Energy Aware Model for Cloud Computing

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Information Security

Submitted By
Vipul Moudgil
Roll No. 801633009

Under the supervision of:

Dr. Neeraj Kumar
Associate Professor, CSED, Thapar Institute of Engineering and Technology(TIET)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

June 2018

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "Energy Aware Model for Cloud Computing", in partial fulfillment of the requirements for the award of degree of Master of Engineering in Information Security submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Neeraj Kumar and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Vipul Moudgil)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Neeraj Kumar)
Associate Professor, CSED

ACKNOWLEDGEMENTS

I am truly thankful to my advisor Dr. Neeraj Kumar whose encouragement and guidance at every step enabled me to develop an understanding of the subject. I am grateful for his patience, time and support that he showered on me throughout the length of my research.

I am equally thankful to the entire faculty members of CSED for their direct and indirect help and cooperation.

Last but not least, I would like to thank my parents and friends for their encouragement and support. They have always wanted the best for me and I admire their determination and sacrifice.

Above all, I would like to thank the Almighty for the kindness who blessed me during this journey.

(Vipul Moudgil)

ABSTRACT

DEVELOPMENT in the field of communication networks gave the birth to a revolutionizing technology called Cloud Computing. Cloud Computing (CC) facilitate from corporate sector to start-ups, to use the on-demand IT services without concerning about the management, security and maintenance of the infrastructure. Establishing all these services using highly powered Cloud Data Centers (DCs), there is a considerable amount of energy consumption in CC environment that can't be neglected. Indeed, the current energy consumption analyzed by ICT (Information Communication Technologies) is estimated to be more than 10% of the worldwide consumption, expecting to elevate in upcoming years. These high end DCs utilizes the cloud energy to accomplish requests or tasks generated by the users. This energy consumption not only effects the economies of governments and companies, but also the planetary environment through carbon emissions. Recent research indicates that towards the ending of 2020, the carbon emission footprints will elevate by 20%, making the energy efficient cloud systems one of today's major challenge. This thesis basically attempts to design the possible efficient solutions which can be adopted to reduce these carbon footprints and practice a green cloud computing environment.

Keywords: ***Cloud Computing (CC) Environment, Data Centers (DCs), Clustering, Energy-Aware Scheduling, ANOVA, Tukey's HSD.***

TABLE OF CONTENTS

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS

CC	Cloud Computing
DCs	Data Centers
VMs	Virtual Machines
IT	Information technology
ANOVA	Analysis of Variance
EC2	Elastic Cloud Compute
VPN	Virtual Private Network
SOA	Service-Oriented Architecture
PAYG	Pay as You Go
NIST	National Institute of Standards and Technology
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
QoS	Quality of Service
API	Application Program Interface
SLA	Service Level Agreement
Tukey’s-HSD	Tukey’s Honest Significant Difference
CA	Cluster Analyst
CE	Cluster Engine
DM	Decision Maker
GUI	Graphical User Interface

CHAPTER 1
INTRODUCTION

Cloud Computing (CC), a strapping IT (Information Technology) archetype that provides the users with ubiquitous and convenient access to the on-demand IT resources and higher-level services with least management efforts. It uses well established distributed network Internet to communicate with the users and uses high end Data Centers (DCs) to accomplish various tasks. It helps the organization to use third party cloud services, in order to do core business rather than expending their valuable resources on maintenance and IT infrastructure. This chapter describes about the interesting concepts and latest technologies that CC uses to deliver the services to the end users.

1.1 History

The term "Cloud Computing" came into boom with the release of Elastic Compute Cloud (EC2) product by Amazon.Inc in 2006. It was revolutionized, when well-known computer scientist John McCarthy stated "computation may one day be used and organized as an essential part of public service utility system" [1]. He said:

> *"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility. The computer utility could become the basis of a new and important industry."*

Its arduous to identify the exact day of existence of CC, as it was formed by the accumulation of distinct technologies that established over years, like: Mainframes in 1970, Client-Server in 1980, the Web and Virtual Private Network (VPN) in 1990 and in 2000 the Service-Oriented Architecture (SOA) was developed. Technological advancements in the telecommunication services and networks architecture provided more power to Cloud computing and till the end of year 2000, CC came in existence. Many well established MNCs started investing the technology, and in August of 2006, the subsidiary of Amazon called Amazon Web Services, introduced its first Elastic Compute Cloud (EC2). Then on April 2008, Alphabet.Inc Google's parent organization released its first Google App Engine. During early 2008, NASA brought immense advancement with the product "Open

Nebula", under the project funded by European Commission, being the first ever open-source software developed for the deployment of hybrid and private clouds systems, followed by Microsoft Azure in 2010 and IBM Smart Cloud in 2011.

The notion "Cloud" in CC indicates the classic analogy of advance Internet technology, often denoted pictorially as a sky's Cloud. Technically Cloud works as a dedicated network of distributed computer aided resources either hardware or software) that are virtualized, making them remotely available to the active online user requests and self-managed using the dedicated network architecture via Internet [2-5].

1.2 Cloud Computing

The major grail of CC lies in facilitating the users to take benefits from all the advances IT technologies, without any need of deep knowledge or having any expertise in these technologies. The cloud archetype aims in cutting down the establishing and maintain value and letting the prime users to focus on the core business ideas rather than concerning about impeded IT obstacles. Now a day, CC has developed an important enabling technology for various applications and domains in recent years [6, 7, 8, 9]. It is an extensively acquired archetype with resource virtualization approach having service-oriented architecture. The archetype of CC is enough in fulfilling the crucial on-demand services of the tenants, taking in context the Service-Level Agreement (SLA's) signed between the tenants and service providers [10, 11]. The concept of acquiring and utilizing the distributed network resources not only expanded its desire but also exponentially elevated its application, services and management.

The development of services across the Web and the complications in dealing with the colossal and complex data, led to the emergence and success of CC archetype where service providers provides access to the collective computing resources such as computing hardware, storage units and networking to the remote users via the Internet technology. These computer resources are charged using Pay as You Go (PAYG) utility, which indicates, only registered clients have to pay for amount of selected resources that are provided during the period of use, and the same constraints applies for the billing also like the use of other utilities such as electricity, water, gas, etc.

Many authors define CC differently and little consensus is always appreciated in having a universally applied definition. But, this multitude of CC reflects the variety and applicability of the technological prosperity that CC technology has. In [4], the authors proposed the definition of CC by aggregating distinct technological elements to satisfy wide application of CC:

> *"Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs."*

However, National Institute of Standards and Technology (NIST) in [12] proposed the description of CC which became the most referred and widely putative definition in IT community.

> *"Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

The NIST definition concentrates on the prime features that CC offers from other existing IT services. The following figure, demonstrates the vital concepts and that enables the understanding of the CC terminology, including prime cloud service variants and its deployment models [12].

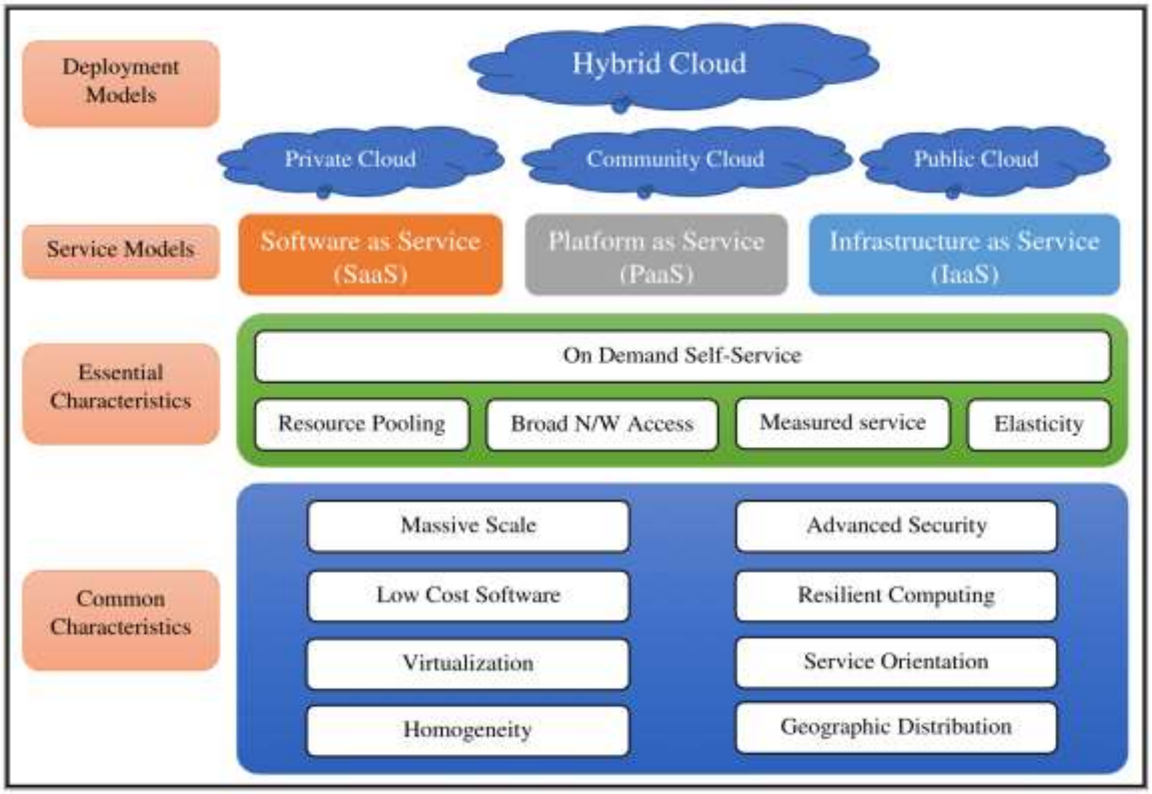

Figure 1.1 NIST Cloud Definition Framework [12]

The cloud compounds three service models with five essentially defined characteristics and four deployment models, which are described below.

1.2.1 Cloud Deployment Models

The deployment model, initially indicates the manner with which Cloud infrastructure is established and managed. As per NIST, the four main deployment models [12]:

1. **Public Cloud**: These clouds are made accessible to general public. The model where everyone has the provision and subscription to cloud services free and sometimes even pay for certain segments.
2. **Private Cloud:** Unlike the Public Cloud, the private Cloud are operated and established in a single private organization. The dedicated infrastructure can be either compered by primary organization or can authorize some "third-party" organizations. This type of Cloud archetype offers limited amount of facilities for

the defined clients, using the firewall technology that guarantees better control and management.

3. **Community Cloud:** This model has the ability to share services between several organization, so as to meet their specific demands such as collaborative mission, security, political etc. these clouds are designed to meet the general and high computational needs of a community and other existing individual companies.
4. **Hybrid Cloud:** This is the advanced combination of distinct Cloud archetype. An organization has the option of creating its own private CC archetype, which exploits some existing public cloud services.

1.2.2 Cloud Service Models

As we know that cloud archetype follows PAY AS YOU GO (PAYG) utility, Classification is done by the level and kind of service provided by the providers or customer takes, there are three foremost models depending on the administration level offered, initiating from the physical (hardware) to the highest level (Applications).

1. **Infrastructure as a Service (IaaS):** The physical but the most significant layer includes the services that provides with all the essential computing resources to users in form of Virtual machines (VMs). These VMs are hosted by the dedicated physical servers present in established DCs that fits to the IaaS service providers. Virtualization techniques allows us to consolidate the usage of physical server usage, by allowing multiple instances in form of VMs operating on the same server, further improving the degree of exploitation for the available crucial resources. Virtualization techniques play a significant role in preventing the interference between VMs that are owned by distinct users. IaaS providers have the responsibility for providing and hosting the virtual images of different platforms in form of VMs. The management of additional computational, storage resource components and inter-VMs communication functionalities in the virtual networks and assurance for the quality of service (QoS). Digital Ocean, Microsoft Azure, Amazon EC2 [13] are the leading IaaS providers.

2. **Platform as a Service (PaaS):** Provisioning of software platforms for the developers is done by PaaS providers, the services are delivered with dedicated programming interfaces often called (APIs) that have the ability to host and run multiple CC applications simultaneously. On using PaaS services, the work of application designers gets further simplified and manageable, as the complexity related to the infrastructure is by default hidden behind the API offered by the chosen PaaS provider. Thus, leading in the simplification for the process of developing a software service, an application designer who utilizes a PaaS platform only provides the PaaS provider with the source code of the application, in return it's the duty of the IaaS provider to manage and deploy it and make it accessible to the registered end-users. The PaaS model has the tendency to reduce the complexity for developing the dedicated Cloud applications, thus the source code has to obey the technological constraints that are imposed by the PaaS providers, such as the limited support for number of programming languages, the obligation usage of databases servers and some libraries. However, these constraints create convenience for the integration of the source code and the technology used in the infrastructure. Popular example of the PaaS services are: App Engine [14] offered by Google and Amazon Web services [15] offered by Amazon.
3. **Software as a Service (SaaS):** It is the highest layer or the application driven layer of Cloud Services. Capability of making service accessible to the end-users using thin client (Web Browser, Mobile application etc.). These services are accessible from distance; thus, the core part of the code is executed on the infrastructure of the SaaS service provider or a third party. The benefits of this service model are many, lies in the complexity for the deployment of essential services as well as maintaining aspects which are hidden from the end user. Hardware requirements are also adequate, as service execution is mostly done on the infrastructure level of the service Provider. The end-user has the full warranty of quality of service (Service Level Agreement) without even investing in specialized and high-performance hardware. Any type breach or an act of mismanagement to this agreement leads the provider pay for the user. As an example, Gmail and Yahoo mailing service [16] by Google and Facebook social network [17].

1.2.3 Essential Characteristics of Cloud

The general listed services are treated as the essential characteristics that are to be performed by the cloud archetype to provide the users with the resources [12].

1. **On-demand self-service:** Providing high computing capabilities to the consumer without any human interaction with the cloud providers like storage units, computational time etc.
2. **Broad network access:** Multiple capabilities and distinct option are also available in the dedicated network that are accessed using the prior defined standard mechanisms promoting the use of heterogeneous client platforms (e.g., tablets, laptops, mobile phones, and workstations).
3. **Measured service:** Cloud archetype is intelligent enough to automatically manage and optimize hardware and software resource utilization, by borrowing a metering capability at different levels of abstraction that are significant to a particular service like, storage, bandwidth, processing units and online user accounts managements. The resource utilization can be controlled, supervised and stated by maintaining transparency between both the provider and client according to their service usage.
4. **Rapid elasticity:** The capabilities can be released and provisioned elastically, sometimes even automatically, to rapidly scale the resources and communication with demand. The provisioning of the computational capabilities designed for the customers often appear as unlimited but these can be altered and augmented in any form and quantity at any particular time.
5. **Resource pooling:** With the use of multitenant computing model, the providers give access to the remote computing resources as per the consumers demand by dynamically allocating and reassigning distinct physical and virtual resources. The sensing of location independence plays a significant role as customer, generally has no clue and knowledge about the exact location from where the resource are provided.

