization and migrate all the VMs based policy (migrate the VM with maximum utilization to host with minimum utilization to achieve the balancing)
STEP 11: Run applications
STEP 12: Stop simulation.

The proposed algorithm for VM consolidation with self-healing is given in Algorithm 3. First of all application and VM configurations are setup according to the user requirements. CloudSim is initialized and cloudlets are created. After scheduling the tasks on the VM, simulation is started. Now when the system is working, VMs need to be

consolidated effectively to utilize the resources efficiently. CPU utilization is calculated for each and every host and on the basis of this three hostLists are created, Overutilized, underutilized and safe.

For every host in Overutilizedhostlist, self-healing is applied. If self-healing not possible then VMs are selected for migration and migrated to safe hosts. At last for underutilized hosts all the VMs are migrated to safe hosts.

5.5. Experimental Setup

The CloudSim toolkit [67] has been selected as platform for simulation. CloudSim 3.0.3 version is used. Data centers as physical nodes have been simulated half of which are HP ProLiant ML110 G4 servers, and the other half consists of HP ProLiant ML110 G5 servers. For the existing algorithm as well as for the proposed algorithm Power Consumption, and Number of VM migrations have been calculated by using combinations of host overloaded detection algorithms and VM Selection Algorithms.

Combinations used are:

- **IQR-MMT**(Inter quartile Range as host overloading detection algorithm and Minimum Migration Time Policy as VM selection policy)
- **LR-MMT**(Local Regression and Minimum migration Time policy)
- **MAD-MMT**(Mean Absolute Deviation and Minimum migration Time policy)
- **THR-MMT**(static Threshold and Minimum migration Time policy)
- **LR-RC**(Local Regression and Random Choice Policy)

5.6. Performance Metrics

The efficiency of the algorithm is evaluated using two metrics. Metrics used are total Energy Consumption and Number of VM migrations. Energy Consumption in kWh shows the consumption of energy by the server when applications are run on it. It is calculated according to the values assumed and mentioned in table 8. VM migrations are managed by the VMM while consolidating VMs.

Table 8: Power consumption by the selected servers at different load levels in Watts

	CPU UTILIZATION IN PERCENTAGE (%)										
SERVER	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiantG4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

5.7. Results

According to proposed scheme simulated results have been shown and compared with the existing values of the metrics. Table 8 shows the results of Energy Consumption in kWh of the proposed algorithm and existing system.

Table 9: Energy consumption in kWh by the policies

Policy	**Existing system**	**Proposed system**
IQR-MMT	47.85	38.93
LR-MMT	35.37	36.61
MAD-MMT	45.61	32.65
THR-MMT	41.81	32.24
LR-RS	34.41	28.77

According to the proposed algorithm, above results can be visualized in graph to evaluate the performance of the algorithm in terms of energy consumption. LR-RS policy consumes minimum energy.

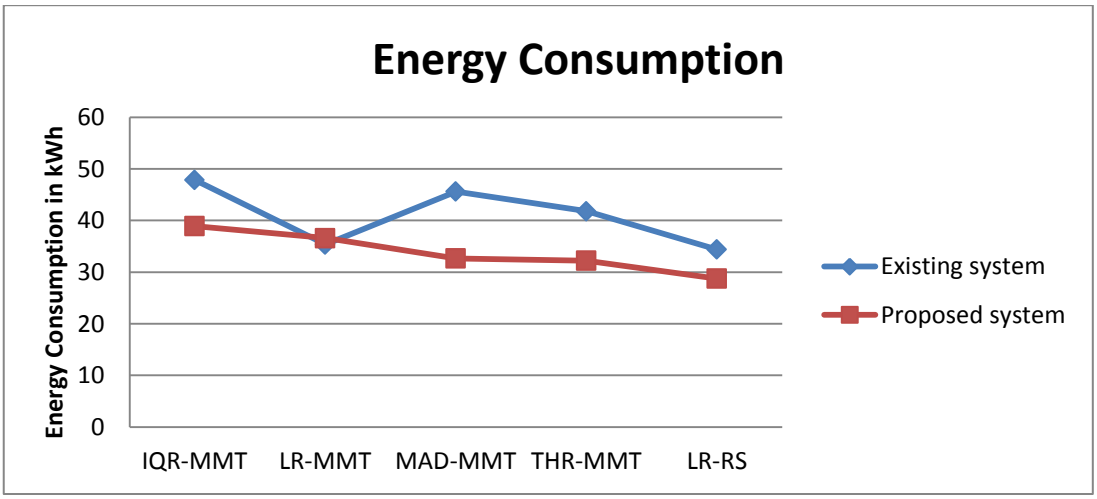

Figure 12: Comparison of policies based on Energy Consumption.

Above Figure 12 shows comparison of policies based on energy consumption using proposed and existing system.

Table 10: Number of VM migrations

Policy	**Existing system**	**Proposed system**
IQR-MMT	5502	3395
LR-MMT	2872	1867
MAD-MMT	5265	3103
THR-MMT	4839	3146
LR-RS	2434	1194

Table 10 shows the results of number of VM migrations of the proposed algorithm and existing system.

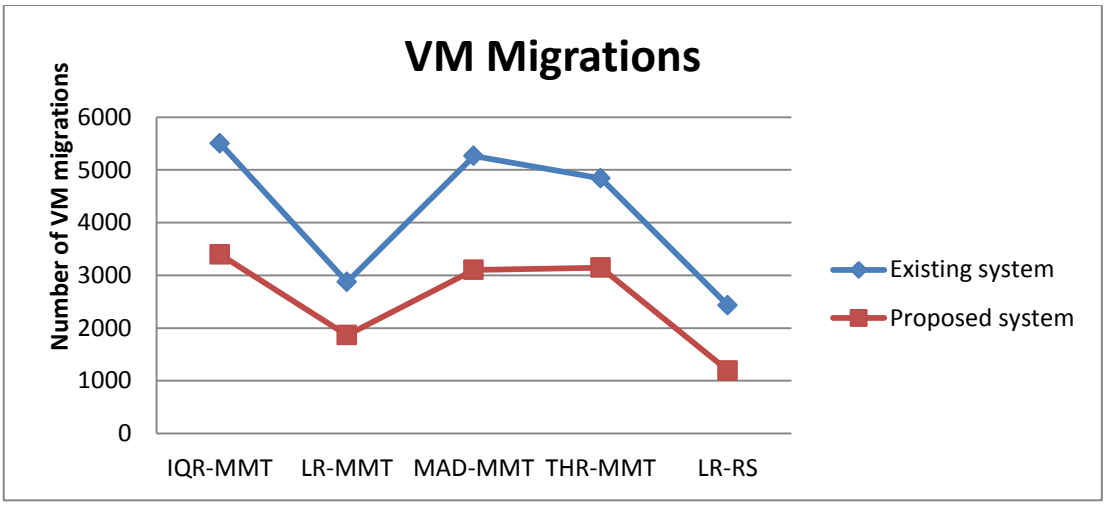

Figure 13: Comparison of policies based on VM Migrations.

Above Figure 13 shows comparison of policies based on number of VM migrations using proposed and existing system.

Conclusion

The results from the proposed algorithm show significant improvement in the parameters. The power consumption has been reduced without increasing the number of VM migrations. The five policies were evaluated for the existing system as well as for the proposed algorithm. LR-MMT(Local Regression and Minimum Migration Time) and LR-RS (Local Regression and Random Choice) give least number of migrations as depicted in figure 3 as well as quite less power consumption in comparison with other policies like IQR-MMT(Inter-Quartile Range and Minimum migration Time), MAD-MMT(Mean Absolute Deviation and Minimum Migration Time) and THR-MMT(Static Threshold and Minimum Migration Time) .

Chapter 6

Conclusion and Future Work

This thesis discusses the issues in cloud computing system while handling the dynamic workload. Large data centers consume large amount of electrical power which is damaging the environment and also increasing the operational cost. So this thesis is all about dealing with these problems.

To improve their profit cloud providers need to apply the power-efficient resource management strategies, such as dynamic consolidation of VMs and switching idle servers to power saving modes. However, such consolidation is not trivial, as it can result in violations of the SLA negotiated between the customers. So there is a trade-off between performance and energy efficiency. Due to the dynamic and elastic nature of cloud resource pool, optimal online deterministic algorithms are used for these problems. Randomized and adaptive algorithms improve their performance. Many novel adaptive heuristics are used that are based on the analysis of historical data of the resource usage for energy and performance efficient dynamic consolidation of VMs.

To start with **Chapter 1** had the detailed introduction to Cloud Computing technology with its evolution. Definition for Cloud Computing provided by NIST is mentioned which further describes the service models, deployment models, five essential characteristics and the various challenges found while working with Cloud Computing. At last problem context was discussed describing the complete scenario of problems found in this context. With respect to these problems, literature survey was shown in **Chapter 2**. Firstly elasticity as a feature was reviewed and VM allocation problem was discovered. VM allocation problem and all its parameters are studied and literature has been reviewed. It was concluded that to reduce power consumption without violating SLA, live VM migration method needs to be used.