1.3 Virtualization

After having an overview of different deployment models in CC, this section addresses a very salient concept of CC, the idea on which the whole CC environment operates to execute the users demands, called “Virtualization”. [18, 19] Virtualization acts as the main powering technology for CC environment. Virtualization is a concept of separating a physical computing device into one or more "virtual" devices using dedicated software as a base, each of which can be easily used and managed for the execution of the computing tasks.

The vitality of the technique lies in stimulating the behavioral aspects of the physical machine by using a software tool called hypervisors. The software tool running VMS came into sound first time in 1960, as a program having the capability of stimulating the whole computer hosting an operating system which consist of similar key components as that of the dedicated computer such as GPU, CPU, network, storage units etc. VM is a computer program that is temporarily running on the hardware of the machine that has the ability to run multiple VMs together.

Supremacy of the VMs lies in providing high level of abstraction to the hardware components and allowing the heterogeneous hardware to execute the services. It also provides inter-process isolation for the processes running between or within multiple VMs. Which is best process security policy, for the services that belongs to different customers but are running on the same infrastructure.

In IaaS systems, VMs act as the most popular virtualization technique. Installation of a complete operating system is necessary for the operation of each VM. An alternative advanced technique called Containers recently became immensely prevalent. They are also called light VMs. The containers take advantage providing inter-process isolation (using chroot command) of modern era operating systems by giving users an impression, that their services are running on operating systems in abstraction to other, whereas in reality they run in different environments, but have the same dedicated machine that belongs to prime server. Docker’s [20] success made container technology more popular and are attracting many IaaS managers which earlier used PaaS providers.

1.4 Clustering

Clustering plays a significant role while doing data analysis. The main purpose of clustering techniques is to fetch useful information from large data sets. Clustering analysis partitions the data into some sort of logical groups before analyzing it. In the simplest form, clusters are set of data points which share similar characteristics and clustering algorithms categorizes the data values (represented in points) in form of clusters according to their characteristics, data points with similar attributes are assigned one cluster and the data points with dissimilar attributes belong to different clusters. Cluster analysis clusters the data points based only on historic evident information found in provided datasets that describes relationships of points among each other. Clustering is measured as the most important "*unsupervised learning"* problem; it pacts with finding a significant *structure* in a assortment of unlabeled data points. In simple words, the aim is to separate out groups having similar features and assign them into a common cluster.

1.4.1 Aim of Clustering

The aim of clustering lies in determining the fundamental grouping of the given unlabeled data. But how to make sure what type of clustering significant clustering? It's fore sure that there is no complete "best" criterion which would be sovereign of the concluding aim of clustering. It is totally dependent on the users that what kind of the clustering technique provides him with the desired results.

For illustration, we could be interested in just finding the representatives for distinct homogeneous groups for the *data reduction purpose* and finding the clusters natural formed. which defines their unidentified possessions (*"natural" data types*), for discovering the valuable and appropriate groups (*"useful" data classes*) or in finding unusual data objects (*outlier detection*). Focal goals of clustering are:

1. To minimize the Intra-cluster distance
2. To maximize Inter-distance cluster

1.5 Forms of Clustering

Clustering is categorized into different types based on different criteria. Clustering is broadly identified in two types: Partition clustering (K-Means Clustering) and Nested

Clustering (Hierarchical Clustering), these techniques embeds a wealth of subtypes of clustering, which are described as below:

1.5.1 Partition Clustering

Partitional clustering is performed by unswervingly disintegrating the given dataset into distinct clusters. These clustering algorithms are based on the criteria to minimize the accent of the existing structure of dataset, this is achieved by assigning clusters to peaks in the probability density function. They minimize the amount of dissimilarity among the samples present in each cluster entity, while maximizing the dissimilarity of different clusters. The most frequently practiced partitional clustering technique is K-Means clustering technique.

1.5.2 Hierarchical Clustering

Hierarchical clustering, on the other hand, proceeds in successive iterations by integration smaller distinct clusters into bigger clusters, or by disintegrating bigger clusters into smaller clusters. This approach varies by certain regulation by which a decision is taken, that on what basis will the two clusters be integrated together or a big cluster will get disintegrated. The resultant forms a finite tree structure of the clusters, often called as a dendrogram, which indicates the relationship of the clusters. By dissecting the dendrogram at a desired level the significant clusters of data items in distinct groups is attained.

1.6 Clustering Algorithms

There are many clustering techniques in literature [21] providing with different types of division of data set, but K-Means and Hierarchical clustering techniques are widely used so in this thesis these two clustering techniques are discussed below in detailed manner.

1.6.1 K-Means Clustering Technique

The K-Means clustering technique being extensively used because of its uncomplicatedness and good efficiency in distinct applied practical fields. K-Means technique clusters the observations in predefined number of distinct clusters. “K” in k-means indicates the number of clusters to be formed for the given dataset. Numerous distance measures and techniques exist to determine the membership of an observation to be appended in a cluster. Algorithm aims reducing the distance between the selected observation and the centroid of the cluster by iteratively appending the observation to different clusters and terminating when the lowest distance measure is attained. The steps of the algorithm is as follows [21]:

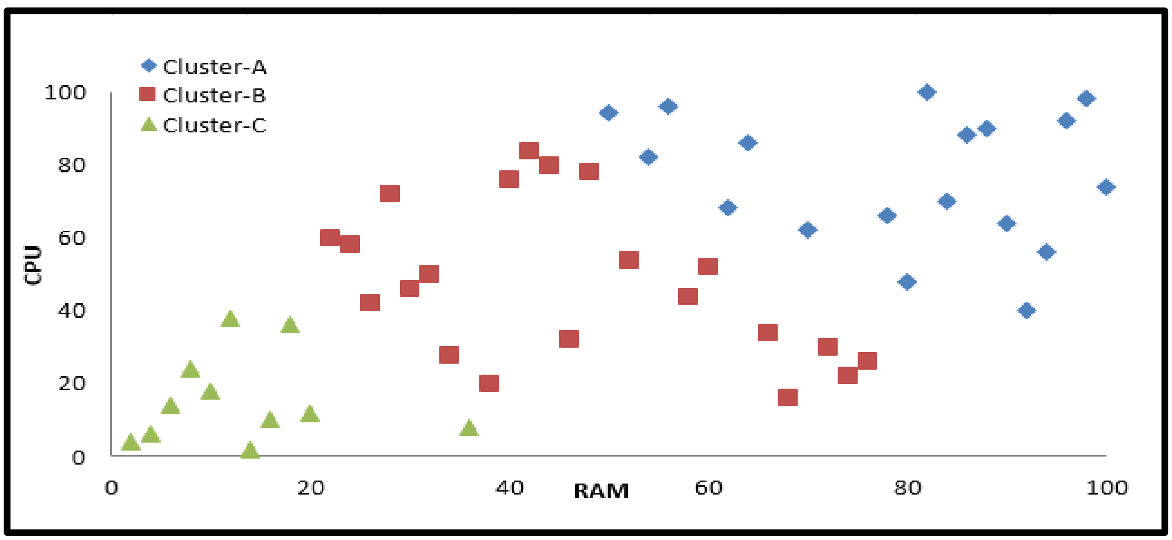

Figure 1.2 Example of K-Means Clustering

Step 1: Label the dataset $x_1, x_2, \dots, x_n$ into k clusters, where k is a user-defined bound that symbolizes the number of clusters in the dataset.

Step 2: Select centroids $c_1, c_2, \dots, c_k$ at random locations.

Step 3: Allocate entities to their nearby cluster center according to the *Euclidean distance* function. Repeat until convergence.

a) For each point x_i :

- Find nearest centroid $c_j \operatorname{argmin}_j D(x_i, c_j)$
- Allocate the point x_i to cluster j

b) For every cluster j = 1, 2, . . ., K: $c_j(a) = \frac{1}{n_j}\sum_{x_i \to c_j} x_i(a)$ for a = 1, 2, . . . ,d

- New centroid c_j = average of all points x_i allocated to cluster j in earlier step

Step 4: Repeat Step 2 and Step 3 until the centroids stop moving.

1.6.2 Hierarchical Clustering Algorithm

Hierarchical clustering is a nesting clustering algorithm which strive for to make a hierarchy of clusters. Hierarchical clustering algorithm is of two types [21]:

1.6.2.1 Divisive Algorithm: This is a top down approach where a single cluster is assigned to all of the data points and then partitioned into two least similar clusters. Separations are executed recursively as one move down the hierarchy and in the end we are left with one cluster for each data point.

1.6.2.2 Agglomerative Algorithm: This is a bottom up approach where one assigns each data point to its own cluster and then the two most similar clusters are joined. Merging of clusters is performed recursively as one move up the hierarchy and finally we are left with single cluster.

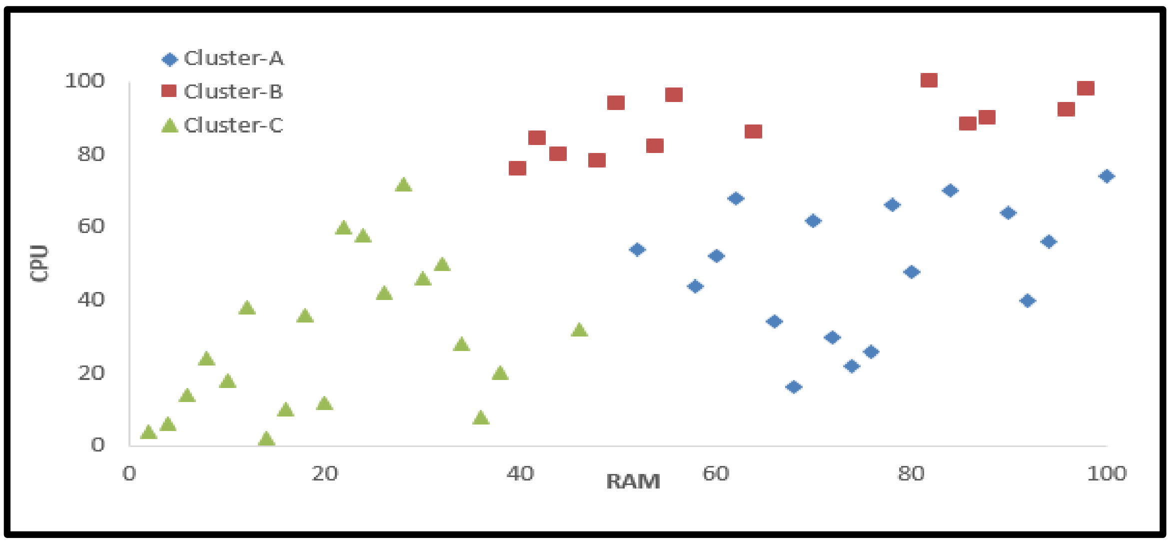

Figure 1.3 Example of Hierarchical Clustering

The hierarchy of clusters produces a tree, called dendrogram, which represents the hierarchy of the clusters. It can be easily interpreted from the above figure that both clustering algorithms are just reverse of each other. So, we are covering the Agglomerative Hierarchical Clustering algorithm in detail.

Agglomerative Hierarchical Clustering Algorithm: Given the set the *N* data points $(x_1, x_2, \dots, x_N)$ and each data point has its own clusters so we have *N* clusters as well. The distance or proximity or similarity matrix *D* of *N***N* size represents the distance between different clusters by using the metric $d[\, i, j\,]$, denotes the distance among two clusters *i* and j

The steps of the algorithm are as follows:

Step 1 Begin with N disjoint clusters

Step 2 Find the least unlike pair of clusters in the present clustering, say *r* and *s*, then

$d[r, s]$ = minimum $d[\, i, j]$

i.e., the distance between clusters r and s is minimum.

Step 3 Combine the clusters *r* and *s* into a single cluster say (p), $(p) = \{\, r\, , s\}$

Step 4 Update the distance matrix *D* by deleting the rows and columns corresponding to clusters

r and *s* and adding the row and column corresponding to new formed cluster (p).

Step 5 If all data points are in one cluster then stop, otherwise go to Step 2 and repeat the process.

So, here the clustering is performed on the basis of distance matrix. Firstly, each cluster has only single data point so the distance is calculated by using a distance function (Euclidean distance, squared Euclidean distance etc.). After performing the first iteration, we get more than one points in clusters so there is need to find the distance between different clusters and the following different methods are used to find the distance between different clusters:

Single linkage: The distance between two clusters is defined as the shortest distance between two data points in each cluster.

$$d[(r), (s)] = minimum\ d[(x_r, x_s)] \tag{1.1}$$

Complete linkage: The distance between two clusters is defined as the longest distance between two data points in each cluster.

$$d[(r),(s)] = maximum\ d[(x_r, x_s)] \tag{1.2}$$

Average linkage: The distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

$$d[(r),(s)] = \frac{1}{rs}\sum_r \sum_s d[(x_r, x_s)] \tag{1.3}$$

This hierarchical clustering provides best results in some cases and it does not require prior information regarding the number of clusters. However, it has some limitations too as it can never undo what is previously done, choice of adequate distance function out of many is a difficult task and it’s hard to recognize the accurate number of clusters by the dendrogram.

1.7 Scheduling of tasks in cloud computing

Emerging technology like CC processes a huge amount of data generated by the tasks like computation, data storage, processing etc. Thus, scheduling of the tasks onto the VMs plays a vital role in the managing the tasks. There are scheduling protocols that are designed so as to lessen the switching time, enhancing the resource usage and also augmenting the server performance and throughput. CC has attracted enormous amount popularity by allowing the cloud applications to get executed in remotely configured DCs. Parallel applications show different behavior when run simultaneously like: decrease in CPU resources utilization, as there is an increase in parallelism. Jobs not scheduled accurately have the tendency to reduce the machine performance. Several algorithms & protocols are proposed so as to attain efficient scheduling in CC. Many authors mostly focus on regular monitoring in their defined protocol. Practically the monitoring region tends to be irregular in real-life scenario as the clouds are randomly deployed.

In CC, job scheduling is the biggest and crucial point to attain green computing Environment. An efficient job scheduling strategy aims at making the CC system energy efficient and also lower the response time of execution of submitted jobs with minimum possible time and resources utilization. There are diverse kinds of scheduling present, such as:

- **Static scheduling:** All the information regarding tasks and availability of resources is known in advance, subsequently the tasks are assigned with the resources [22].
- **Dynamic scheduling:** The allocation of the resources to the tasks is done at runtime by the scheduler. It has more flexibility compared to static scheduling. The overhead is more as compared to static scheduling [22].
- **Online mode scheduling:** In online mode scheduling, the resources are allocated to a job as soon as it arrives in the system, on the basis of available resources at that very moment of time by the scheduler [23].
- **Batch mode scheduling:** In Batch mode scheduling, resources are allocated to tasks in batches that is, initially tasks are pilled over a period of time and then the execution process starts after a specific time interval, also referred offline scheduling [23].
- **Pre-Emptive scheduling:** This scheduling has the option of interrupting the tasks during its execution and the jobs can also be migrated to another resource other than its actually allocated resources. [24-26].
- **Non Pre-Emptive scheduling:** In Non Pre-Emptive scheduling, resources don't have the liability to be re-allocated until the currently running tasks are executed. [24-26].
- **FCFS:** It is the most popular and basic scheduling algorithm, which works on the simple principle that task that arrives first in the system will be execute first then the priority is given according to the time of arrival.
- **Round-Robin algorithm**: This scheduling algorithm works on time slicing mechanism in which each task is provided with resources for a definite period of time. So as to eliminate the fact of starvation.
- **Min-Max Algorithm**: Min-Max algorithm follows the constraint of selecting the smaller tasks and executing it first. Then the priority is given to the task with higher resource utilization.

- **Max-Min algorithm**: Max-Min algorithm is the vice-versa of the Min-Max Algorithm it prioritizes bigger tasks then the smaller tasks for the execution.

CHAPTER 2
LITERATURE SURVEY

Management and classification of VMs are considered as the essential components so as to achieve energy efficient CC environment. Thus, clustering of the VMs on the basis of their resource utilization plays a significant role in the clustering mechanism. Many researchers showed their interest in developing different clustering techniques for the CC architecture. Canali and Lancellotti proposed multiple studies based on various automatic clustering methods [27, 28] which are used to improve the scalability of IaaS DCs [29-30]. However, existing solutions for automatic VMs clustering have limitations as they have high computational costs [31], or not able to provide quick results [28]. To overcome these limitations, a new technique, namely AGATE [32], was introduced by using the concept of centroids, which provides clustering results with accuracy for a subset of VMs, based on their CPU utilization. However, the most widely used technique for centroid based classification is K-Means clustering. However, the centroid based techniques have their own limitations such as- spherical and equal number of cluster formation, and repeated iteration which leads to inconsistency in results and more computational power consumption. Further the clustering process is accompanied by the energy aware scheduling process that schedules the tasks onto the VMs by keeping the energy consumption under a particular threshold.

To promote green CC environment there has been a wide exploration in designing energy aware resource management systems in order to control the power ingesting of servers. In the high paced world of today, demand for denser systems and high performance, giving rise to different technology-scaling subjects which are posing massive challenges for the server systems to manage their powering and cooling systems. Some authors have proposed techniques to reduce the crowning power consumption of servers curtailing the influence on performance [33,34]. Additional research has familiarized us with techniques to allocate the workloads and manage the system computational resources in such a fashion that workloads should be distributed evenly over a set of servers taking the power requirements in consider [35]. Many [34-37] have proposed techniques in order to lessen the power consumption of servers by managing the power distribution and energy costs as well.

It is not tough to forecast the power consumption of a server if it has specific and stable workload in relations to resource utilization. However, it is a challenging task to predict the power and energy consumption during multiple VMs running in it. As the VMs have interference with each other, a statistical approach [37] has been proposed for effective prediction of average and constant power consumption to the DCs with combined VMs based on visions obtained from detailed summarizing of several applications both separate and combined. Thereafter, a mechanism referred Soft Fuse [36] is introduced by using the existing statistical approach [37] which confines the power ingesting of a server by controlling the processor clocking capacity when power consumption is getting too high. However, this approach assumes that the targeted system has only one processor and there is no interference and confliction of resources which makes it less realistic as their suitability for modern multicore systems was a question.

An advanced power distribution setup was needed to be planned in order to provide server with required power as the crowning power consumption is the more important factor rather than average power consumption. So, an approach was proposed [35] by considering the peak power consumption that is directly related to energy consumption. Thereafter, less attention was to be paid on the energy consumption made by the servers and introducing an energy effectual VM to the server mapping techniques [38]. This method observes the energy consumption and performance for all the distinct possible mappings and figures out the most energy efficient allocation. This technique without taking the individual energy consumption of a VM tries to lower the overall energy efficacy of the cloud system. Ecosystem [39], is a prototype for energy-centric operating system, takes energy as the significant factor. This technique concentrates on the task priorities and energy budget of server which guarantee the batter life for scheduling. This proposed approach is applicable to conventional single-core embedded systems but cannot deal with the multicore issues.

Joule meter [40] being a software method by which tracks the energy consumed by the VMs in a dedicated server environment by monitoring the resource usage of VMs dynamically. The authors employed two models to estimate the total energy consumption. The first model, which calculates energy consumed by multiplying processor time with the processor's average power consumption. However, the oversimplification of this model has compromised with its accuracy. The second model is a sophisticated because it keeps the records of the power consumption for the entire system by using an integrated power. It observes the fluctuation in power consumption for the first 200 seconds when a new VM is created and evaluates the power consumption characteristics of newly created VM by using statistical techniques. Although, this model is comparatively more accurate than the first one but it will perform poorly if there is a lot of fluctuation in the workload of a VM and moreover it is based on assumption that for first 200s the etiquette of the VMs is same. Many authors have proposed models [41-43] to estimate the power consumption more accurately by scanning the internal activities like integer or floating-point arithmetic operations, memory accesses, etc. of the processing unit and demonstrates the power estimation models by observing of impact of internal activities.

Energy aware scheduling fetched the interest of many researchers over the past years. Wang *et al*. [44] proposed an algorithm in multicore environment to lessen the amount of energy consumed by parallel jobs without augmenting the overall time of execution. However, the approach does not consider any resource other than CPU for parallel operations. Afterwards, a supportive two-tier energy-aware scheduling approach [45] was proposed for tasks analysis in real-time which profits both the customers and cloud service providers but the resources are not fully utilized. Wu [46] introduced a VM scheduling algorithm to enhance the energy efficacy of DC by using statistical methods but due to the task dependency the cloud utilization lowered.

An optimization model [47] task scheduling is presented for to lessen energy consumption in cloud DCs. This model uses an integer linear programming problem to lessen energy consumption by concentrating on the task response time constraints and scheduling tasks to a minimum number of servers. Whereas, the homogeneous workload distribution requires a large number of servers so it may cause an uneven network load distribution.

Hossain *et al*. [48] proposed a scheme for the migration of VMs in order to lessen the energy consumed by the systems. The algorithm merges selected VMs with the appropriate ones and shuts down or halts the redundant machines. This algorithm was practically able to reduce the energy consumption by migrating VMs but may cause hindrance in the performance of services.

Szymanski [49] used the lately proposed the Future-Internet network along with a Maximum-Flow and Minimum-Energy routing algorithm at Global Scale CC systems to attain a level of energy-efficient communications. Cao [50] introduced a scientific workflow scheduling procedure to lessen energy ingesting and carbon release simultaneously providing with QoS. However, this technique of trailing low latency, low energy ingesting urgently may lead to number of errors.

CHAPTER 3
RESEARCH PROBLEM

3.1 Problem Statement

CC is one of the most popular technologies of the modern era, which provides seamless connectivity and services (smart transportation, e-health, smart energy management etc.) to the end users as per their demands. In CC, using virtualization, efficient resource utilization can be achieved which in turn increases the performance of any implemented solution in this environment with respect to the parameters such as throughput and delay. However, to provide the aforementioned services to the registered users, there is an exponential increase in the size and number of servers within the cloud DCs which further raises the electrical consumption by the DCs. Thus, to maintain a green computing cloud environment an efficient VM management and scheduling mechanism for modern DCs is obligatory. The basic criteria for achieving better management and classification of the VMs, is by using good clustering mechanism for identifying distinct VMs on the basis of their resources utilization. In addition to clustering, an energy aware scheduling algorithm which tries to counter the energy consumption of the DC by scheduling the tasks between the give energy consumption thresholds to process green computing.

3.2 Research Gaps

The earlier proposed Hierarchical and Centroid based techniques have limitations as these proposals cluster the data on the basis of distance between the elements (not according to the characteristics of the elements), which requires predefined number of clusters, initialization for cluster centroids, more iterations and spherical clusters. So, it is difficult to determine correct cluster membership of the element and due to repeated iterations, clusters membership changes frequently with the change in cluster’s centroid. The prior identified scheduling techniques counters the tasks under the single threshold with poses the extra VMs to accumulate even with lowest resource utilization. Thus, the identified research gaps are:

1. Most of the clustering techniques reported are centroid based in which the number of VM clusters are predefined.
2. Cluster allocation is highly dependent on the location of the data.
3. The existing proposals emphasize on arranging the data in a spherical structure with equal number of VM clusters.
4. Counter the VMs resource utilization under single threshold.

CHAPTER 4
ENERGY AWARE SCHEDULING

In this chapter, proposed statistical techniques that are used to obtain the optimized clustering results and scheduling of tasks on the basis of the resource utilization of the VMs running in the DC are discussed. Tukey's-HSD Based Clustering (TBC) and Energy Aware Scheduling mechanism, which provides high accuracy for clustering a set of VMs and maps the upcoming tasks onto the VMs in an energy efficient manner. In the proposed scheme we the two key attributes-CPU and RAM utilization, to cluster the VMs. VMs having different resource utilization are compared using the Normal Curve and Analysis of Variance (ANOVA) test along with the Tukey's HSD test for evaluating the membership of all VMs. Once the distinct clusters are identified the energy aware scheduling mechanism schedules the tasks in an energy efficient manner and is evaluated using Two-Way Cutoff scheduling mechanism with respect to the defined metrics, the performance of the proposed scheme is found to be superior in comparison to the other state-of-the-art competing schemes of its category.

4.1 Methodology

In the proposed research work, there are multiple VMs running in the DC with distinct resource utilization (CPU and RAM utilization in our case). In order to achieve green CC environment, we initially clustered the VMs using the concept of normal distribution accompanied by the bivariate analysis done using ANOVA and TBC (Tukey's HSD Based Clustering) technique. The significance of the clustering lies in defining the VMs in different clusters on the basis cumulative resource utilization. The clustering is compared with the existing popular techniques like K-Means and hierarchical clustering and the results found are more impressive and accurate.

After the classification of the VMs the Energy-Aware Scheduling process comes into play. Multiple sharp thresholds are set edging mostly at the cluster ends. The tasks are then mapped onto the running VMs in such a way that the overall utilization of the VMs is occupied in the defined threshold. Once all the VMs are vacated in the given threshold the cloud system is set to be in optimized state. Further the scheduling process is examined using the Two-Way cut-off scheduling technique.

The crux of the research is creating an energy efficient model in the CC environment, that delivers the users with the best and required services in an energy optimized manner. The proposed clustering technique is designed and experimented using the IBM SPSS termed as Statistical Package for the Social Sciences.

IBM SPSS is an ease of use software platform that offers users with advanced prior defined statistical analysis, built with vast libraries for machine-learning algorithms, open-source extensibility, text analysis and also having capability of integration with advance technology like big data and its dedicated deployment into applications. Advanced flexibility and scalability brands IBM SPSS easily accessible. It helps to allocate new opportunities for organizations, improve their efficacy and lower the menace in projects. The statistics included in basic software is:

- **Descriptive Analysis:** Frequencies, Cross tabulation, Descriptive ratio statistics, Explore.
- **Bivariate Analysis:** Means, Correlation (bivariate, distance, partial), t-tests, ANOVA, Bayesian, Non-parametric tests.
- **Prediction of groups:** Cluster analysis, Factor Analysis, Discriminant.
- **Simulation and Geographical Spatial analysis**
- **Prediction of Numerical outcome:** Linear Regression

Most of the features in SPSS software are accessible by using the pull-down menus with a single click or can be programmed using high level programming language. The advances 4th generation programming language are capable in producing desired output, enhanced simplification of the iterative tasks, and handling colossal and complex data for manipulations and analyses purpose. The composite applications are advised to be programmed syntactically as they aren't accessible using the drop-down menu. The command syntax generated by the pull-down menu interface is displayed in the output panel, although changes are to be made in default settings, so as create user visible syntax. Python programming extensions allows the users to access the information from the data dictionary and helps to dynamically build the command syntax programs to write command language subroutines. Python extension powers SPSS to execute any of the statistical analysis in R.

This chapter is organized as follows. The system model and system architect is discussed in Section 4.1 and Section 4.2 respectively. In Section 4.3, the proposed clustering technique is discussed, with data collection followed by testing of the data using ANOVA and Tukey's-HSD test. Pseudo Code is proposed in Section 4.4. In Section 4.5 energy aware scheduling and its proposed pseudo algorithm is discussed.

4.2 System Model

As CPU and RAM utilization shares linear relation with the energy consumed by them so CPU and RAM energy consumption can be estimated by linear regression equations, where CPU Energy Consumption (Ω) and RAM Energy Consumption (Ω') are dependent on the regressors CPU (Φ) and RAM (Φ') Utilization.

4.2.1 Linear Equations

The following linear regression equations represents the energy consumption of CPU and RAM respectively.

$$\Omega = \beta\Phi + \alpha \qquad (4.1)$$

Similarly,

$$\Omega' = \beta'\Phi' + \alpha' \qquad (4.2)$$

where, β and β' are the least squares estimators and α and α' are the intercepts of the respective equations.

4.2.2 Energy Consumption

The total energy consumption (φ_e) of a VM depends on its CPU and RAM energy consumption. It can be stated as:

$$\varphi_e = f\,(\Omega, \Omega') \qquad (4.3)$$

which can be expressed as the addition of CPU and RAM energy consumption:

$$\varphi_e = \beta\Phi + \alpha + \beta'\Phi' + \alpha' \qquad (4.4)$$

The system architecture explains the prototype scenario of the IaaS cloud infrastructure to attain the scalability through clustering process for management and supervision of the VMs. The diagram states the dedicated DC having multiple servers that runs thousands of VMs in it.

The further subsection explains the working of the system architecture that acts as the prototype for the main cloud archetype. The architecture contains different components like (Hardware, Server OS, Monitor Engine, Decision Maker, Task Scheduler, Virtualized Layers etc.) that make the maintenance and supervision of the tasks and VMs much more amiable process to perform.