The complete survey has been concluded in **Chapter 3** in problem description part. Final Problem statement is also mentioned. In **Chapter 4** existing algorithms have been implemented in CloudSim framework. Test cases were generated and policies were compared. Parameters used for comparison are Power Consumption in kWh, number of SLA violations and number of VM migrations. Policies which were implemented are

IQR-MMT, LR-MMT, MAD-MMT, THR-MMT and LR-RC. It was concluded that LR-MMT and LR-RC outperform the other policies in power consumption and number of VM migrations. Last in the **Chapter 5** is the proposed system.

To minimize the energy consumption of the cloud service providers we have proposed energy-efficient resource management strategies, such as self healing on the overloaded host and dynamic consolidation of VMs. Hence the number of migration are reduced, there waiting time of the running application on the VM is avoided. Hence proposed strategy strictly follows the SLAs assured between the cloud users and cloud service providers. If the self healing is not possible the, VM migration is performed on the basis of round robin algorithm to reduce the host overload. The VMs consolidation is also effectively done by proposed strategy and VMs placement is estimated based on the current resources requirements of the every VMs running on the host. The proposed system is implemented using CloudSim simulation tool and results are compared with the existing algorithms. Results demonstrate that proposed system showed better results for power consumption as well as number of VM migrations.

Research Publications

From this thesis work I have published two Research papers. One is the conference paper and other is a journal paper. The references to these papers are:

1. Kshitiza Vasudeva, Punit Gupta," A Survey on Elastic Resource Allocation Algorithm for Cloud Infrastructure" in International Conference on International conference on Innovation and Challenges in Cyber Security (ICICCS),2016 (Published)
2. Kshitiza Vasudeva, S.P.Ghrera," Adaptive Heuristics with Self-Healing for Efficient Dynamic Consolidation of Virtual Machines in cloud data-centers", in International Journal of Trends in Engineering and Technology(IJLTET), 2016 (Accepted)

References

1. G. Galante and L. C. E. de Bona," A survey on cloud computing elasticity," in IEEE/ACM Fifth International Conference on Utility and Cloud Computing (UCC) , pp. 263-270, 2012.
2. Jamshidi, Pooyan, Aakash Ahmad, and Claus Pahl, "Autonomic resource provisioning for cloud-based software", Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, ACM, pp. 95-104, 2014.
3. E. Caron, F. Desprez, and A. Muresan, "Forecasting for Grid and Cloud Computing On Demand Resources Based on Pattern Matching," Cloud Computing Technology and Science (CloudCom), pp. 456-463, 2010.
4. Ali-Eldin, Ahmed, Johan Tordsson, and Erik Elmroth. "An adaptive hybrid elasticity controller for cloud infrastructures," Network Operations and Management Symposium (NOMS), IEEE, pp. 204-212, 2012
5. M.UthayaBanu, K.Saravanan, "Optimizing the Cost for Resource Subscription Policy in IaaSCloud",International Journal of Engineering Trends and Technology (IJETT),Vol. 6 No. 5, pp. 296-301, 2013
6. R.Han, Li Guo, Moustafa M. Ghanem and YikeGuo, "Lightweight resource scaling for cloud applications."IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp. 644-651,2012.
7. Kupferman, Jonathan, Jeff Silverman,PatricioJara, and Jeff Browne, "Scaling into the cloud," CS270-advanced operating systems, pp. 1-8, 2009.
8. G. Copil, D. Moldovan, H.L.Truong and S. Dustdar, "SYBL: An Extensible Language for Controlling Elasticity in Cloud Applications", in IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, IEEE, pp. 112-119, May 2013
9. H.Nguyen, Z.Shen, X.Gu, S.Subbiah, J.Wilkes, "AGILE: Elastic Distributed Resource Scaling for Infrastructure-as-a-Service", in Proceedings of the 10th International Conference on Autonomic Computing (ICAC), pp. 69-82, 2013
10. Ajila A. Samuel and Bankole A. Akindele ,"Proactive Prediction Models for Web Application Resource Provisioning in the Cloud", in Transition from observation to knowledge, pp. 17-35, 2014
11. Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/.
12. "GoGrid." [Online]. Available: http://www.gogrid.com/
13. "Rackspace." [Online]. Available: http://www.rackspace.com/

14. “Microsoft Azure.” [Online]. Available: http://www.windowsazure.com/
15. I. Neamtiu, “Elastic executions from inelastic programs,” in Proceedings of the 6th Intl. Symposium on Software Engineering for Adaptive and Self-Managing Systems, ser. SEAMS ACM, pp. 178–183,2011.
16. D. Rajan, A. Canino, J. A. Izaguirre, and D. Thain, “Converting a high performance application to an elastic cloud application,” Proceedings of the 3rd International Conference on Cloud Computing Technology and Science, CLOUDCOM IEEE, pp. 383–390, 2011.
17. S. Meng, L. Liu, and V. Soundararajan, “Tide: achieving self-scaling in virtualized datacenter management middleware,” in Proceedings of the 11th International Middleware Conference, ACM, pp. 17–22, 2010.
18. R. N. Calheiros, C. Vecchiola, D. Karunamoorthy, and R. Buyya, “The aneka platform and qos-driven resource provisioning for elastic applications on hybrid clouds,” in Future Generation Computer Systems, vol. 28, no. 6, pp. 861–870, June 2011.
19. J. O. Fit´o, I. G. Presa, and J. G. Fern´andez, “Sla-driven elastic cloud hosting provider,” in Proceedings of the 18th EuromicroConferenceon Parallel, Distributed and Network-based Processing, ser. IEEE, pp. 111–118, 2010.
20. Dutreilh, Xavier, Nicolas Rivierre, Aurlien Moreau, Jacques Malenfant, and Isis Truck, "From data center resource allocation to control theory and back" IEEE 3rd International Conference on Cloud Computing (CLOUD), IEEE, pp. 410-417, 2010.
21. Maurer, Michael, IvonaBrandic, and RizosSakellariou, "Enacting SLAs in clouds using rules", in Euro-Par 2011 Parallel Processing, Springer Berlin Heidelberg, pp. 455-466, 2011.
22. Lim, Harold C., ShivnathBabu, Jeffrey S. Chase, and Sujay S. Parekh,"Automated control in cloud computing: challenges and opportunities", in Proceedings of the 1st workshop on Automated control for datacenters and clouds, ACM, pp. 13-18, 2009.
23. Marshall, Paul, Kate Keahey, and Tim Freeman, "Elastic site: Using clouds to elastically extend site resources," in Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, IEEE Computer Society, pp. 43-52, 2010.
24. Caron, Eddy, Luis Rodero-Merino, FrédéricDesprez, and Adrian Muresan. "Auto-scaling, load balancing and monitoring in commercial and open-source clouds." RR- 7857, INRIA, pp.27, 2012.
25. W. Dawoud, I. Takouna, and C. Meinel, “Elastic vm for cloud resources provisioning optimization,” Advances in Computing and Communications in Computer and Information Science,. Springer Berlin Heidelberg, pp. 431–445 2011.

26. C. Meinel, W. Dawoud, and I. Takouna, “Elastic vm for dynamic virtualized resources provisioning and optimization,” *HPI Future SOC Lab*, pp. 13, 2011.
27. N. Roy, A. Dubey, and A. Gokhale, “Efficient autoscaling in the cloud using predictive models for workload forecasting.”Proceedings of the 4th International Conference on Cloud Computing, IEEE, pp. 500–507, 2011.
28. Z. Gong, X. Gu, and J. Wilkes, “Press: Predictive elastic resource scaling for cloud systems,” Proceedings of the 6th International Conference on Network and Service Management, (CNSM), IEEE, pp. 9–16, 2010.
29. N. Vasi´c, D. Novakovi´c, S. Miu˘cin, D. Kosti´c, and R. Bianchini, “Dejavu: accelerating resource allocation in virtualized environments,” Proceedings of the 17th International conference on Architectural Support for Programming Languages and Operating Systems, (ASPLOS), ACM, Vol. 40, No. 1, pp. 423–436, 2012.
30. Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, “Cloudscale: elastic resource scaling for multi-tenant cloud systems,” Proceedings of the 2nd Symposium on Cloud Computing, (SOCC) p. 5ACM, 2011.
31. U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, “A cost-aware elasticity provisioning system for the cloud,” Proceedings of the 31st International Conference on Distributed Computing Systems, IEEE, pp. 559–570, 2011.
32. Lorido-Botrán, Tania, José Miguel-Alonso, and Jose Antonio Lozano. "Auto-scaling techniques for elastic applications in cloud environments." Department of Computer Architecture and Technology, University of Basque Country, Vol. 12, p. 2012, 2012.
33. Barrett, Enda, EndaHowley, and Jim Duggan. "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud." *Concurrency and Computation: Practice and Experience, Vol.* 25, No. 12,pp. 1656-1674, 2013.
34. Urgaonkar, Bhuvan, PrashantShenoy, Abhishek Chandra, PawanGoyal, and Timothy Wood."Agile dynamic provisioning of multi-tier internet applications."ACM Transactions on Autonomous and Adaptive Systems (TAAS), Vol. 3, No. 1, p. 1, 2008
35. Lim, Harold C., ShivnathBabu, and Jeffrey S. Chase. "Automated control for elastic storage."Proceedings of the 7th international conference on Autonomic computing, ACM, pp. 1-10, 2010.
36. Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).
37. Barroso, Luiz André, Jimmy Clidaras, and UrsHölzle. "The datacenter as a computer: An introduction to the design of warehouse-scale machines."*Synthesis lectures on computer architecture,Vol.* 8, No. 3, pp. 1-154, 2013.