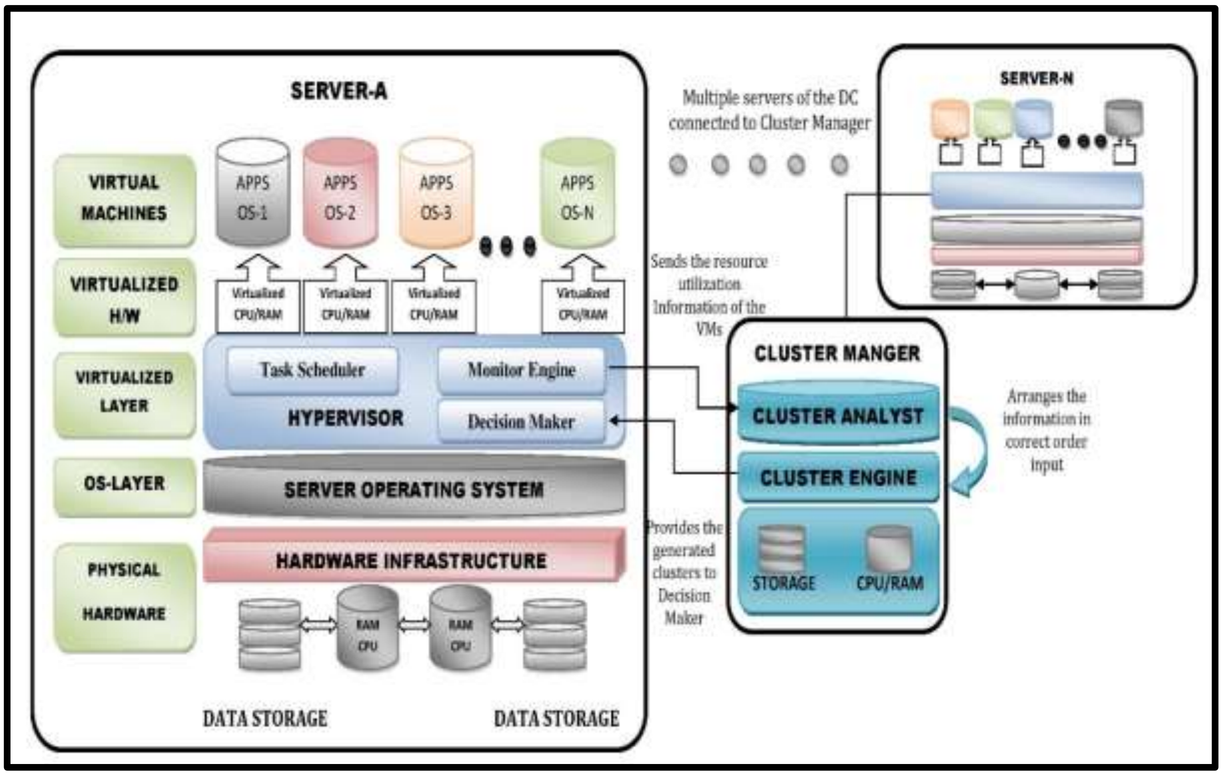

Figure 4.1 System Model

4.3 System Architecture

The system model operates in three phases.

4.3.1 Phase-I

As a server consolidates multiple VMs, it provides each VM with on demand resources for the requests made by the clients. The Monitor Engine (ME) installed on every server plays a key role as it diagnoses the resource utilization of each VM (CPU and RAM percentage utilization in our case) running on dedicated server. The ME collects the resource utilization of each VM and passes it to the Cluster Manager (CM) node.

4.3.2 Phase-II

The Cluster Manager is a global node that has resource utilization data of all the VMs from the servers running in a DC. The CM consists of two main components.

1. **Cluster Analyst:** The Cluster Analyst (CA) has all the data entries of resource utilization for every VMs running in the dedicated servers of the DC. CA arranges the data for the Cluster Engine (CE) in a correct order input, so that it become easy for the Cluster Engine to cluster the VMs.
2. **Cluster Engine:** The main component of the whole clustering process, in this segment the clustering is done mathematically using the power of ANOVA test and Tukey's HSD test. Depending on the obtained significant values, all the entities (VMs) are clustered in different classes.

4.3.2 Phase-III

It is the last phase of the process. Clusters formed by the CE are subjected to the Decision Maker (DM). On the basis of the clusters formed, the DM has the power to apply different operations such as task scheduling, task migration, VM migration in a DC. Thus, the clustering process increases the scalability, monitoring and decision-making capability of the DC.

4.4 Clustering Technique

Clustering of the VMs plays a significant role in cloud management and also serves as a base for the processes like VM migration, task scheduling and traffic management. Thus, using the powerful statistical techniques i.e. One-Way Analysis Of Variance (ANOVA) and Tukey's HSD Test (Post-Hoc test). we classify the presently running VMs of the DC into distinct clusters, depending upon their etiquette. The methodology consists of the following steps:

1. **Data Collection** in the form of tuples contain the percentage resource utilization of VMs.
2. **Testing** the tuples using ANOVA and Tukey's HSD.

3. **Clustering** them on the basis of their adjusted p-value obtained.

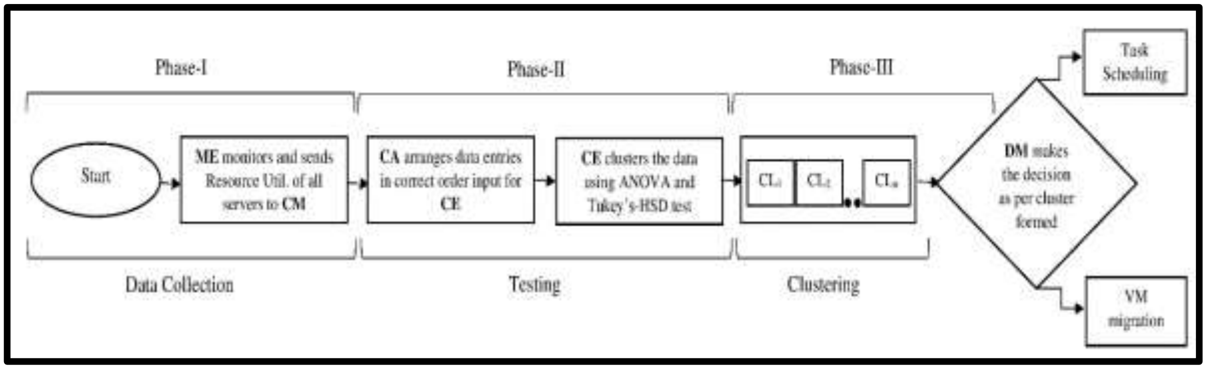

Figure 4.2 Working Flow Chart

4.4.1 Data Collection

The existing data in IaaS cloud DC can be collected and stored to analyze a pattern. To recognize the similarity of VMs there is need to measure the variation among the use of different resource (such as CPU and RAM utilization in our case).

Let S be the set of *n* VMs taken at an instance from IaaS cloud DC.

$$S = (\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_n)$$

Each VM (Φ_n) consist of *m* parameters represented as ($\theta_{i1}, \theta_{i2}, \theta_{i3}, \ldots, \theta_{im}$) for $i = 1, 2, \ldots, n$ where θ_{ij} represents the j^{th} observation of i^{th} sample and each parameter *j* of a VM has a distinct type and value. The number of parameters can be customized according to the accuracy, but all the VMs should be compared on the basis of same parameters and assumed to obey the Normal Distribution. As the comparison is made on the basis of same parameters for each VM thus number of observations also remain same for each sample to perform ANOVA test. Mathematically each VM Φ_i, i = *1, 2, . . . , n* is treated as a sample belonging to similar or distinct population (class), and there need to identify the class, so that the samples belonging to the identical population may be clustered together.

4.4.2. Testing

Before using any statistical techniques on the data, fractional percentile ranking procedure is carried out on the data. Fractional percentile ranking procedure helps us to build ample significant difference between the data items, so as to carry out the clustering process. Further the comparison of the n samples is made using the ANOVA technique, on the basis of multiple observations θ_{ij} , $j = 1, 2, \ldots, m$. The null hypothesis (H_0) tested by ANOVA states that the means μ_{Φ_i}, $i = 1, 2, \ldots, n$ of the populations, from which the samples are randomly drawn are all equal. Whereas alternate hypothesis (H_A) states that atleast one of mean is different from others. But we are keen in testing of null hypothesis

$$H_0 : \mu_{\Phi_1} = \mu_{\Phi_2} = \dots = \mu_{\Phi_n}$$

against the alternative hypothesis

$$H_A\colon \exists\ 1 \le i\ , l\ \le n \text{ such that } \mu_{\Phi_i} \ne \mu_{\Phi_l}$$

(i.e., there is at least one pair with unequal means) The one-way ANOVA, assuming the test conditions are satisfied, uses the following F-statistic:

$$F = \frac{MS_{\Phi}}{MS_{S(\Phi)}} \tag{4.5}$$

where MS_{Φ} = mean squares between samples

$MS_{S(\Phi)}$= mean squares within sample (error)

Under H_0 this statistic has Fishers distribution *F (n-1, N-n)* with *n-1*and *N-n* degrees of freedom for samples and error respectively, and total number of observations is denoted as *N*. In case it holds for the test criteria $F > F_{n-1,\ N-n,\ 1-\alpha}$ where $F_{n-1,\ N-n,\ 1-\alpha}$ is $(1 - \alpha)$ = quantile of F-distribution, then hypothesis H_0 is rejected on significance level α [51]. If F-statistic outweigh the critical value of F then one can reject the null hypothesis, otherwise we fail to reject the null hypothesis. When the null hypothesis gets rejected using the F-test in ANOVA, then difference among the means is to be apprehended. If our analysis fails to reject the null hypothesis, then it means that all the samples are drawn from same population, forming a single cluster. Rejection of null hypothesis using the F-test in ANOVA, means that samples are drawn from more than one populations so there is need to distinguish the pattern of difference between means. To determine which pairs of means are significantly different, and which are not, a Tukey's Honest Significance Difference (HSD) test is widely used. The Tukey's HSD method controls type 1 error (it's the probability of incorrect rejection of the Null Hypothesis) very well and is generally considered an acceptable technique. Tukey's HSD procedure allows us to compare all obtained pairs of means. When used with equal sample sizes, the familywise error rate is exactly equal to α, which is usually set to 0.05. Choose an optimum VM Φ_p (with maximum utilization of resources i.e.

$$\Phi_p = \left\{\Phi_k \middle| \frac{1}{m} \textstyle\sum_{j=1}^{m} \theta_{kj} \ \ge\ \frac{1}{m} \sum_{j=1}^{m} \theta_{ij}\ , i = 1, 2, 3, \dots, n\ ; i \ne k\right\})$$

and compare the sample Φ_p to all the remaining samples Φ_i, i = $1, 2, . . . , n$; $i \neq p$ so that we can check which samples are significantly different from sample Φ_p and which are not. A statistical hypothesis test is used to lay down a comparison of each pair of means, μ_{Φ_p} and μ_{Φ_i}, $i = 1, 2, . . . , n$; $i \neq p$ where the null and alternative hypotheses are of the form

$$H_0 : \mu_{\Phi_p} = \mu_{\Phi_i}$$

$$H_A : \mu_{\Phi_p} \neq \mu_{\Phi_i}$$

Tukey's HSD test computes the highest significance difference between samples using a statistical studentized q distribution. Statistic q helps to evaluate the difference between two samples Φ_p and Φ_i:

$$q = \frac{\overline{\Phi_p} - \overline{\Phi_\iota}}{\sqrt{\frac{MS_{S(\Phi)}}{m}}} \tag{4.6}$$

with n and N - n degrees of freedom for α level of significance, $\overline{\Phi_p}$ and $\bar{\Phi_\iota}$ represents the mean of samples Φ_p and Φ_i respectively. Rewriting equation (1) shows that a difference between the means of sample Φ_p and Φ_i will be significant if

$$\left|\overline{\Phi_p} - \bar{\Phi_\iota}\right| \geq HSD = q_{n,\ N-n,\ 1-\alpha} \sqrt{\frac{MS_{S(\Phi)}}{m}} \tag{4.7}$$

Similarly, compare all the remaining samples Φ_i to sample Φ_p and categorize the samples which are significantly different from Φ_p and which are not, using the below mentioned procedure.

4.4.3 Clustering

Once the Testing is done using ANOVA and Tukey's HSD test, cluster formation of samples (VMs) is carried out and adjusted p-value plays a significant role in clustering samples that are similar to prime sample Φ_p (VM_p). The initial prime sample (Φ_p) is a hypothetical sample with maximum average of attributes. As we are comparing all the samples to prime sample Φ_p, so the samples having adjusted p-value 1 will be similar to prime sample Φ_p and will form one cluster. After the first cluster is formed, the selection of the second prime sample $\Phi_{p'}$ is a cardinal task. The second prime sample $\Phi_{p'}$ is selected from the population, but with the help of the table, which is generated during the formation of first cluster using Tukey's-HSD test. The sample from the population, having minimum p-value difference from the prime sample Φ_p, is selected as the next prime sample $\Phi_{p'}$ for the next cluster formation. If there occurs more than one samples that have the same minimum p-value difference, then any of the sample can be selected from them. During the formation of the $(i+1)^{th}\ for\ i = 1, 2, \ldots, n$ cluster, the samples that are already allocated to their corresponding cluster are not to be excluded during the testing (ANOVA and Tukey's-HSD test) process performed with non-initial prime sample $\Phi_{p'}$. But they all will follow the rejection criteria i.e. the samples will not be included in the cluster even if they have the p-value 1 if they are already a member of a particular cluster. Thus, the whole population has to participate in the testing process for each prime sample selected, but all the sample have to follow the rejection criteria during the formation of clusters. Testing and formation of clusters are to be performed simultaneously in a sequence for every prime sample. After the repetition of both the process for a while a scenario will occur where there will be no sample left to be selected as prime sample because it will already be a part of previously formed clusters, thus the whole process is to be terminated. All the population will be defined under distinct clusters

4.5 Algorithm: Pseudo Code

1) **Input:** VM-List$_t[\alpha_n, \beta_n]$ VM-List$_c[\alpha_n, \beta_n]$

   SET k = 1

2) **Output:** Cluster-List$_k$[VM′s]

3) VM$_{opt}[\alpha_0, \beta_0]$: tuple contain the optimal value of parameters also called the Prime Sample Φ_p

4) **While** VM-List$_t[\alpha_n, \beta_n] \neq$ NULL, **do**

5) Tuk-List( ) = Tukey-HSD (VM$_{opt}[\alpha_0, \beta_0] \rightarrow$ VM − List$_m[\alpha_n, \beta_n]$)

6) **For each** VM in Tuk-List( ) **do**

7) **If** φ[VM$_m[\alpha_n, \beta_n]$] = = 1 and do not belong to any cluster-List$_k$ **then**,

8) Cluster-List$_k$.add (VM $[\alpha_0, \beta_0]$)

9) Cluster-List$_k$.add (VM $[\alpha_n, \beta_n]$)

10) VM-List$_c$.remove (VM $[\alpha_n, \beta_n]$)

11) VM-List$_c$.remove (VM $[\alpha_0, \beta_0]$)

12) **Return** Cluster-List$_k$[VM′s]

13) **If** VM-List$_c[\alpha_n, \beta_n] \neq$ NULL, **then**

14) **do** k = k+1,

15) **Select** new. VM $[\alpha_s, \beta_s]$ **From** Tuk-List( ) **Where**,

16) [1 − φ(VM$[\alpha_n, \beta_n]$)] = **MIN**.

17) **Set** new**.** VM $[\alpha_s, \beta_s]$ = = VM$_{opt}[\alpha_0, \beta_0]$

18) **Goto** step **4** repeat.

19) **else**

20) **Return** All Cluster-List$_k$[VM′s]

Where,

- ❖ VM-List$_m[\alpha_n, \beta_n]$ have the VM-Lists initially, VM-List$_t$ participate in testing process and VM-List$_c$ participate in clustering process.
- ❖ φ represents the significant or (p)-value.

4.6 Energy-Aware Scheduling

Energy aware scheduling algorithm works on the principle of allocating the task onto the VMs in a fashion that the overall consumption of independent VMs and the whole CC architecture is countered within the selected thresholds. The significance of the scheduling algorithm lies in having multiple thresholds, rather than the prior defined scheduling techniques that used just single threshold to arrange the tasks. The threshold can be manually set by the user or the administrator, but we suggest to use the proposed clustering technique so as to get better results. While using the proposed clustering technique the point of cumulative point of change of the cluster should be treated as the cut-off or the point where the data elements shows membership for the different clusters.

After the selection of the cut-offs is made the most essential part of the scheduling algorithm comes in play i.e. finding the best fit VM for the tasks so that the total consumption of the VM after the allocation of the task do not exceed the given cut-off. In some case there may be the tasks that initially have the high resource utilization requirement, so in that case the cut-off are ignored and the task is allocated to the vacant VM. But for most of the cases the tasks are scheduled according to the cut-offs. The VMs that have the utilization less than the smaller cut-off are often desired to halt or suspend so as to save the energy.

4.7 Algorithm: Pseudo Code

1. **Input:** EN_{conspt}-$List_t[\alpha_n, \beta_n]$ Task-$List_c[\alpha_n, \beta_n]$

   SET k = 1
2. **Output:** Scheduled tasks onto VMs
3. $EN_{opt}[\alpha_n; \beta_n]$: tuple contain the optimal value of parameters also called the prime Sample $€_p$
4. **While** EN_{conspt}-$List_t[\alpha_n, \beta_n] \neq$ NULL, **do**
5. Pro_{Clust}-List() = Pro($EN_{opt}[\alpha_n, \beta_n] \rightarrow EN_{conspt}$-$List_t[\alpha_n, \beta_n]$)
6. **After** the ENs cluster formation, mapping the tasks onto perfect ENs.
7. **Arranging** all ENs in increasing or decreasing order of their mean consumption.
8. **Setting** up the cut-o_s at mean of consumption cluster boundaries.
9. **For** each task in Task-List( ), **do**
10. **Select (** Cut-$off_m[\alpha_n]$) = MIN and MAX or user defined interval.
11. **Find** $Task_m[\alpha_n, \beta_n] \rightarrow EN_{conspt}[\alpha_n, \beta_n]$
12. **Where** $Task_m[\alpha_n, \beta_n] + EN_{conspt}[\alpha_n, \beta_n]$, lies
13. **Between** MIN (Cut-$off_m[\alpha_n]$) $\leq EN_{conspt}[\alpha_n, \beta_n] \leq$ MAX (Cut-$off_m[\alpha_n]$)
14. **Look up** for further tasks.

CHAPTER 5
EXPERIMENTAL AND COMPARITIVE RESULTS

As discussed in chapter-4, we try to achieve energy optimized CC environment using the proposed clustering and Energy-Aware scheduling process. we are targeting the generic Cloud archetype i.e. IaaS, it is vital to test the analysis on the large-scale virtualized DC, but as its immensely hard to conduct the repeatable and regular large-scale experiments. Thus, we render the data from google traces and apply data cleaning according to our use. We compounded DC that has distinct VMs running on it, having variational resource utilization (CPU and RAM). Total of 50 VMs running at the time T0, thus we intend to cluster these VMs into distinct cluster depending to their resource utilization (CPU and RAM). The VMs run a web-application or other application having variable workload in order to create a difference in their resource utilization to validate the clustering process. The CPU and RAM have the range of utilization from 0% to 100%, The utilizations are taken as the input attributes to the algorithm and the clustering process is carried out accordingly. The VMs with same etiquettes (having significance value 1) as the 50th VM (selected optimum VM), are clustered together.

The experimental analysis consists of two main components:

- Experiment 1: Clustering Analysis
- Experiment 2: Energy-Aware Scheduling Analysis

5.1 Experiment 1

Experiment 1 has been performed using the SPSS software platform. VMs with distinct resource utilization are taken into the input. Fractional percentile ranking is applied on the data so as to create significant difference among the data points (VMs). The VM having the highest resource utilization is choose as the prime VM. For initiating the clustering process all the remaining VMs are compared with the prime VM using the ANOVA and Tukey’s HSD post-hoc test. The VMs having the same *p-value* belongs to the same cluster. The further clustering process is carried out by choosing the next prime VM and comparing ten remaining VMs with it.

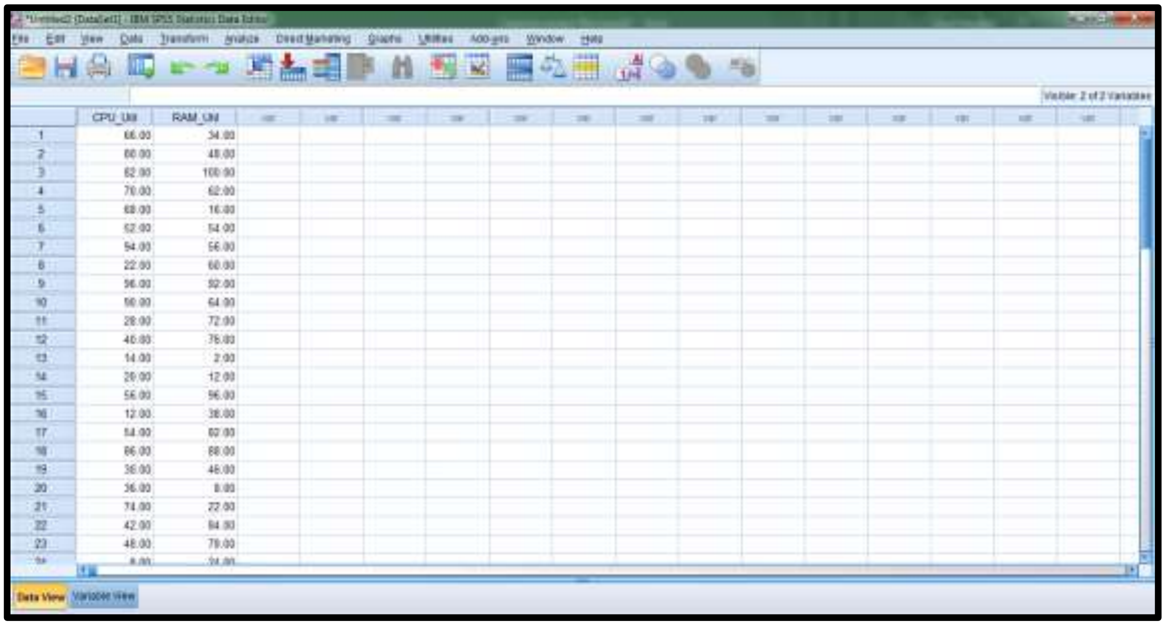

Figure 5.1 GUI (Graphical user interface) of SPSS

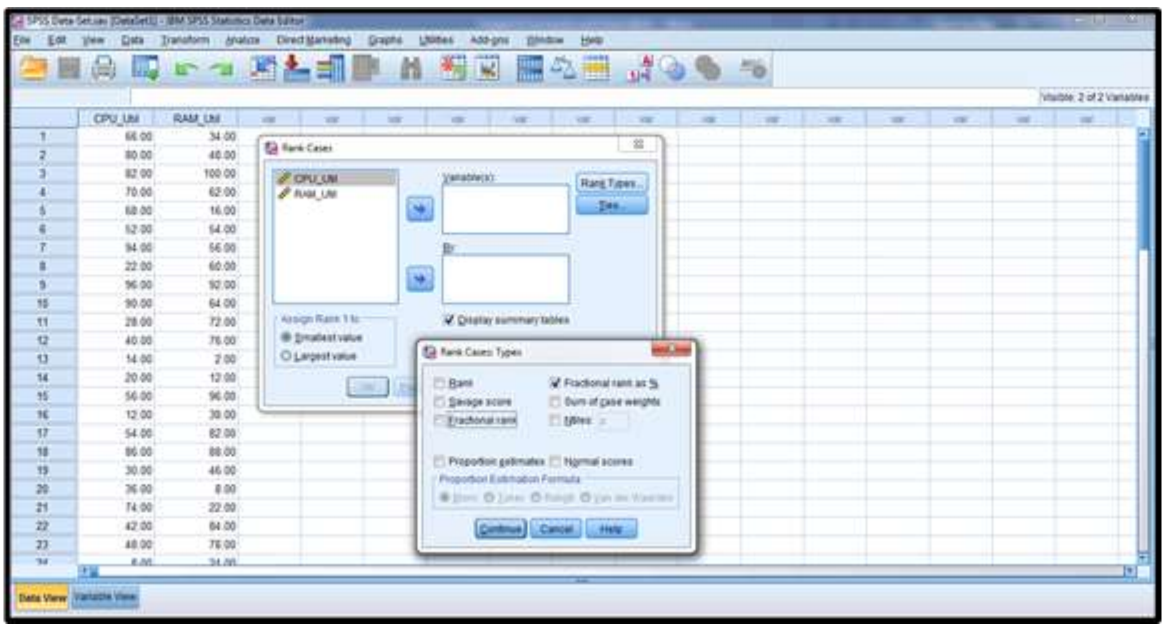

Figure 5.2 Fractional Percentile Ranking of VMs resource utilization dataset

The results of the proposed clustering are compared with the prior existing K-Means and hierarchical clustering techniques. The steps of the clustering mechanism are as follows:

Step 1: To arrange the data in a particular order and to create significant difference between the data items choose the **"Transform"** dropdown menu and select **"Rank Cases"** followed by **"Fractional rank as %".**

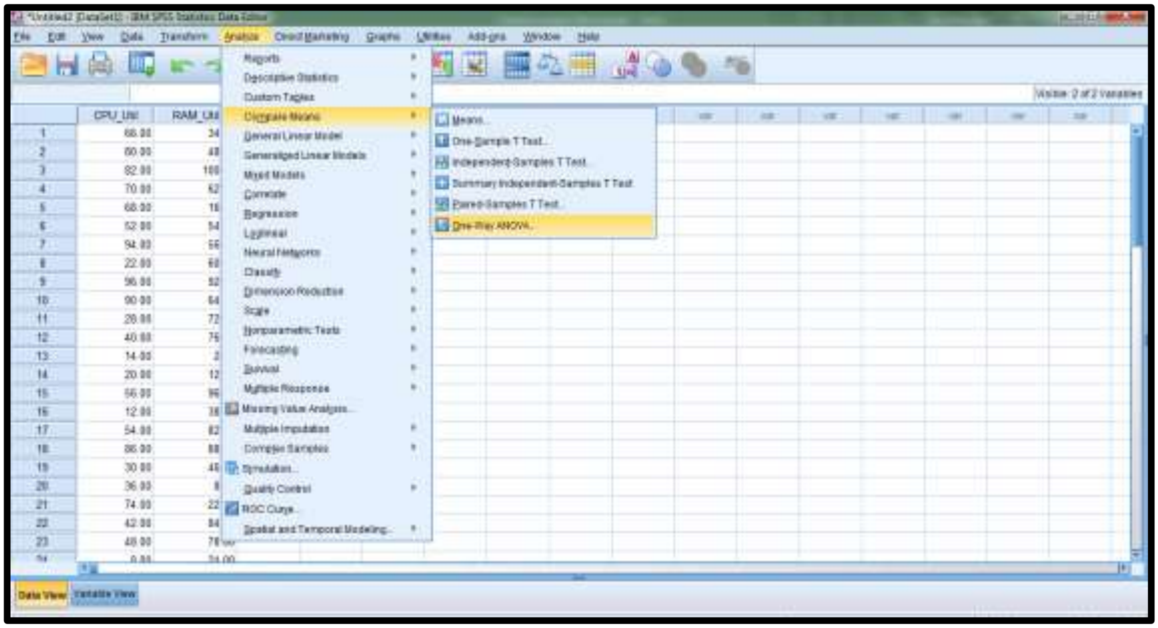

Figure 5.3 Bivariate ANOVA statistical analysis

Step 2: Apply the ANOVA statistical on the Fractionally ranked data using **Analyze** dropdown menu and selecting **Compare Means** followed by **One-way ANOVA test.**

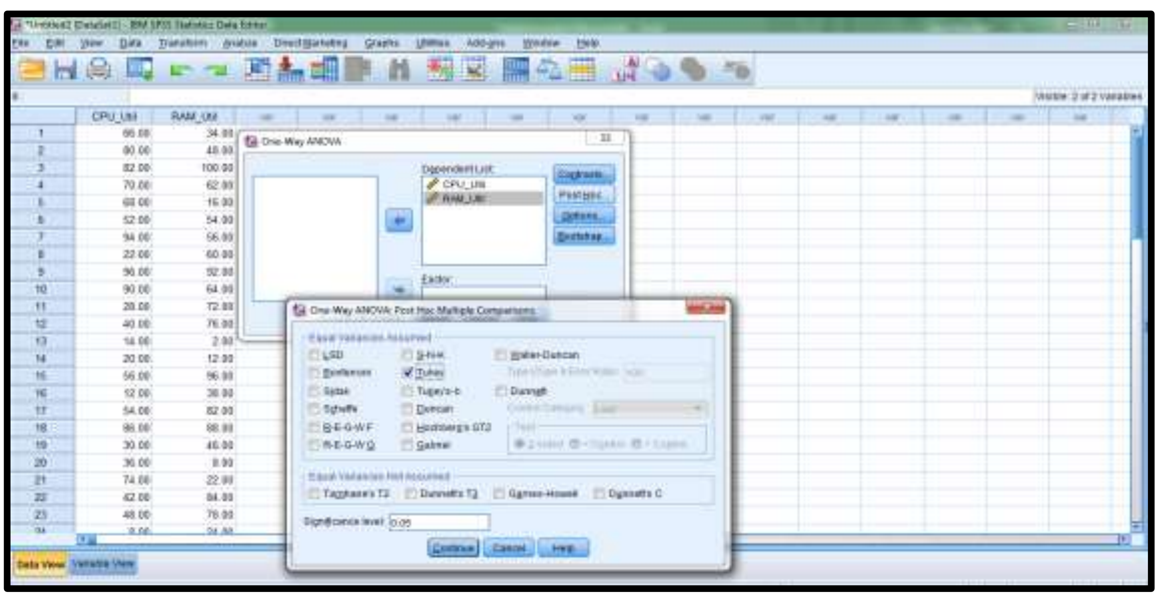

Figure 5.4 Tukey's HSD post-hoc test

Step 3: Analyze the data using the Tukey's HSD post-hoc test to find the significant difference between the given VMs on the basis of obtained "*p-value*".