38. Qi Zhang, Lu Cheng, and RaoufBoutaba.”Cloud computing: State-of-the-art and research challenges.”,Journal of Internet Services and Applications, Vol. 1, No. 1, pp. 7–18, 2010
39. Koomey, Jonathan G. "Estimating total power consumption by servers in the US and the world.",2007.
40. Digital Power Group. 2013.” The Cloud Begins with Coal—Big Data, Big Networks, Big Infrastructure,and Big Power.” Retrieved July 14, 2015
41. H. Cademartori, “Green Computing Beyond the Data Center.” http:// www.powersavesoftware.com/Download/PS WP Green Computing EN.pdf, 2007.
42. RajkumarBuyya, Anton Beloglazov, and JemalAbawajy. “Energy-efficient management of data center resources for cloud computing. A vision, architectural elements, and open challenges.”Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 6–17, 2010
43. Srikantaiah, Shekhar, AmanKansal, and Feng Zhao. "Energy aware consolidation for cloud computing." Proceedings of the 2008 conference on Power aware computing and systems, Vol. 10, pp. 1-5. 2008.
44. Yong qiangGao, Haibing Guan, Zhengwei Qi, Yang Hou, and Liang Liu. “A multi-objective ant colony system algorithm for virtual machine placement in cloud computing.”Journal of Computer and System Sciences, Vol. 79, No. 8, pp. 1230-1242, 2013.
45. EfthymiaTsamoura, AnastasiosGounaris, and Kostas Tsichlas. “Multi-objective optimization of data flows in a multi-cloud environment”, Proceedings of the 2nd Workshop on Data Analytics in the Cloud, pp.6–10, 2013
46. Jing Xu and Jose, A. B. Fortes, ”Multi-objective virtual machine placement in virtualized data center environments”, Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications and the International Conference on Cyber, Physical, and Social Computing (GREENCOM-CPSCOM’10), pp. 179–188, 2010
47. Mann, ZoltánÁdám. "Allocation of Virtual Machines in Cloud Data Centers—A Survey of Problem Models and Optimization Algorithms." *ACM Computing Surveys (CSUR), Vol.* 48, No. 1, p. 11, 2015.
48. AtefehKhosravi, Saurabh Kumar Garg, and RajkumarBuyya. “Energy and carbon-efficient placement of virtual machines in distributed cloud data centers”,Proceedings of the 19th International Conference on Parallel Processing, pp. 317–328, 2013
49. OferBiran, Antonio Corradi, Mario Fanelli, Luca Foschini, Alexander Nus, Danny Raz, and Ezra Silvera, “A stable network-aware VM placement for cloud systems”, Proceedings of the

12th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. IEEE, Los Alamitos, CA, pp. 498–506, 2012

50. Lei Shi, John Furlong, and Runxin Wang. "Empirical evaluation of vector bin packing algorithms for energy efficient data centers." Proceedings of the IEEE Symposium on Computers and Communications(ISCC), pp. 9–15, 2013.
51. Weijia Song, Zhen Xiao, Qi Chen, and HaipengLuo, "Adaptive resource provisioning for the cloud using online bin packing," IEEE Transactions on Computers, Vol. 63, No. 11, pp. 2647–2660, 2014.
52. Luis Tomas and Johan Tordsson,"An autonomic approach to risk-aware data center overbooking",IEEE Transactions on Cloud Computing, Vol. 2, No. 3, pp. 292–305, 2014.
53. Daniel GuimaraesdoLago, Edmundo R. M. Madeira and Luiz Fernando Bittencourt, "Power-aware virtual machine scheduling on clouds using active cooling control and DVFS", Proceedings of the 9thInternational Workshop on Middleware for Grids, Clouds, and e-Science, pp. 1-6, 2011.
54. PengchengXiong, Yun Chi, Shenghuo Zhu, Hyun Jin Moon, CaltonPu and HakanHacgumus,"Smart-SLA: Cost-sensitive management of virtualized resources for CPU-bound database services," IEEE Transactionson Parallel and Distributed Systems, Vol. 26, No. 5, pp. 1441–1451, 2015.
55. Zhen Xiao, Qi Chen, and HaipengLuo, "Automatic scaling of Internet applications for cloud computing services", in IEEE Transactions on Computers, Vol.63, No. 5, pp. 1111–1123,2014.
56. Anton Beloglazov and RajkumarBuyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," in Concurrency and Computation: Practice and Experience, Vol. 24, No. 13, pp. 1397–1420, 2012
57. Anton Beloglazov and RajkumarBuyya, "Energy efficient allocation of virtual machines in cloud data centers", in Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud, and Grid Computing, pp.577–578, 2010.
58. Anton Beloglazov and RajkumarBuyya, "Energy efficient resource management in virtualized cloud data centers", in Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud, Grid Computing, pp. 826–831, 2010.
59. Anton Beloglazov, JemalAbawajy, and RajkumarBuyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." In Future Generation Computer Systems, Vol.28, pp.755–768, 2012.

60. Anton Beloglazov and RajkumarBuyya, “Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints,”IEEE Transactions on Parallel and Distributed Systems Vol.24, No. 7, pp. 1366–1379, 2013.
61. Norman Bobroff, AndrzejKochut, and Kirk Beaty,” Dynamic placement of virtual machines for managing SLA violations,” Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, pp. 119–128, 2007.
62. AkshatVerma, PuneetAhuja, and AnindyaNeogi,”pMapper:Power and migration cost aware application placement in virtualized systems,” Proceedings of Middleware, Springer Berlin Heidelberg, pp. 243–264,2008
63. Zhen Xiao, Weijia Song, and Qi Chen,“Dynamic resource allocation using virtual machines for cloud computing environment,” IEEE Transactions on Parallel and Distributed Systems, pp. 1107–1117, 2013.
64. R. Jeyarani, N. Nagaveni, R. Vasanth Ram, “Self adaptive particle swarm optimization for efficient virtual machine provisioning in cloud”, in International Journal of Intelligent Information Technology,Vol. 7, No. 2, pp. 25–44, 2011.
65. Jeyarani, Rajarathinam, N. Nagaveni, and R. Vasanth Ram, "Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence," in Future Generation Computer Systems, pp.811-821, 2012.
66. A. BeloglazovandRajkumarBuyya,"Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers." In Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science,Vol. 4, ACM, 2010
67. R.N. Calheiros, R. Ranjan, A.Beloglazov, Cesar A. F. De Rose, R. Buyya, “CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms,” Software: Practice and Experience pp.23–50, 2011
68. Sijin He, Li Guo,MoustafaGhanem and YikeGuo, “Improving resource utilisation in the cloud environment using multivariate probabilistic models,”Proceedings of the IEEE 5th International Conferenceon Cloud Computing, IEEE, pp. 574–581, 2012.
69. Gaojin Wen and Jue Hong, “Energy-aware Hierarchical Scheduling of Applications in Large Scale Data Centers”, in International Conference on Cloud and Service Computing, IEEE, pp.158-165, 2011.
70. Knauth, Thomas and ChristofFetzer, “Energy-aware scheduling for infrastructure clouds”,in 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, pp. 58-65, 2012.

71. Asyabi, Esmail, and Mohsen Sharifi, "A New Approach for Dynamic Virtual Machine Consolidation in Cloud Data Centers", in International Journal of Modern Education and Computer Science (IJMECS), Vol. 7, No. 4, pp. 61-66, 2015.
72. Esha Barlaskar and Y. Jayanta Singh, “Dynamic Consolidation of Virtual Machines with Multi-Agent System”, International Journal of Computer Applications, Vol. 94, No. 9, pp. 30-38,2014.

A Survey on Elastic Resource Allocation Algorithm for Cloud Infrastructure

Kshitiza Vasudeva
Department of Computer Science Engineering,
Jaypee University of Information Technology
Himachal Pradesh, India
kshitizavasudeva@gmail.com

Punit Gupta
Department of Computer Science Engineering,
Jaypee University of Information Technology
Himachal Pradesh, India
Punitg07@gmail.com

ABSTRACT

Elasticity is an asset of cloud computing which makes it better or may be best over the other conventional grid and cluster computing. Cloud elasticity is the ability of the cloud infrastructure to rapidly change the amount of resources allocated to a service during runtime in order to meet the actual varying demands on the service while enforcing SLAs. Initially, elasticity being a key feature of clouds is classified based on Scope, Policy, Purpose and Method of elasticity. This completely explains where, how and why elasticity is crucial. Secondly, diversified related work to elasticity is reviewed and discussed in order to define the state of the art of elasticity in clouds. In this paper we have compared various proposed algorithm based on Quality of service and whether they support elasticity of not.

Keywords

Cloud computing, Power aware computing, Resource Utilization, Hybrid Cloud, Cloud Infrastructure as a service.

INTRODUCTION

Among all the newfangled technologies, cloud computing has completely revolutionized IT industry. To cover all the features, cloud computing has acquired all limitations, checks and advancements of other computing research areas like virtualization, utility computing, service oriented architecture, autonomic computing, distributed and grid computing. It follows 'pay-per-use' model, customer has to pay for only those resources which he/she has used. Developers these days need not to worry about the hardware to deploy their service or the problem of over provisioning / under-provisioning of resources. Cloud computing on whole covers the applications delivered over the internet as a service and the Data Center hardware and software. The goal of this computing model is to make software even more attractive as a service, increase availability of resources and higher throughput.

US Government's National Institute of Standards and Technologies (NIST) [38] defines," cloud computing is 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". This cloud model clearly describes five essential characteristics, four deployment models and three service models. The characteristics include 'on-demand self-service capabilities', 'rapid elasticity', 'broad network access', 'resource pooling' and 'measured service'. The deployments models include private cloud, community cloud, public

cloud and hybrid cloud. The service models include 'SAAS','PAAS' and 'IAAS'.

Elasticity in the context of cloud computing means the capability of the system to expand or shrink the number of resources in an automatic manner so that SLA is not violated with minimum cost incurred. The main elasticity elements or dimensions of any cloud application are cost, resource and quality. Each cloud application or process tries to increase or decrease the cost, maximize resource utilization and improve the quality so as to accommodate the specific requirements. Furthermore, elasticity captures two core aspects speed and precision. Speed can be considered as the time taken by it to swap under-provisioned state and optimal state or vice versa. Precision is difference in the number of currently allocated resources and actual demand.

Now, the issues or challenges which come up while using elasticity as a feature in cloud computing. Resource availability, clouds interoperability, resource granularity, start-up time of VMs and tools and platforms for elastic applications development are some of those and need to be handled. The foremost challenge is to meet these issues without violating Service Level Objectives (SLOs).

This paper propose and dynamic way to minimize the computation and maximize utilization of resource by allocation resource to request by getting energy efficiency and factors which make it more efficient and reliable computation over cloud environment. Power consumption based scheduling lead to efficient computation and increase computation of data center economical.

RELATED WORK

This section discusses the research work on elasticity as a feature of cloud computing to cope up with issues till date. It presents the basic terminologies and classification for elasticity solutions.

Elastic resource provisioning [11] is somewhat like winsome feature provided by Infrastructure As a Service (IAAS) clouds but to decide how many resources to get and when to get make it bit complicated while changing application workload dynamically.

Guilherme Galante et al. [1] proposed a solution for complex and vast elasticity mechanisms by classifying these mechanisms on the basis of features found in studied academic & commercial solutions. They classified elasticity on basis of scope, policy, purpose and method.

The two most common policies of an elastic cloud application are: manual and automatic. The term policy as a characteristic is related to those interactions which are needed while executing elasticity actions. The manual policy and automatic are different on the basis of who is responsible for monitoring his/her virtual environment, applications and then taking an action to perform elasticity. In manual policy user is responsible for all this work and in automatic policy the control and actions are taken by the cloud system or the application itself without the intervention of users. Some public providers which manage resources manually are:GoGrid [12], Rackspace [13] and Microsoft Azure [14], and the frameworks Elastin [15] and Work Queue [16].

Automatic / auto-scaling techniques can be further classified into reactive, proactive and hybrid.

Reactive techniques are also known as rule based methods. The system reacts to changes but doesn't anticipate them. Each rule has some conditions which when satisfied some action is triggered. These conditions are based on the threshold values which vary according to the system. Reactive methods are popular in research and practice ([1],[17],[18],[19]). For instance, reactive methods are used by many public cloud providers (like Amazon, Microsoft),

cloud platforms(like OpenNabula), and third party tools (likeRightScale). Threshold-based rules are explicitly mentioned and popular among current researchwork(e.g., [20] [21] [22] [23]). RightScale's auto-scaling algorithm [24] uses a voting process, that is, all nodes vote for scaling up or down and if majority of the nodes agree then that particular action is performed. This RightScale's auto-scaling algorithm is a complement to reactive rules.

Reactive approach has some shortcomings like the parameters and threshold values which are keys in rules require deep knowledge, an extra effort and expertise. Furthermore, all the existing approaches don't deal with the uncertainty caused by noise and unexpected events in cloud based software which is very common if we think out of theoretical concepts. So, PooyanJamshidi et al. [2] proposed a solution to this problem by developing RobustT2Scale, an elasticity controller which enables quantitative specifications of elasticity rules by utilizing fuzzy logic. These Fuzzy logic systems can manipulate linguistic rules so that conflicting rules can be handled. It is robust to noisy data too.

The virtual resources that cloud computing uses while scaling dynamically don't have negligible setup time. Reactive approach can't solve this problem and also incurs huge cost. So, Proactive techniques are used. These techniques try to predict future resource demand in order to ensure that sufficient resources are available before time.For this prediction some heuristics and analytical methods are used so as to conjecture the systems load behavior and then to decide to scale in/out resources on the basis of the results. In this context, Caron et al. [3] was the one who initiated groundwork for this new approach by developing resource usage prediction algorithm. Some references of the works done by authors using predictive techniques to scale resources are Gong et al. [28], Vasi´c et al. [29], Shen et al. [30], Sharma et al. [31] Roy et al. [27], Dawoud et al. [25], [26].

Time series analysis, machine learning, queuing models and control theory are some popular techniques used in predictive approach. Time series analysis[30]use historical data usage to predict the future resource demand and work on particular domain. It performs well and better if provided with large historical data and interval size is optimum[32]. Reinforcement learning [33] enables the policies to learn from observations. But it is suitable for only stable workloads because prolonged learning is required. Queuing theory [34] sets many restrictive assumptions. And due to this restriction only stationary scenarios fulfill these assumptions so whenever conditions change it needs to calculate the values again[32]. Last, the controllers [22],[35] take some input and give an output which should be maintained at some desired level. Outputs change as the values of the input parameters change. In some works [36] prediction algorithms are neglected due to dynamics.

Hybrid auto-scaling, the third category of automatic policy combines the other two reactive and proactive. Reactive is considered when working on short time scale and proactive when time scale is long[34]. In this category Ahmed Ali-Eldin et al. [4] introduced two adaptive hybrid controllers Pc1 and Pc2 that use hybrid approach to know the current and predict future demand. Then on the basis of this prediction it dynamically scales the VM resources in a cloud. Results proved that after using reactive technique for scaling up and predictive for scaling down SLA violations rate improved 2-10 times when compared to only reactive approach.

As mentioned before, the main elements of any cloud application are cost, resource and quality. So, an ideal situation is to incur less cost with better resource utilization and good quality

without violating SLA. Many authors proposed solutions for this problem. Some used reactive approach, proactive or hybrid.

Ms.M.UthayaBanu et al. [5] proposed a solution to minimize the service provision cost in both reservation and on-demand plan using hybrid approach. They divided the resource subscription problem into two sub-problems: how many long term resources to be reserved and how many on-demand resources to be acquired. They proposed a two-phase algorithm. In the first phase, a mathematical formula is used to reserve correct and optimal amount of resources during reservation and in second phase, Kalman Filter is used to predict resource demand. The results showed that it significantly reduced provision cost and prediction is of reasonable accuracy.

Many cloud services using VM level scaling may overuse resources increasing the operating cost of the cloud provider. Rui Han et al. [6] gave a solution for this extra cost as well as overuse of resources in cloud services. They proposed a lightweight scaling (LS) algorithm to enable fine grained scaling of an application at the level of underlying resources namely CPU, memory and input/output.The algorithm tries to use the idle resources at the max to release overload resources before scaling in other nodes which increases resource utilization of PMs. This approach efficiently meets the QoS requirements and also reduces the cost by scaling resources in/out.

Cloud applications handle dynamic scalability through virtualization. The drawback of virtualization is that the setup time of virtual machines is non-zero. This drawback can't be neglected when considering efficiency and performance.

Eddy Caron et al.[3] proposed a new resource usage prediction algorithm for solving this problem of non-zero setup time. The algorithm uses historic data saved in the past to match similar usage patterns of the current window of records. Then after matching, algorithm predicts the system usage by interpolating what follows after those matched patterns from the historic data. Algorithm proves better when provided with input data of same application domain and improving on the data size plus interval size. Kupfermanet al. [7] suggested some set of scoring metrics to measure and compare the efficiency of existing dynamic scaling algorithms (one developed by RightScale, two with prediction approach linear regression and auto-regression of order 1) . The metrics is based on the terms of availability and operation cost. After analyzing, dynamic provisioning including reactive and proactive policies proved to be better in terms of cost with negligible availability drop. Moreover, predicting of future resource demands enhances the performance and efficiency of the system at times when traffic is random with sharp spikes or low.