Table 5.1 Tukey's HSD Table Readings

Dependence (i)	Dependence(j)	Mean Difference	Std. Error	Sig-Value	Lower-Bound	Upper-Bound
50	1	48.00000	19.15202	.913	-33.5195	129.5195
	2	34.00000	19.15202	1.000	-47.5195	115.5195
	3	7.00000	19.15202	1.000	-74.5195	88.5195
	4	32.00000	19.15202	1.000	-49.5195	113.5195
	5	56.00000	19.15202	.681	-25.5195	137.5195
	6	45.00000	19.15202	.958	-36.5195	126.5195
	7	23.00000	19.15202	1.000	-58.5195	104.5195
	8	57.00000	19.15202	.644	-24.5195	138.5195
	9	4.00000	19.15202	1.000	-77.5195	85.5195
	10	21.00000	19.15202	1.000	-60.5195	102.5195
	11	48.00000	19.15202	.913	-33.5195	129.5195
	12	40.00000	19.15202	.992	-41.5195	121.5195
	13	90.00000*	19.15202	.014	8.4805	171.5195
	14	82.00000*	19.15202	.047	.4805	163.5195
	15	22.00000	19.15202	1.000	-59.5195	103.5195
	16	73.00000	19.15202	.152	-8.5195	154.5195
	17	30.00000	19.15202	1.000	-51.5195	111.5195
	18	11.00000	19.15202	1.000	-70.5195	92.5195
	19	60.00000	19.15202	.531	-21.5195	141.5195
	20	76.00000	19.15202	.105	-5.5195	157.5195
	21	50.00000	19.15202	.869	-31.5195	131.5195
	22	35.00000	19.15202	.999	-46.5195	116.5195
	23	35.00000	19.15202	.999	-46.5195	116.5195
	24	82.00000*	19.15202	.047	.4805	163.5195
	25	21.00000	19.15202	1.000	-60.5195	102.5195

The above table demonstrates the analysis obtained after applying Tukey's HSD post-hoc test. The independent (j) VMs that have the Significant or *p-value* as "1" are similar to the dependent machine (i). Thus, they belong the same cluster. The independent VM having the *p-value* nearest to "1", is treated as the next dependent prime VM. If there exist more than one VM having the *p-value* nearest to "1", then any of them can be selected as the prime VM.

5.1.1 Comparison

Results obtained by comparing the proposed technique and the prior existing K-Means and Hierarchical clustering. The below demonstrated graphs depicts the significant difference between the approaches.

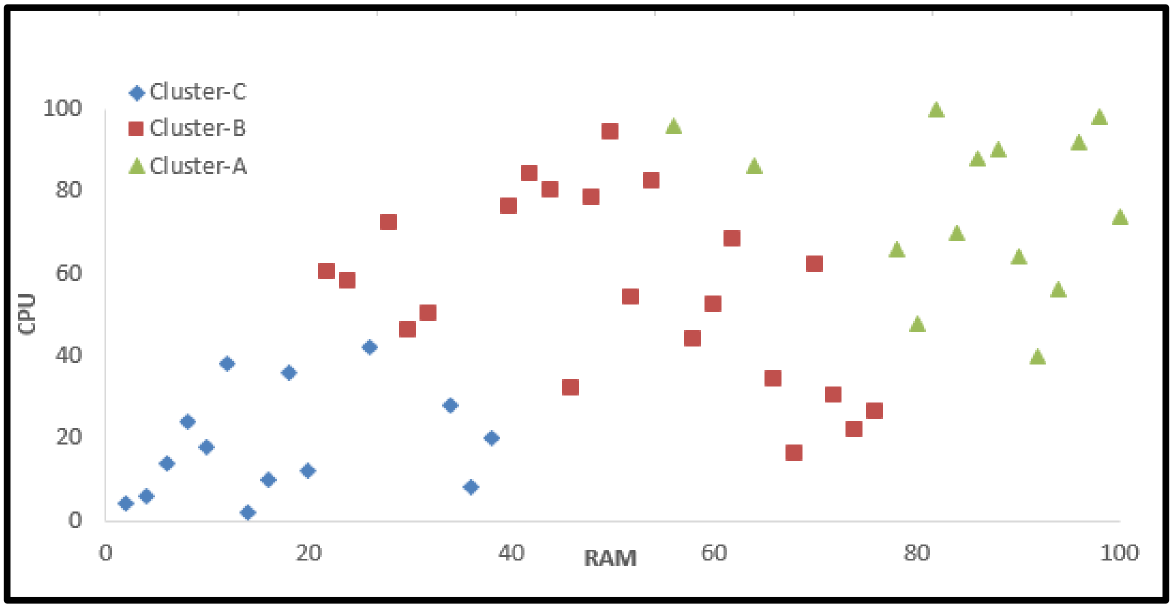

Figure 5.5 Proposed Clustering

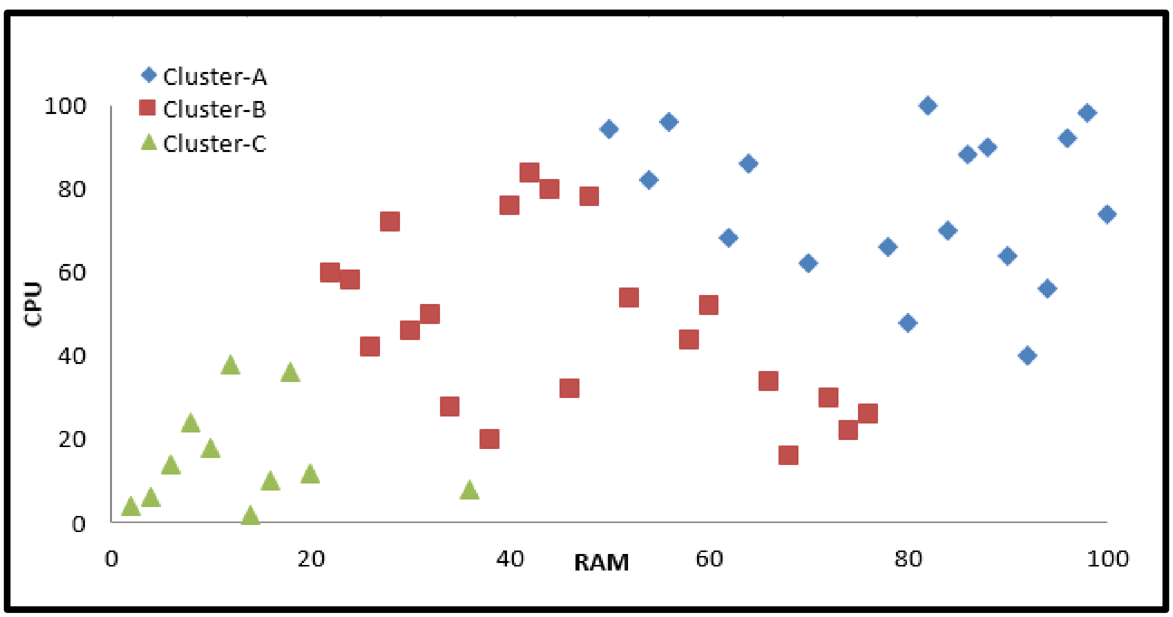

Figure 5.6 K-Means Clustering

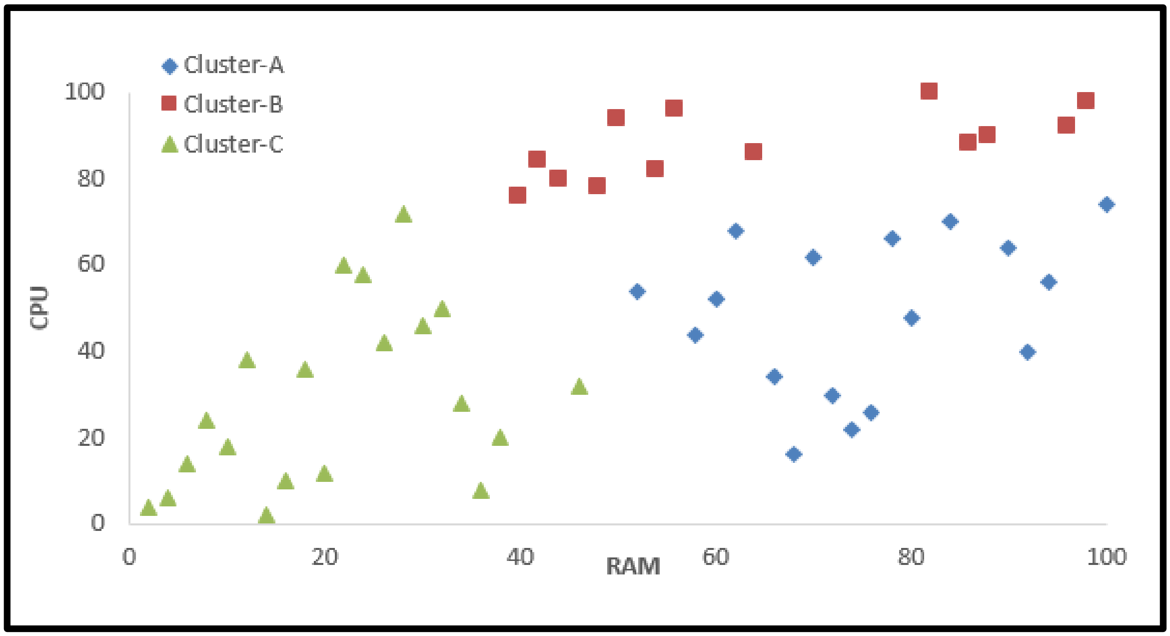

Figure 5.7 Hierarchical Clustering

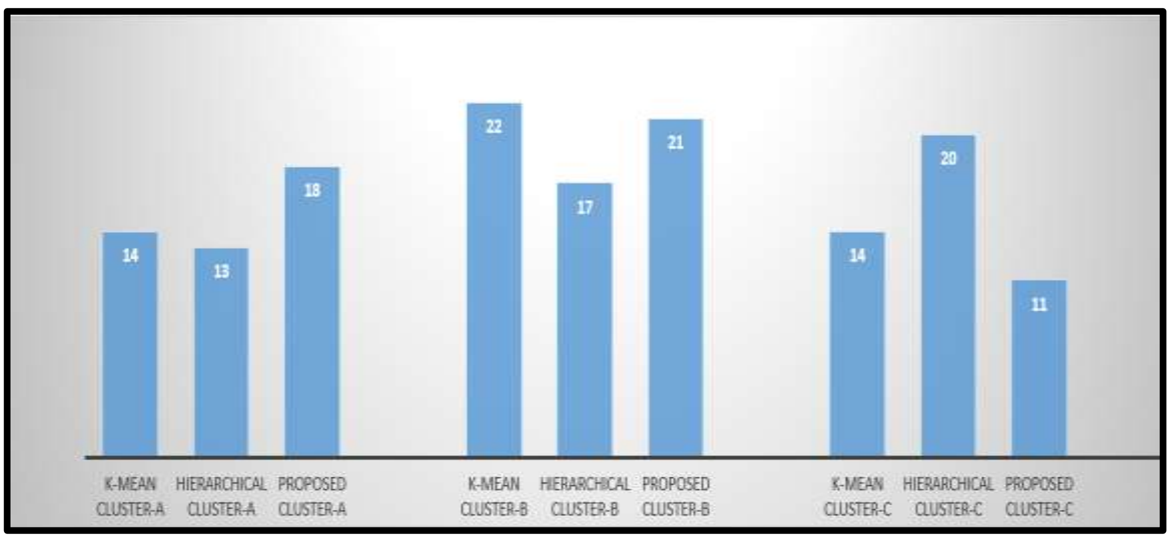

Figure 5.8 Comparison of number of elements in the cluster

The above graphs demonstrate the distinct cluster formation among the different clustering techniques. Further to explain the significant difference between the techniques, we use the cumulative utilization band analysis that helps us depict the issues and drawbacks of using K-Means algorithm and Hierarchical clustering algorithm. The cumulative utilization will also help to identify the point which were allocated in wrong clusters.

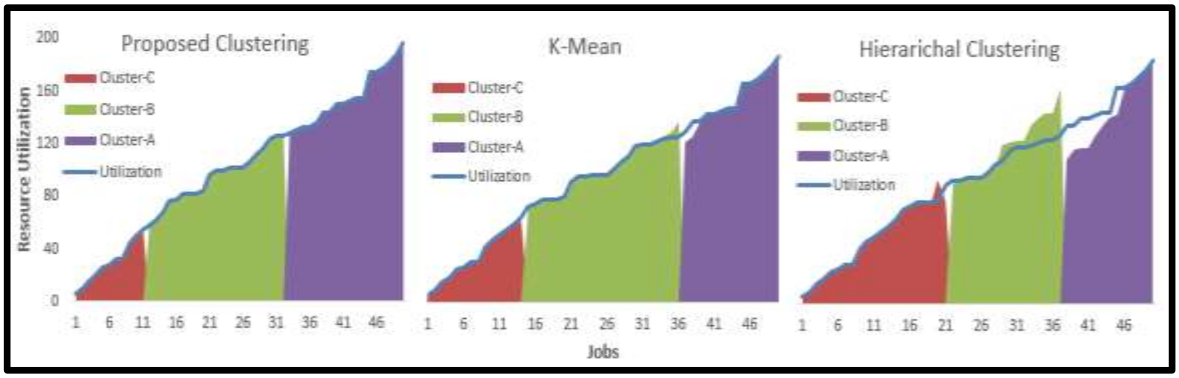

Figure 5.9 Cumulative resource analysis

Table 5.2 Color band Analysis

Decending Order	Task Number	Proposed Technique	K-Mean (K=3)	Hierarchical
196	50	50	50	50
188	9	9	9	9
182	3	3	3	3
178	47	47	47	47
174	18	18	18	18
174	38	38	38	38
154	10	10	10	10
154	25	25	25	25
152	15	15	15	15
150	7	7	7	7
150	48	48	48	48
144	26	26	26	26
144	41	41	41	41
136	17	17	17	17
132	4	4	4	4
132	42	42	42	42
130	28	28	28	28
128	2	2	2	2
126	23	23	22	22
126	22	22	23	23
124	30	30	30	30
116	12	12	12	12
112	40	40	40	40
106	6	6	6	6
102	31	31	31	31
102	36	36	36	36
102	39	39	39	39
100	1	1	1	1
100	11	11	11	11
96	21	21	21	21
84	5	5	5	5
82	8	8	8	8
82	27	27	27	27
82	49	49	49	49
78	34	34	34	34
76	19	19	19	19
68	29	29	29	29
62	37	37	37	37
58	33	33	33	33
54	32	32	32	32
50	16	16	16	16
44	20	20	20	20
32	14	14	14	14
32	24	24	24	24
28	35	35	35	35
26	46	46	46	46
20	43	43	43	43
16	13	13	13	13
10	45	45	45	45
6	44	44	44	44

The cumulative utilization band analysis and color band directly depicts wrong allocation using the increasing resource utilization curve. By analyzing the strangled area under the curve, we can observe that the elements having low resource utilization are allocated in the cluster that contains the elements with high resource utilization and vice versa.

5.2 Experiment 2

As demonstrated in chapter 4, how the proposed Energy-Aware scheduling works using the two-way cut-off mechanism for the resource utilization and tries to counter the VMs under the specified resource utilization. The scheduling algorithm also works as the best-fit algorithm for mapping the tasks onto the VMs in-order to create energy aware green CC environment. To depict the benefits and working of our proposed scheduling approach, we tried to map total 30 tasks having distinct resource utilization on the presently running 50 VMs and compare the proposed scheduling technique with the prior existing FCFS (First Come First Serve) scheduling algorithm. The results and comparison depict the difference between the two phases:

- **Phase 1** Proposed clustering of VMs according to resource utilization.
- **Phase 2** Mapping tasks onto VMs using the Energy-Aware scheduling technique.