Georgiana Copilet al. [8] proposed and presented SYBL, an extensible language and its runtime system to control elasticity requirements with respect to Service Level Objectives (SLOs) in cloud applications. While controlling elasticity of an application, rules make use of some parameters, constraints and directives for monitoring those values of constraints. SYBL features, covers all required elasticity constraints, monitoring directives and strategies for controlling application's elasticity in all flexible ways. They also propose and present a prototype implementation along with some experiments illustrating how SYBL can be used in real world scenarios.

Table 1: Literature review

S.No	Paper	Cost	Resource	Quality	SLA	Elasticity policy
1.	Nguyen et al.[2013]	Yes	no	no	yes	Automatic(hybrid)
2	Eldin et al.[2012]	No	no	no	yes	Automatic(hybrid)
3.	Jamshidiet al.[2014]	Yes	yes	no	yes	Automatic(Reactive)
4.	Caron et al.[2010]	Yes	no	no	no	Automatic(proactive)
5.	Rui Han et al.[2012]	Yes	yes	no	no	Automatic(reactive)
6.	M.UthayaBanu, K.Saravanan [2013]	Yes	yes	no	no	Automatic(Hybrid)
7.	Samuel et al.[2013]	No	no	no	yes	Automatic(proactive)
8.	Jara et al.[2009]	Yes	no	no	yes	Automatic(hybrid)
9.	Copil et al.[2013]	Yes	yes	yes	yes	automatic
10.	Islam et al. [2012]	Yes	no	no	yes	Automatic(proactive)

Hiep Nguyen et al.[9] proposed a system AGILE , a practical elastic distributed resource scaling system for IAAS cloud infrastructures. Along with dynamic work load changes AGILE also considers the interference from other users at runtime. AGILE provides medium-term resource predictions so that there is enough time for scaling up the resources of the server and application's SLO is not affected by workload increase. AGILE implements live cloning to
Scale up the performance by replicating running VMs ahead of time. In contrast to previous resource demand prediction schemes [37, 28], AGILE achieves enough lead time for setting up the VMs with good prediction accuracy. By combining this medium term resource demand predictions and online profiling AGILE can very well predict whether an application will face an extra workload. And if it happens then how many new servers should be added to avoid that situation.

AJILA A. Samuel et al. [10] proposed a solution for efficient scaling of VM resources and proactive provisioning to meet SLA (Service Level Agreement) in cloud computing environment. A cloud client prediction model for transactional web e-commerce benchmark (TPC-W) application is developed and evaluated using three techniques of machine learning: Support Vector Machine (SVM), Neural Network (NN) and Linear Regression(LR). Results and analysis from the experiments carried out on Amazon elastic compute cloud(EC2) showed that SVM provides best prediction model for random like workload traffic pattern.

5. Conclusion

In this paper, a short survey is done on the current research on elasticity dimension of cloud. All the approaches proposed by authors are differentiated on the basis of policies used like manual or automatic. (reactive or proactive). Elasticity is measured using three parameters cost, resource and elasticity. Some authors have worked on one of the parameters, two or all of the three. In [3] Eddy Caron et al. used modified KMP to reduce the running time of the KMP algorithm to Θ(n*m) . Still it is very time consuming to search for some pattern over the entire set of historical data. In [2] PooyanJamshidi et al. proposed a solution for noise and conflicting rules by developing an elasticity controller which considers horizontal scaling. This can be improved by scaling vertically too.

References

[1] G. Galante and L. C. E. de Bona," A survey on cloud computing elasticity," in IEEE/ACM Fifth International Conference on Utility and Cloud Computing (UCC) , pp. 263-270, 2012.

[2] Jamshidi, Pooyan, Aakash Ahmad, and Claus Pahl, "Autonomic resource provisioning for cloud-based software", Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, ACM, pp. 95-104, 2014.

[3] E. Caron, F. Desprez, and A. Muresan, "Forecasting for Grid and Cloud Computing On Demand Resources Based on Pattern Matching," Cloud Computing Technology and Science (CloudCom), pp. 456-463, 2010.

[4] Ali-Eldin, Ahmed, Johan Tordsson, and Erik Elmroth. "An adaptive hybrid elasticity controller for cloud infrastructures," Network Operations and Management Symposium (NOMS), IEEE, pp. 204-212, 2012

[5] M.UthayaBanu, K.Saravanan, "Optimizing the Cost for Resource Subscription Policy in IaaSCloud",International Journal of Engineering Trends and Technology (IJETT),Vol. 6 No. 5, pp. 296-301, 2013

[6] R.Han, Li Guo, Moustafa M. Ghanem and YikeGuo, "Lightweight resource scaling for cloud applications."IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp. 644-651,2012.

[7] Kupferman, Jonathan, Jeff Silverman,PatricioJara, and Jeff Browne, "Scaling into the cloud," CS270-advanced operating systems, pp. 1-8, 2009.

[8] G. Copil, D. Moldovan, H.L.Truong and S. Dustdar, "SYBL: An Extensible Language for Controlling

Elasticity in Cloud Applications", in IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, IEEE, pp. 112-119, May 2013

[9] H.Nguyen, Z.Shen, X.Gu, S.Subbiah, J.Wilkes, "AGILE: Elastic Distributed Resource Scaling for Infrastructure-as-a-Service", in Proceedings of the 10th International Conference on Autonomic Computing (ICAC), pp. 69-82, 2013

[10] Ajila A. Samuel and Bankole A. Akindele ,"Proactive Prediction Models for Web Application Resource Provisioning in the Cloud", in Transition from observation to knowledge, pp. 17-35, 2014

[11] Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2/.

[12] "GoGrid." [Online]. Available: http://www.gogrid.com/

[13] "Rackspace." [Online]. Available: http://www.rackspace.com/

[14] "Microsoft Azure." [Online]. Available: http://www.windowsazure.com/

[15] I. Neamtiu, "Elastic executions from inelastic programs," in Proceedings of the 6th Intl. Symposium on Software Engineering for Adaptive and Self-Managing Systems, ser. SEAMS ACM, pp. 178–183,2011.

[16] D. Rajan, A. Canino, J. A. Izaguirre, and D. Thain, "Converting a high performance application to an elastic cloud application," Proceedings of the 3rd International Conference on Cloud Computing Technology and Science, CLOUDCOM IEEE, pp. 383–390, 2011.

[17] S. Meng, L. Liu, and V. Soundararajan, "Tide: achieving self-scaling in virtualized datacenter management middleware," in Proceedings of the 11th International Middleware Conference, ACM, pp. 17–22, 2010.

[18] R. N. Calheiros, C. Vecchiola, D. Karunamoorthy, and R. Buyya, "The aneka platform and qos-driven resource provisioning for elastic applications on hybrid clouds," in Future Generation Computer Systems, vol. 28, no. 6, pp. 861–870, June 2011.

[19] J. O. Fit´o, I. G. Presa, and J. G. Fern´andez, "Sla-driven elastic cloud hosting provider," in Proceedings of the 18th EuromicroConferenceon Parallel, Distributed and Network-based Processing, ser. IEEE, pp. 111–118, 2010.

[20] Dutreilh, Xavier, Nicolas Rivierre, Aurlien Moreau, Jacques Malenfant, and Isis Truck, "From data center resource allocation to control theory and back" IEEE 3rd International Conference on Cloud Computing (CLOUD), IEEE, pp. 410-417, 2010.

[21] Maurer, Michael, IvonaBrandic, and RizosSakellariou, "Enacting SLAs in clouds using rules", in Euro-Par 2011 Parallel Processing, Springer Berlin Heidelberg, pp. 455-466, 2011.

[22] Lim, Harold C., ShivnathBabu, Jeffrey S. Chase, and Sujay S. Parekh,"Automated control in cloud computing: challenges and opportunities", in Proceedings of the 1st workshop on Automated control for datacenters and clouds, ACM, pp. 13-18, 2009.

[23] Marshall, Paul, Kate Keahey, and Tim Freeman, "Elastic site: Using clouds to elastically extend site resources," in Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, IEEE Computer Society, pp. 43-52, 2010.

[24] Caron, Eddy, Luis Rodero-Merino, FrédéricDesprez, and Adrian Muresan. "Auto-scaling, load balancing and monitoring in commercial and open-source clouds." RR- 7857, INRIA, pp.27, 2012.

[25] W. Dawoud, I. Takouna, and C. Meinel, "Elastic vm for cloud resources provisioning optimization," Advances in Computing and Communications in Computer and Information Science,. Springer Berlin Heidelberg, pp. 431–445 2011.

[26] C. Meinel, W. Dawoud, and I. Takouna, "Elastic vm for dynamic virtualized resources provisioning and optimization," *HPI Future SOC Lab*, pp. 13, 2011.

[27] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting."Proceedings of the 4th International Conference on Cloud Computing, IEEE, pp. 500–507, 2011.

[28] Z. Gong, X. Gu, and J. Wilkes, "Press: Predictive elastic resource scaling for cloud systems," Proceedings of the 6th International Conference on Network and Service Management, (CNSM), IEEE, pp. 9–16, 2010.

[29]. N. Vasi´c, D. Novakovi´c, S. Miu˘cin, D. Kosti´c, and R. Bianchini, "Dejavu: accelerating resource allocation in virtualized environments," Proceedings of the 17th International conference on Architectural Support for Programming Languages and Operating Systems, (ASPLOS), ACM, Vol. 40, No. 1, pp. 423–436, 2012.

[29] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, “Cloudscale: elastic resource scaling for multi-tenant cloud systems,” Proceedings of the 2nd Symposium on Cloud Computing, (SOCC) p. 5ACM, 2011.

[30] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, “A cost-aware elasticity provisioning system for the cloud,” Proceedings of the 31st International Conference on Distributed Computing Systems, IEEE, pp. 559–570, 2011.

[31] Lorido-Botrán, Tania, José Miguel-Alonso, and Jose Antonio Lozano. "Auto-scaling techniques for elastic applications in cloud environments." Department of Computer Architecture and Technology, University of Basque Country, Vol. 12, p. 2012, 2012.

[32] Barrett, Enda, EndaHowley, and Jim Duggan. "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud." *Concurrency and Computation: Practice and Experience, Vol.* 25, No. 12,pp. 1656-1674, 2013.

[33] Urgaonkar, Bhuvan, PrashantShenoy, Abhishek Chandra, PawanGoyal, and Timothy Wood."Agile dynamic provisioning of multi-tier internet applications."ACM Transactions on Autonomous and Adaptive Systems (TAAS), Vol. 3, No. 1, p. 1, 2008

[34] Lim, Harold C., ShivnathBabu, and Jeffrey S. Chase. "Automated control for elastic storage."Proceedings of the 7th international conference on Autonomic computing, ACM, pp. 1-10, 2010.

[35] Gandhi, Anshul, M. Harchol and Raghunathan, "Autoscale: Dynamic, robust capacity management for multi-tier data centers," *TOCS,* 2012

[36] Gmach, Daniel, Rolia, L. Cherkasova, and Kemper.” Capacity management and demand prediction for next generation data centers”. In International Conference on Web Services, 2007.

Adaptive Heuristics with Self-Healing for Efficient Dynamic Consolidation of Virtual Machines in Cloud Datacenters

Kshitiza Vasudeva
Student, Computer Science & Engineering Department
Jaypee University of Information Technology, Solan, Himachal Pradesh, India
kshitizavasudeva@gmail.com

Dr. Satya Prakash Ghrera
Head, Computer Science Engineering& Information Technology Department
Jaypee University of Information Technology, Solan, Himachal Pradesh, India
sp.ghrera@juit.ac.in

Abstract-***Energy consumption in cloud data centers is major issue due to cost expenses, performance degradation and environmental impact. To reduce the power consumption many ideas have been proposed among which live VM migration is the most popular one. Live migration further increases the overhead, delay and degrading the performance. The concept of self healing is proposed to reduce the power consumption as well as number of virtual machine (VM) migrations. The main aim of this work is to self-heal the overloaded host before going for VM migration. The proposed algorithm is developed and implemented using CloudSim toolkit. The results demonstrate that the proposed system can handle dynamic workloads and show better performance***.

Keywords-***CloudComputing,Virtualization,DynamicConsolidation,Self-Healing***

INTRODUCTION

Cloud computing is among the most fast growing and symbolic contemporary technologies that has brought up sort of revolution in modern ICT [1]. Cloud Computing " is a model for allowing ubiquitous, convenient, and on-demand network access to a number of configured computing resources (e.g., networks, server, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [2]. It has proved to be a powerful architecture which can handle large scale systems and perform complex computing. Cloud computing has wiped off the need to store, maintain and evaluate huge datasets through virtualization .It has not only minimized the cost of infrastructure storage and maintenance but also made systems more secure, reliable and efficiently manageable [3]. It efficiently addresses issues of rapid growth of economies and limited number of resources. Cloud computing gives full opportunity to organizations to just work on main logics related to business rather than being concerned about the infrastructure, availability of resources, man power, cost, security, etc. [4]

The fundamental basis of cloud is that content of users is not stored on local systems but is kept and processed in the datacenters through internet. The main technology used by cloud computing is virtualization. It is used for abstraction of the computing resources. The cloud providers are responsible for the management and maintenance of these data centers. The cloud providers provide Application Programming Interface (API) to the users so that they can access the data stored through any computing device connected to the internet.

Energy Consumption Issue In Cloud Data Centers

Cloud computing service is a collection of virtual data centers with high optimization which not only software but also hardware and other resources.Companies and other organizations in need of any resource can use resources through pay-per-use model by simply connecting to the cloud.This curtails down their capital expenditure on extra resources at premises. To cope up with growing demand of computational power for high performance applications, companies need large scale data centers which are the core part of system.

However, these data centers devour colossal amount of electrical power which has exceeded the cost of actual infrastructure. To make maximum profit, saving operational cost is preferred over performance. People have begun to pay more attention to energy consumption rather than only considering performance [5].

According to the survey presented [6], data centers in US consumed around 61 billion KWh in 2006 which is almost equivalent to 1.5% of the total energy consumption in US. Moreover, in 2011 it jumped to more than 100 billion KWh [7]. In spite of these economical reasons, energy consumption has adverse effects on the environment too. Servers emit CO2 which is the main cause of greenhouse effect. And according to the survey, this high energy consumption and CO2 emissions will keep growing in upcoming years.

Many different applications are run at the same data center which contains many heterogeneous servers and network devices. To keep these applications isolated and exploit features of cloud like elasticity, flexibility and reliability, cloud uses virtualization technology. Virtual machines(VMs) are the basic blocks of resources which are provided to customers either directly or indirectly through the provisioned applications.[8]

To save energy in data center best way is to efficiently utilize the resources i.e the VMs. VMs are consolidated to minimum number of hosts and idle hosts are switched off or put in other mode of operation like sleep mode. However, sometimes VM consolidation becomes too combative which may overload hosts and violate SLOs fixed in Service Level Agreement (SLA). Hence, there is a trade off between the QOS and energy consumption which is optimized by allocating VMs efficiently [9, 10].

Elasticity as a feature of cloud is the ability of the system to scale up and down the resources according to the current demand. So, when the load is high or low, the system should consolidate VMs accordingly with respect to QoS and saving power. Methods used for implementing elasticity are re-dimensioning, migration and replication.

VM allocation problem is divided in four steps [11]:

a) Overloaded host detection

b) VM selection for live migration

c) Detecting underutilized host

d) Migrating all VMs and turning off the host.

This live VM migration causes delay and overburdens the network as well as the physical hosts involved. Hence, violates SLA and degrades the performance of the system.

In this work we have proposed the concept of self-healing to reduce the number of VM migrations and power consumption without violating the SLA.

The remaining paper is setup as follows. Next section 3 discusses the literature review by other research scholars. Section 4 has the proposed system model with all the details of architecture and algorithm. Section 5 shows the simulated results and their analysis by comparing with the existing system. Section 6 concludes the paper with respect to proposal and results.

LITERATURE REVIEW

Energy consumed by distributed systems has brought up many issues and has become an outstanding question and requires consideration. Among all the existing methods to save energy, energy consumption can be reduced by proper scheduling of the applications and consolidating them to reduce the running servers.However, many scheduling approaches yet did not acknowledge the cost of energy consumption on network devices, which is also contributesto power consumption in data centers. Hierarchical Scheduling Algorithm (HSA) was proposed to curtail the energy consumed by both servers and network devices. In HSA, a Dynamic Maximum Node Sorting (DMNS) method dealt with optimizing the placement of applications on servers. To further lessen the number of working servers, Hierarchical crossing-switch adjustment is used. Results showed that the number of working servers as well as data transfer speed reduced to good extent.The HSA is simple and robust to minimize the energy consumption by effectively scheduling the applications but HSA and DMNS are not suitable for dynamic workload [12].

An immediate fix to curtail the power consumption in data centers is to utilize the modes with lower power. To measure the variation in energy consumption due to virtual machine scheduler's simulation was conducted and besides also demonstratedthe inability of default schedulers, using optimized scheduler. The customized scheduler has reduced the complete machine uptime by up to 60.1% after using many real simulation scenarios. OptSched optimizes the virtual machine to physical host mapping by utilizing the reservation length. The parameters covered in this study were heterogeneity of data centers and VMs, the long effect of run time distributions and sensitivity to batch requests. The cumulative machine uptime is balanced for heterogeneity of virtual machines but energy consumption is not efficient if the work load is highly dynamic [13].

Beloglazov et al. gave two step proposal to efficiently allocate the VMs. In the first step, new requests for VM provisioning are allowed and VMs placed on hosts, and in next step current allocation setup of VMs is optimized. The first part is somewhat like a bin packing problem with variable bin sizes and prices. As a solution, modification of the Best Fit Decreasing (BFD) algorithm is applied. In MBFD, VMs are sorted in decreasing order of CPU utilization and then VM is allocated to a host which shows minimum increase in power consumption after allocation.This gives a chance to choose the most efficient one with respect to power. The complexity of the algorithm is n·m, where n is the number of VMs. Optimization part of VM consolidation is further carried out in two steps: in first part VMs are selected to migrate and then chosen VMs are allocated to the host based on MBFD algorithm.The energy consumption is less with respect to the reliable QoS. The proposed approach does not follow the strict SLAs between the service provider and user under the dynamic workload [14].