Phase-1

The below graph represents the resource utilization of the distinct VMs at time T_0. After analyzing the present resource utilization of the VMs, we clustered the VMs using the proposed clustering technique.

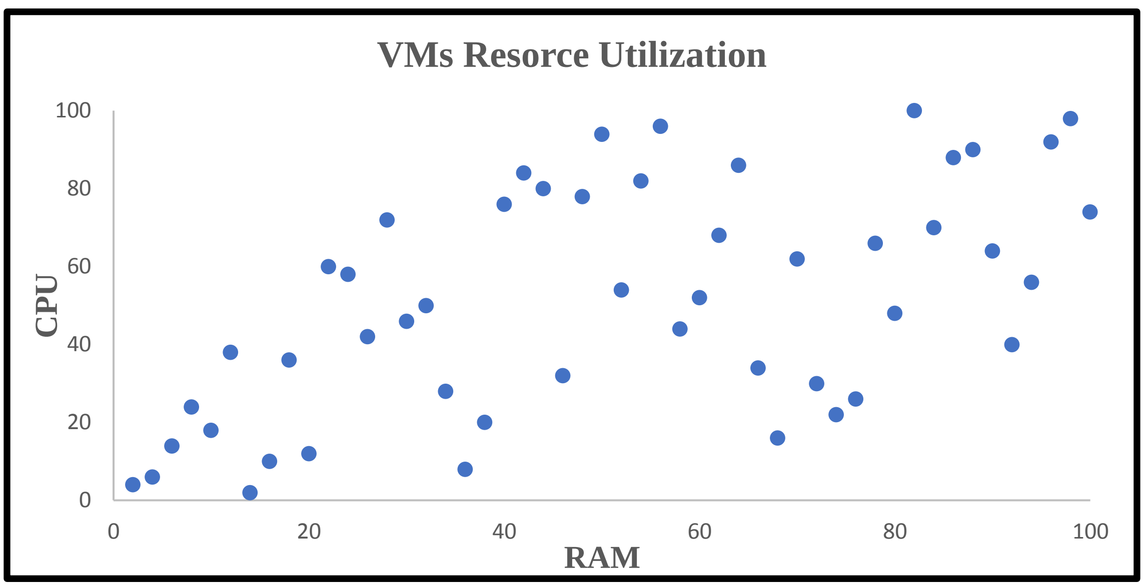

Figure 5.10 VM Resource Utilization

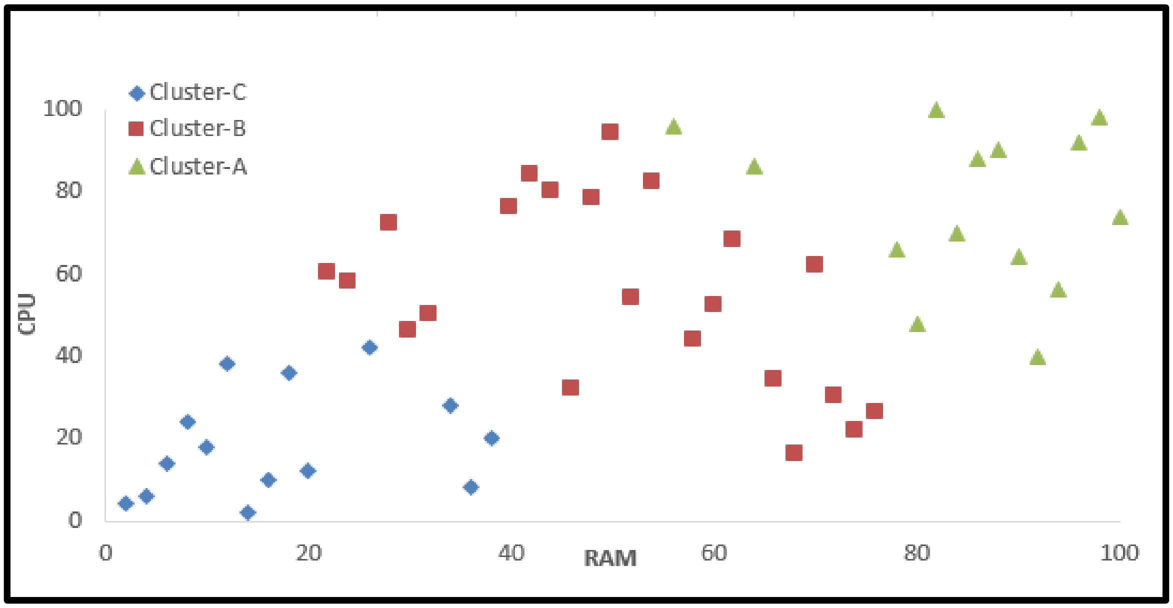

Figure 5.11 Proposed Clustering of the VMs according to the resource utilization.

Phase 2

After the clustering process, the VMs are arranged in increasing order of their resource utilization (shown in the graph below) . The cutoffs that are finalized by depicting the utilization point at which the cluster changes (the two straight lines depicts cut-off). These cut-offs can be user defined also. The proposed scheduling algorithm tries to counter the machine in the given cut-off when the allocation process initiates.

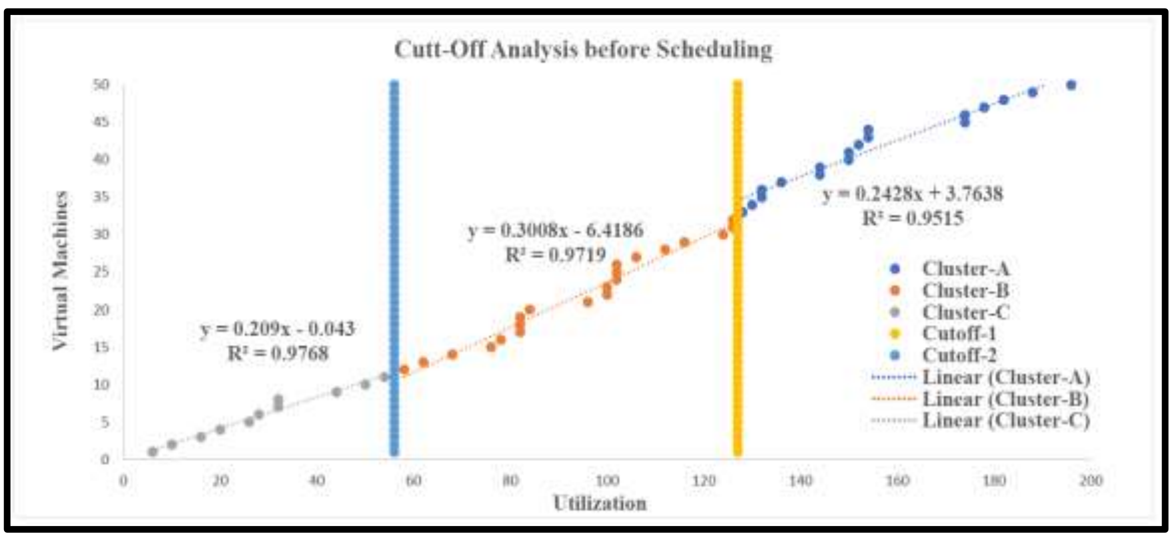

Figure 5.12 Cutoff analysis before Energy Aware Scheduling

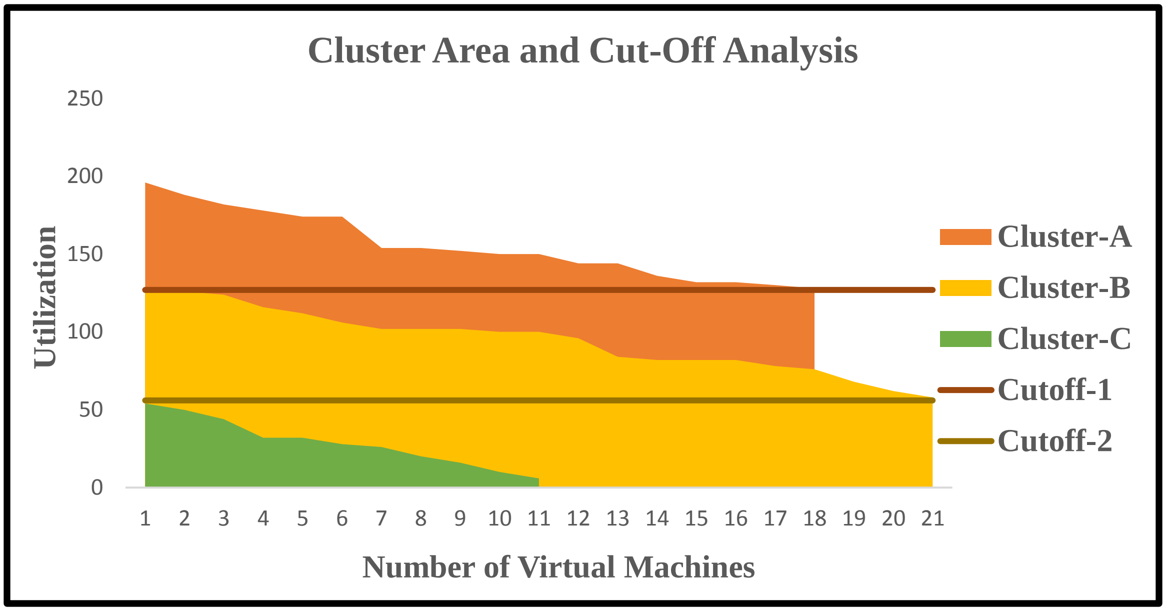

Figure 5.13 Cluster Area and Cut-off Analysis

The above graph demonstrates the utilization area under a particular cluster and the number of VMs present in the following cluster. The two lines indicates the selected cut-off.

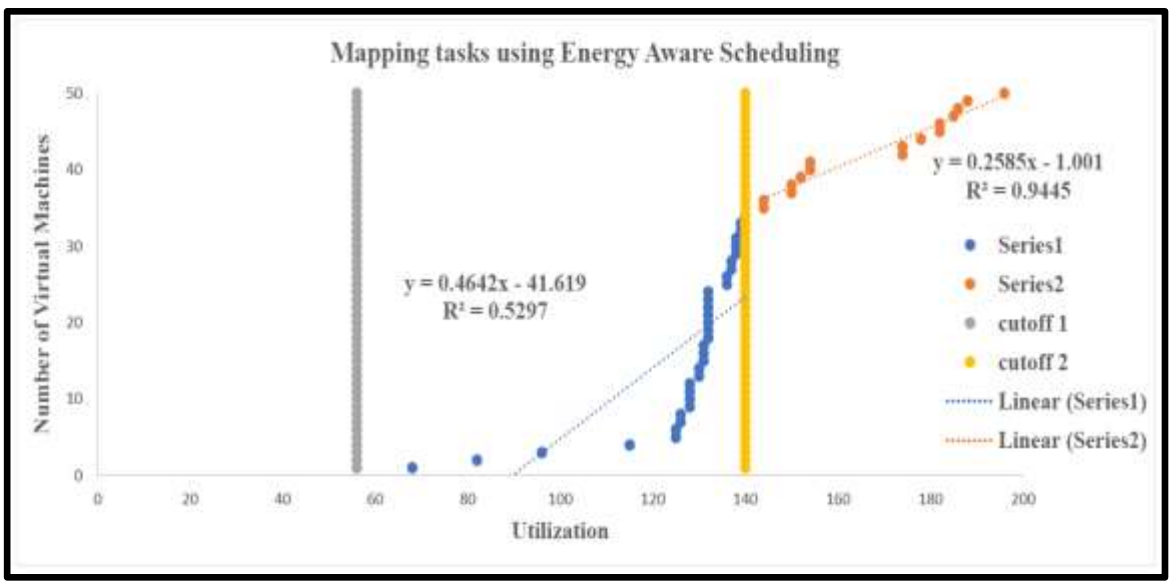

Figure 5.14 Mapping tasks using Energy Aware Scheduling

After completing all the initial steps, we executed the scheduling process by mapping 30 odd tasks onto the VMs. The tasks are allocated by countering total utilization according to the desired cut-off. The results obtained are shown graphically.

5.2.1 Comparison

We have compared the proposed scheduling technique with the prior existing FCFS scheduling mechanism.

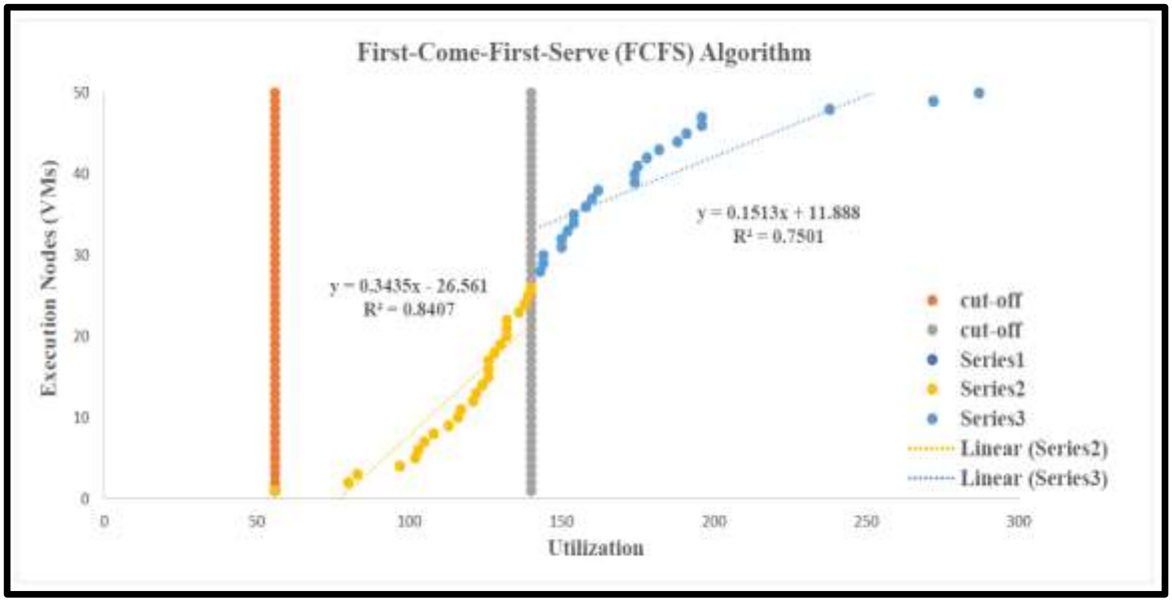

Figure 5.15 Regression Analysis First Come First Serve

FCFS Regression analysis, to determine the utilization of the overall system when tasks are mapped onto the VMs.