The core technology used by cloud is virtualization so as to adequately consolidate the VMs into physical host for better utilization of resources and to save power. A survey done by many show that the average utilization of servers is still less than expected.A new concept was proposed for this dynamic consolidation of VMs by a dynamic programming algorithm which chooses the VMs from an overutilized host taking into consideration the overhead caused by migrating a VM. Since, all VMs are attached to a storage area network (SAN), the cost of live migration of a VM is decided by its memory imprint. Therefore, time taken by a VM to migrate is calculated by dividing the memory size of VM by network bandwidth. As a result, cost of migration is measured by memory size of the VM. Thus, while selecting the one with less memory size is the best. The cost based approach of VM migration minimizes the power consumption cost of the service provider but when the workload is variable with respect the application, the approach is failed to meet the SLAs [15].

The large-scale data centers contain thousands of servers whichconsume large amount of electrical power leading to high operating costs. Therefore, to curtailthis cost of power the cloud providers need to optimize resource usage effectively by consolidating VMs efficiently in order to improve energy efficiency. The problem of VM consolidation is divided into four sub-problems: physical host overload detection; host under-load detection; VM selection and VM placement. Each of the sub partswork together to optimize the trade off between energy and QoS. For dynamic consolidation of VMs, a new multi-agent system (MAS) was proposed to make the cloud system smarter by blending the five traits of multi agent systems which are ubiquity, intelligence, delegation, interconnection, and human orientation. MASs provide the cloud systems intelligent and insightful based software which can help in effective and better system. The proposed method has significantly reduced energy consumption and also kept constant with the objectives of the Service Level Agreements (SLA).The number of VM migration and energy consumption is effectively minimized but when the workload is variable with respect the application, the approach is failed to meet the SLAs [16].

Extra load on server machine causes performance degradation of applications because resources available are not sufficient. Currently all the proposed methods towards the issue of host overload detection are mostly heuristic-based, or depend onhistorical data analysed statistically. The drawback of this way is that it gives sub-optimal results.Beloglazov et al have given a solution to host overload detection problem by maximizing the mean inter-migration time keeping in mind the specified SLA based on a Markov chain model. Multisize Sliding Window workload estimation technique is also proposed to heuristically adapt the algorithm to unknown non-stationary workloads. It is quite probable that when the resources are utilized at max, the applications are more prone to lack of resources and performance degradation. To address this problem, most of the schemes for dynamic VM consolidation apply either heuristic-based technique, such as static utilization thresholds. The mean inter-migration time is reduced of the VM migration but the number of migration of VM is high which violates the SLAs [17].

In this paper, the main framework considered is an Infrastructure as a Service (IaaS) environment. It represents N number of different physical hosts. Each host is portrayed by the CPU performance defined in MIPS, RAM and transfer speed. Virtual machines (VMs) are hosted on the physical machines on-demand of the users.Customers submit their requirements specifying MIPS, bandwidth, number of processors, etc. Different users hosting various types of applications use resources simultaneously. For this transparency, cloud system uses virtualization technology.

The Proposed Model

The system model shows software layer of the framework which consists of global resource manager, local managers and VMs hosted on physical machines. Each node has one local manager as a part of VMM which keeps a continuous check on resource requirements of VMs and also makes decision regarding selection and migration of VMs at time of overload. The global manager is a part of master node which is in contact with all the local managers and collects information so as to optimize the resource utilization and Service level objectives. It gives the orders regarding VM placement. Actual migration and resizing of VMs is performed by VMM.

PROPOSED WORK

An adaptive heuristics is used for dynamic consolidation of VMs based on an analysis of previous data from the resource usage by VMs. The proposed algorithm significantly reduces energy consumption, while ensuring a high level of adherence to the Service Level Agreement(SLAs). The proposed algorithm performs dynamic consolidation of VMs at run-time on the basis of current utilization of resources which may involve live VM migration, changing the mode of unused host to lower power mode so that power can be saved.The system efficiently handles firm SLAand multi-core CPU architectures. The algorithm adapts the behaviour with respect to observations and characteristics of VMs.

SELF-HEALING

With respect to the proposed work mentioned above, the live migration of virtual machine from the overloaded host is not performed at the first instance. Instead of that, for each of the over utilized host self healing is performed. All virtual machines utilization is analyzed in each overloaded host, then add the MIPS to the more utilized VM and remove MIPS from less utilized VM or if the host has some free PE or MIPS that can be added to more utilized VM. So, overloaded host adjusts VM parameters using self healing and balance the utilization without violating the SLAs. If self healing is not possible, then proposed approach performs the normal VM live migration algorithm where the migrating VMs are selected from the overloaded host and these VMs are migrated to the other host using some policy. For all underutilized hosts, all the VMs are migrated to the safe host.

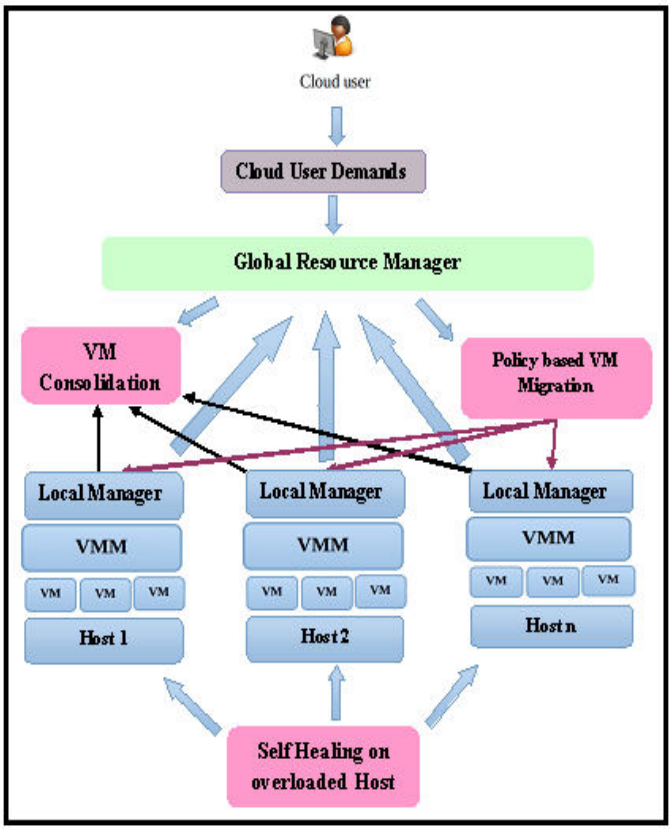

Fig 1. The System Model.

The proposed algorithm with self-healing so as to curtail down the number of migrations at first instance and energy consumption is shown below:

STEP 1: Create the user demands such as application configuration and VM configuration
STEP 2: Initialize the CloudSim and create the datacenter, broker, hosts, and VM based on the user demands
STEP 3: Create cloudlets (jobs or applications) for user requirements
STEP 4: Schedule the task on the VM based on VM allocation policy
STEP 5: Start simulation
STEP 6: Calculate the CPU utilization on the every host
STEP 7: Iteration
7a: get the first host in the list
7b: if host CPU utilization is lower than 0.2 then move the host to underutilized host list
7c: if host CPU utilization is greater than 0.8 then move the host to over utilized host list
7d: else mover the host to safe host list
Close the for loop
STEP 8: Iteration over utilized host list get the VM with maximum utilization
8a: get the available MIPS from the host of maximum utilized VM
8b: if MIPS is available then add available MIPS to over utilized VM
8c: else migrate the VM to safe host based on some policy
STEP 9: Iteration for each underutilized host
9a: Consolidate the every VM on overloaded host and move those
9b: VMs to migration list
Close the for loop
STEP 10: Sort the safe host in increasing order based on CPU utilization and migrate all the VMs based policy (migrate the VM with maximum utilization to host with minimum utilization to achieve the balancing)
STEP 11: Run applications
STEP 12: Stop simulation.

SIMULATED RESULTS AND ANALYSIS

The CloudSim toolkit [20] has been selected as a platform for simulation, as it is a modern simulation framework setup for Cloud Computing environments. CloudSim 3.0.3 version is used. Data centers as physical nodes have been simulated half of which are HP ProLiant ML110 G4 servers, and the other half HP ProLiant ML110 G5 servers. For evaluation of the proposed system with respect to existing system in cloudSim we have chosen two metrics.The two are energy consumption by the server machine in data center and Number of VM migrations due to application workloads. Table 1 shows the assumptions by which power consumption is calculated in cloudSim.