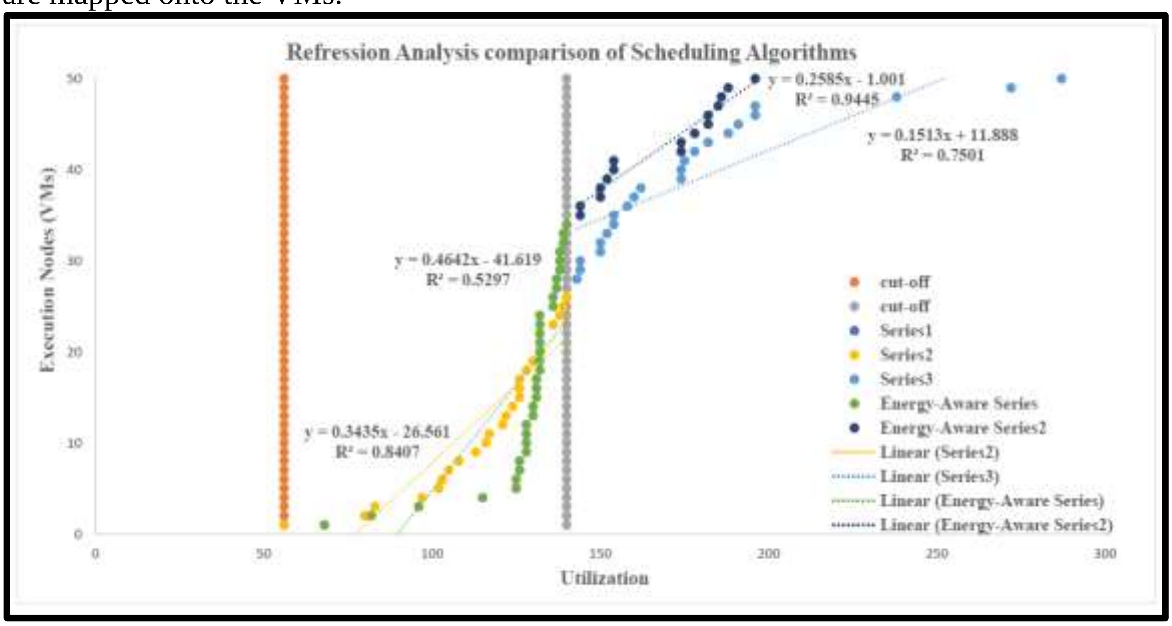

Figure 5.16 Energy-Aware and FCFS scheduling comparison

Comparison of the proposed scheduling technique and prior existing FCFS Scheduling technique. The number of elements within the threshold and the average utilization can be easily observed.

5.3 Research Contributions

1. The new clustering technique is proposed to generates the clusters of unequal length with non-spherical structure
2. An automatic detection of the number of clusters is used in the proposed scheme which reduces the boundary problem for the element to be allocated in a specific cluster.
3. Keeping focus on the value of the elements, rather than distance between them, a single iteration algorithm is designed in the proposed scheme.
4. The scheduling user more than one user defined thresholds to create an upper limit and lower limit for the resource utilization to achieve better task allocation.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In the coming future of technology, Expansion of CC leads its ability to emerge as a dominant architecture in the field of computational landscape. The dedicated infrastructure that offers cloud services to the users are elevating both in number and computation to meet the growing demands. Obviously, this elevation leads to various challenges, including that of energy consumption, green computing, effective use of resources and many more.

In this thesis, we try to mitigate some challenges, by concentrating on the optimization of the energy consumption in different cloud architecture. Unlike the prior existing methods that handles the related challenges separately, we in our research we have tried to enlarge the perspective and vision, by jointly considering the management and scheduling problems in single frameworks. In Chapter-4 we proposed an Energy Aware model for the CC environment that elaborates the proposed clustering technique accompanied by the Energy-Aware scheduling algorithm. The clustering techniques clusters the VMs on the basis of their etiquette, whereas the scheduling algorithm tries to allocate the tasks to VMs depending on the cut-off interval selected by the use or administrator. The beneficial results obtained using the proposed model can be listed as:

1. The proposed clustering approach does not follow the centroid fashion for clustering the VMs together rather it uses the p-value i.e. the level the significance to find the significant difference between two VMs to classify them into different clusters.
2. It overcomes the imitations of other centroid based clustering techniques which require predetermination of the number of clusters and initial centroids and face compatibility issues regarding elongated data sets.
3. The proposed approach is easy to apply and applicability is also illustrated with the help of an experimental data set.
4. The Energy-Aware scheduling technique uses multiple threshold rather than using just single threshold to counter the energy utilization.

5. It acts as the best-fit algorithm, so as to determine the best task for ten VM to achieve consumption under the adjusted cut-off.
6. Helps to halt or standby the VMs with very less resource utilization.

6.2 Future Scope

In this chapter, the future directions and scope of the proposed clustering algorithm and scheduling algorithm are discussed.

1. Our future goal is to show the applicability of the proposed scheduling scheme for which we wish to apply the proposed scheme to a real DC and analyze the results so that the applicability and the authenticity of the proposed scheme can be shown.
2. In the proposed scheme, all the VMs are firstly categorized into different clusters on the basis of their resource utilization and afterwards, the tasks are mapped to different VMs residing in different clusters in such a fashion to lower the energy utilized. The scheduling of tasks on the clustered VMs is comparatively more adequate as it takes all the attributes into consideration. In the proposed scheme, energy use is the primary focus while scheduling the tasks on clustered VMs. However, no single scheduling technique can be fit in all the environments so our future goal is to propose a cost aware demand scheduling for the clustered VMs.
3. Our future direction is to use the Regression Discontinuity Design for the scheduling analysis of the tasks to the clustered VMs which will increase the efficiency of the proposed scheduling algorithm.

REFERENCES

[1] Garfinkel, Simson. Architects of the information society: 35 years of the Laboratory for Computer Science at MIT. MIT press, 1999.

[2] Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee *et al*. "The magazine archive includes every article published in Communications of the ACM for over the past 50 years." Communications of the ACM 53, no. 4 (2009): 50-58.

[3] Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Future Generation computer systems 25, no. 6 (2009): 599-616.

[4] Vaquero, Luis M., Luis Rodero-Merino, Juan Caceres, and Maik Lindner. "A break in the clouds: towards a cloud definition." ACM SIGCOMM Computer Communication Review 39, no. 1 (2008): 50-55.

[5] Zhang, Qi, Lu Cheng, and Raouf Boutaba. "Cloud computing: state-of-the-art and research challenges." Journal of Internet services and applications 1, no. 1 (2010): 7-18.

[6] Kushida, Kenji E., Jonathan Murray, and John Zysman. "Cloud computing: from scarcity to abundance." Journal of Industry, Competition and Trade 15, no. 1 (2015): 5-19.

[7] Zhou, Minqi, Rong Zhang, Dadan Zeng, and Weining Qian. "Services in the cloud computing era: A survey." In Universal Communication Symposium (IUCS), 2010 4th International, pp. 40-46. IEEE, 2010.

[8] Viega, John. "Cloud computing and the common man." Computer 42, no. 8 (2009): 106-108.

[9] Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee *et al*. "A view of cloud computing." Communications of the ACM 53, no. 4 (2010): 50-58.

[10] Obaidat, Mohammad S., and Petros Nicopolitidis. Smart cities and homes: Key enabling technologies. Morgan Kaufmann, 2016.

[11] Tsai, Wei-Tek, Xin Sun, and Janaka Balasooriya. "Service-oriented cloud computing architecture." In Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, pp. 684-689. IEEE, 2010.

[12] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." Communications of the ACM 53, no. 6 (2010): 50.

[13] https://aws.amazon.com/it/ec2/.

[14] https://appengine.google.com/.

[15] https://aws.amazon.com/.

[16] https://www.google.com/gmail/.

[17] https://it-it.facebook.com/.

[18] Barham, Paul, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. "Xen and the art of virtualization." In ACM SIGOPS operating systems review, vol. 37, no. 5, pp. 164-177. ACM, 2003.

[19] Smith, James E., and Ravi Nair. "The architecture of virtual machines." Computer 38, no. 5 (2005): 32-38.

[20] https://www.docker.com/.

[21] Rokach, Lior, and Oded Maimon. "Clustering methods." In Data mining and knowledge discovery handbook, pp. 321-352. Springer, Boston, MA, 2005.

[22] Henzinger, Thomas A., Anmol V. Singh, Vasu Singh, Thomas Wies, and Damien Zufferey. "Static scheduling in clouds." (2011): 1-6.

[23] Lee, Yun-Han, Seiven Leu, and Ruay-Shiung Chang. "Improving job scheduling algorithms in a grid environment." Future generation computer systems 27, no. 8 (2011): 991-998.

[24] Xhafa, Fatos, and Ajith Abraham. "Computational models and heuristic methods for Grid scheduling problems." Future generation computer systems 26, no. 4 (2010): 608-621.

[25] Types of Scheduling: http://orzhovgilden.site11.com/projects/CPU.html.

[26]Types of Scheduling Algorithms
http://webdocs.cs.uerta.ca/~tony/C379/Notes/PDF/05.4.pdf

[27] Canali, Claudia, and Riccardo Lancellotti. "Automated clustering of virtual machines based on correlation of resource usage." (2012): 102-109.

[28] Canali, Claudia, and Riccardo Lancellotti. "Improving scalability of cloud monitoring through PCA-based clustering of virtual machines." Journal of Computer Science and Technology 29, no. 1 (2014): 38-52.

[29] Canali, Claudia, and Riccardo Lancellotti. "Exploiting classes of virtual machines for scalable iaas cloud management." In Network Cloud Computing and Applications (NCCA), 2015 IEEE Fourth Symposium on, pp. 15-22. IEEE, 2015.

[30] Canali, Claudia, and Riccardo Lancellotti. "An adaptive technique to model virtual machine behavior for scalable cloud monitoring." In Computers and Communication (ISCC), 2014 IEEE Symposium on, pp. 1-7. IEEE, 2014.

[31] Canali, Claudia, and Riccardo Lancellotti. "Exploiting ensemble techniques for automatic virtual machine clustering in cloud systems." Automated Software Engineering 21, no. 3 (2014): 319-344.

[32] Canali, Claudia, and Riccardo Lancellotti. "AGATE: Adaptive Gray Area-based TEchnique to Cluster Virtual Machines with Similar Behavior." IEEE Transactions on Cloud Computing(2017).

[33] Lefurgy, Charles, Xiaorui Wang, and Malcolm Ware. "Server-level power control." In Autonomic Computing, 2007. ICAC'07. Fourth International Conference on, pp. 4-4. IEEE, 2007.

[34] Felter, Wes, Karthick Rajamani, Tom Keller, and Cosmin Rusu. "A performance-conserving approach for reducing peak power consumption in server systems." In Proceedings of the 19th annual international conference on Supercomputing, pp. 293-302. ACM, 2005.

[35] Fan, Xiaobo, Wolf-Dietrich Weber, and Luiz Andre Barroso. "Power provisioning for a warehouse-sized computer." In ACM SIGARCH Computer Architecture News, vol. 35, no. 2, pp. 13-23. ACM, 2007.

[36] Govindan, Sriram, Jeonghwan Choi, Bhuvan Urgaonkar, Anand Sivasubramaniam, and Andrea Baldini. "Statistical profiling-based techniques for effective power provisioning in data centers." In Proceedings of the 4th ACM European conference on Computer systems, pp. 317-330. ACM, 2009.

[37] Choi, Jeonghwan, Sriram Govindan, Bhuvan Urgaonkar, and Anand Sivasubramaniam. "Profiling, prediction, and capping of power consumption in

consolidated environments." In Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on, pp. 1-10. IEEE, 2008.

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

[39] Zeng, Heng, Carla S. Ellis, Alvin R. Lebeck, and Amin Vahdat. "ECOSystem: Managing energy as a first class operating system resource." ACM SIGOPS operating systems review36, no. 5 (2002): 123-132.

[40] Kansal, Aman, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A. Bhattacharya. "Virtual machine power metering and provisioning." In Proceedings of the 1st ACM symposium on Cloud computing, pp. 39-50. ACM, 2010.

[41] Snowdon, David C., Etienne Le Sueur, Stefan M. Petters, and Gernot Heiser. "Koala: A platform for OS-level power management." In Proceedings of the 4th ACM European conference on Computer systems, pp. 289-302. ACM, 2009.

[42] Tiwari, Vivek, Sharad Malik, Andrew Wolfe, and Mike Tien-Chien Lee. "Instruction level power analysis and optimization of software." In Technologies for wireless computing, pp. 139-154. Springer, Boston, MA, 1996.

[43] Sinha, Amit, and Anantha P. Chandrakasan. "JouleTrack: a web based tool for software energy profiling." In Proceedings of the 38th annual Design Automation Conference, pp. 220-225. ACM, 2001.

[44] Wang, Lizhe, Samee U. Khan, Dan Chen, Joanna KołOdziej, Rajiv Ranjan, Cheng-Zhong Xu, and Albert Zomaya. "Energy-aware parallel task scheduling in a cluster." Future Generation Computer Systems 29, no. 7 (2013): 1661-1670.

[45] Hosseinimotlagh, Seyedmehdi, Farshad Khunjush, and Seyedmahyar Hosseinimotlagh. "A cooperative two-tier energy-aware scheduling for real-time tasks in computing clouds." In Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on, pp. 178-182. IEEE, 2014.

[46] Wu, Kehe, Ruo Du, Long Chen, and Su Yan. "An Energy-Saving Virtual-Machine Scheduling Algorithm of Cloud Computing System." In Information Science and Cloud

Computing Companion (ISCC-C), 2013 International Conference on, pp. 219-224. IEEE, 2013.

[47] Liu, Ning, Ziqian Dong, and Roberto Rojas-Cessa. "Task scheduling and server provisioning for energy-efficient cloud-computing data centers." In Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on, pp. 226-231. IEEE, 2013.

[48] Hossain, Mohammad M., Jen-Cheng Huang, and Hsien-Hsin S. Lee. "Migration energy-aware workload consolidation in enterprise clouds." In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pp. 405-410. IEEE, 2012.

[49] Szymanski, Ted H. "Low latency energy efficient communications in global-scale cloud computing systems." In Proceedings of the 2013 workshop on Energy efficient high performance parallel and distributed computing, pp. 13-22. ACM, 2013.

[50] Cao, Fei, and Michelle M. Zhu. "Energy efficient workflow job scheduling for green cloud." In Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International, pp. 2218-2221. IEEE, 2013.

[51] Walpole, Ronald E., Raymond H. Myers, Sharon L. Myers, and Keying Ye. Probability and statistics for engineers and scientists. Vol. 5. New York: Macmillan, 1993.

APPENDIX A

PUBLICATION

[1] Vipul Moudgil, Gagangeet Singh Aujla, Ohammad S. Obaidat, Neeraj Kumar, Radu Prodan DLopC: Data Locality Independency-aware VM Clustering in Cloud Computing (Communicated).

APPENDEX B

VIDEO PRESENTATION LINK

https://www.youtube.com/channel/UCvLaz00Q3wOdGRdGXXjaJnA/featured

APPENDIX C

PLAGIARISM REPORT

ORIGINALITY REPORT

9%	7%	6%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Hertfordshire Student Paper	1%
2	Hancong Duan, Chao Chen, Geyong Min, Yu Wu. "Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems", Future Generation Computer Systems, 2017 Publication	<1%
3	en.wikipedia.org Internet Source	<1%
4	e-kadin-erkek-esitligi-icin-ne-yapmalisiyasihaber2.org Internet Source	<1%
5	www.tachowdhury.com Internet Source	<1%