Table 1. Power consumption by the selected servers at different load levels in Watts

CPU UTILIZATION IN PERCENTAGE (%)

SERVER	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiantG4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

For each particular policy given below, Power Consumption and number of VM migrations have been calculated for the existing as well as proposed algorithm by using combinations of host overloaded detection algorithms and VM Selection Algorithms.
Combinations used are:

- **IQR-MMT**(Inter quartile Range as host overloading detection algorithm and Minimum Migration Time Policy as VM selection policy)
- **LR-MMT**(Local Regression and Minimum migration Time policy)
- **MAD-MMT**(Mean Absolute Deviation and Minimum migration Time policy)
- **THR-MMT**(static Threshold and Minimum migration Time policy)
- **LR-RC**(Local Regression and Random Choice Policy)

According to proposed scheme simulated results have been shown and compared with the existing values of the metrics. Table 2 shows the results of Energy Consumption in kWh of the proposed algorithm and existing system.

Table 2. Energy consumption in kWh by the policies

Policy	Existing system	Proposed system
IQR-MMT	47.85	38.93
LR-MMT	35.37	36.61
MAD-MMT	45.61	32.65
THR-MMT	41.81	32.24
LR-RS	34.41	28.77

According to the proposed algorithm, above results can be visualized in graph to evaluate the performance of the algorithm in terms of energy consumption. LR-RS policy consumes minimum energy.

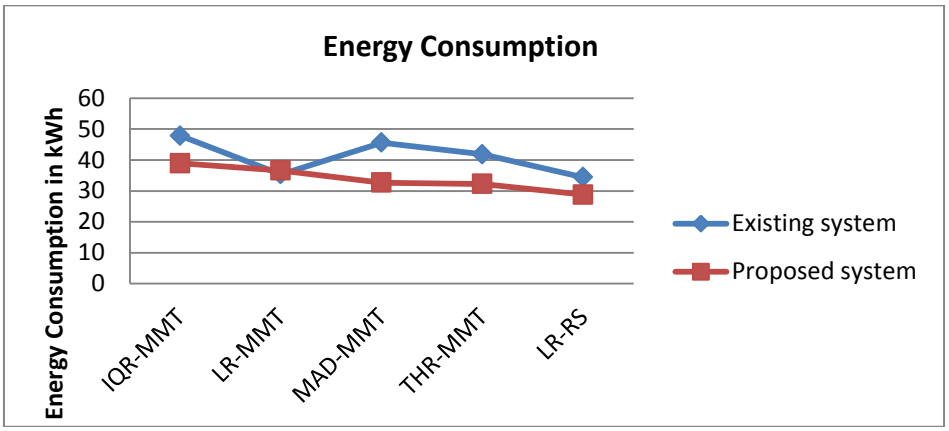

Fig 2. Comparison of policies based on Energy Consumption.

Above Figure 2 shows comparison of policies based on energy consumption using proposed and existing system.

Table 3. Number of VM migrations

Policy	Existing system	Proposed system
IQR-MMT	5502	3395
LR-MMT	2872	1867
MAD-MMT	5265	3103
THR-MMT	4839	3146
LR-RS	2434	1194

Table 3 shows the results of number of VM migrations of the proposed algorithm and existing system.

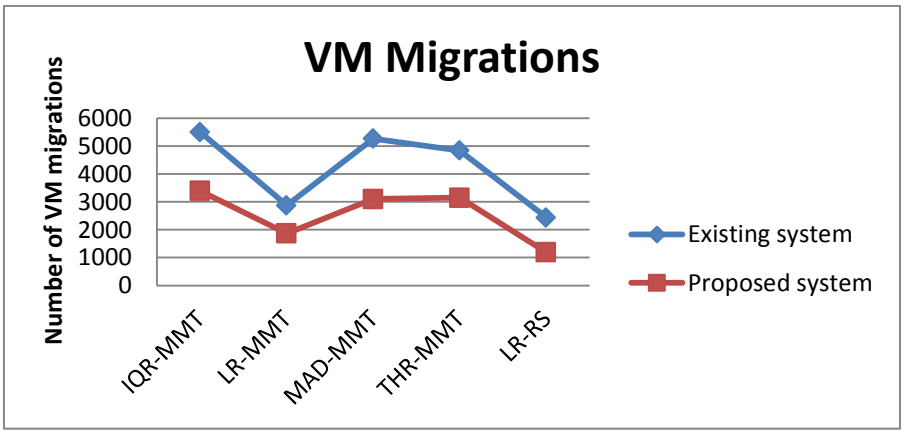

Fig 3. Comparison of policies based on VM Migrations.

Above Figure 3 shows comparison of policies based on number of VM migrations using proposed and existing system.

CONCLUSION

In order to cope up with growing dynamic load and to keep balance with operating cost, cloud providers need to optimize the resource utilization. High Energy consumption at cloud data centers has received much attention in past few years due to SLA violations, high operating cost, CO_2 which has bad impact on environment. This problem of energy consumption with live VM migration is discussed above. Many research scholars have studied this issue and tried to solve by proposing different strategies which are mentioned in literature survey. With respect to the work done before, a concept of self-healing is proposed to reduce the VM migrations and power consumption too. The proposed algorithm is mentioned as well as the simulated results and their analysis. Hence the number of VM migration are reduced, the waiting time of the running application on the VM is avoided. Hence proposed strategy strictly follows the SLAs assured between the cloud users and cloud service providers. If the self healing is not possible the, VM migration is performed on the basis policies existing in cloudSim toolkit to reduce the host overload. The VMs consolidation is also effectively done by proposed strategy and VMs placement is estimated based on the current resources requirements of the every VMs running on the host.

REFERENCES

[1] Hashem, Ibrahim AbakerTargio, IbrarYaqoob, Abdullah Gani, “The rise of big data on cloud computing: Review and open research issues,” Information Systems, Elsevier, Vol. 47, pp. 98–115, 2015.

[2] Lu, Chih-Wei, Chih-Ming Hsieh, Chih-Hung Chang, and Chao-Tung Yang. "An improvement to data service in cloud computing with content sensitive transaction analysis and adaptation." 37th IEEE Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 463-468. IEEE, 2013.

[3] P.Mell, T.Grance, “The NIST definition of cloud computing(draft)”, NIST Special Publication, pp. 800-145.

[4] A.Giuseppe,B.Alessio,D.Walter,P.Antonio,“Survey cloud monitoring: a survey, Computer Network”, International Journal of Computer and Telecommunications Networking, Elsevier, Vol. 57 No. 9, pp. 2093-2115, 2013.

[5] Uddin M, Rahman A A, “Energy efficiency and low carbon enabler green IT framework for datacenters considering green metrics”,Renewable and Sustainable Energy Reviews, Elsevier, Vol. 16, No. 6, pp. 4078–4094, 2012.

[6] Brown R, “Report to congress on server and data center energy efficiency Public law109 431”, Lawrence Berkeley National Laboratory 2008.

[7] Energy Star program requirements for computer systems–draft 4. Washington, DC: Environmental Protection Agency 2009.

[8] Qi Zhang, Lu Cheng, and RaoufBoutaba, “Cloud computing: State-of-the-art and research challenges”, Journal of Internet Services and Applications, Springer, Vol. 1, No. 1, pp 7-18, 2010.

[9] Srikantaiah, Shekhar, Aman Kansal, and Feng Zhao. "Energy aware consolidation for cloud computing", Proceedings of conference on Power aware computing and systems, Vol. 10, pp. 1-5. 2008.

[10] RajkumarBuyya, Anton Beloglazov, and JemalAbawajy. 2010. “Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges”. In Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, 6–17.

[11] AntonBeloglazov, and RajkumarBuyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers."Concurrencyand Computation: Practice and Experience, Vol. 24, No.13, pp. 1397-1420, 2012.

[12] Gaojin Wen and Jue Hong, “Energy-aware Hierarchical Scheduling of Applications in Large Scale Data Centers”, 2011 International Conference on Cloud and Service Computing, IEEE, pp.158-165, 2011.

[13] Knauth, Thomas, and ChristofFetzer, “Energy-aware scheduling for infrastructure clouds.” 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, pages 58-65, 2012.

[14] Anton Beloglazov and Rajkumar Buyya, "Energy efficient allocation of virtual machines in cloud data centers." 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp. 577-578, 2010.
[15] Asyabi, Esmail, and Mohsen Sharifi, "A New Approach for Dynamic Virtual Machine Consolidation in Cloud Data Centers", International Journal of Modern Education and Computer Science (IJMECS), Vol. 7, No. 4, pp. 61-66, 2015.
[16] EshaBarlaskar and Y. Jayanta Singh, "Dynamic Consolidation of Virtual Machines with Multi-Agent System", International Journal of Computer Applications, Vol. 94, No. 9, pp. 30-38,2014.
[17] Anton Beloglazov and RajkumarBuyya, "Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints", IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 7, pp. 1366-1379, 2012.
[18] Barroso LA, Holzle U, "The case for energy-proportional computing" Computer, Vol. 12, pp. 33–37, 2007.
[19] Fan, Xiaobo, Wolf-Dietrich Weber, and Luiz Andre Barroso, "Power provisioning for a warehouse-sized computer." In ACM SIGARCH Computer Architecture News, ACM, Vol. 35, No. 2, pp. 13-23, 2007.
[20] Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R. "CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms" Software:Practice and Experience pp.23–50, 2011.