

Efficient Virtualization Schema for Load Management in Cloud Data Centers

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

Master of Engineering
in
Information Security

Submitted By
Sakshi Jaiswal
(Roll No: 801633007)

Under the supervision of
Mr Abhinav Anand
Senior Manager
SanDisk India Pvt. Ltd.

Dr Neeraj Kumar
Associate Professor
TIET, Patiala

Dr. Raman Singh
Assistant Professor
TIET, Patiala



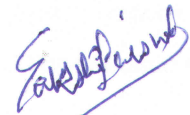
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY PATIALA,
PUNJAB, INDIA

June 2018

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*Efficient Virtualization schema for Load Management in Cloud Data Centers*", in partial fulfilment 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. This is my own work carried out at SanDisk India Pvt. Ltd. under the supervision of Senior Manager *Mr. Abhinav Anand* and at TIET, Patiala under the supervision of *Dr. Neeraj Kumar* and *Dr. Raman Singh*. The study 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.

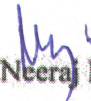


(Sakshi Jaiswal)

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



(Mr. Abhinav Anand)
Senior Manager, Software Tools
SanDisk India Pvt. Ltd.



(Dr. Neeraj Kumar)
Associate Professor, CSED



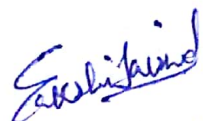
(Dr. Raman Singh) 8/8/18
Assistant Professor, CSED

Acknowledgement

First of all, I would like to thank the almighty who has always guided me to work on the right path of life. The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured success. I would like to express my deepest gratitude to **Mr. Abhinav Anand, Senior Manager, Software Tools Team, SanDisk India Device Design Center Pvt. Ltd.** for all his guidance and support. This work was empowered and maintained by his vision and thoughts. I would also like to express my gratitude to **Dr. Neeraj Kumar, Associate Professor and Dr. Raman Singh, Assistant Professor** who gave me the flexibility to explore on my own and at the same time the guidance to recover when my steps faltered. Their patience and support helped me overcome many crisis situations and this thesis.

I am grateful to **Dr. Maninder Singh, Professor and HOD, Computer Science and Engineering Department** and entire faculty and friends who devoted their valuable time towards successful completion of this thesis work.

Lastly, I would like to thank my family for their years of unyielding love and encouragement. They have always wanted the best for me and I admire their determination and sacrifice.


(Sakshi Jaiswal)

Cloud Computing provides a mechanism of delivering application as services and resources in data center that provides those services. Cloud computing gives a surrounding for sharing resources of different framework, application development platform, and middleware development platform and business application. The cloud services are flexible in terms of their usage i.e., pay-as-you-use. In this thesis, CPU performance of data center have been analysed on the basis of two datasets. Presently cloud computing framework supports/hosts millions of physical servers which produce lot of heat and consume more electricity. Thus, analyse the workload performance of cloud datacenter hosting business applications.

Cloud Infrastructure as Service (CIAS) is a collection of services, which is used to automate the process of test failure reproduction. Virtual machines are used to reproduce model based test failures to improve productivity while performing failure analysis. In this thesis, it is explained that how virtual box is enabled in Cloud infrastructure to make much easier for developers to perform debugging or testing in its local machine without any delay for longer period of time.

Table of Contents

CERTIFICATE.....	ii
ACKNOWLEDGEMENT.....	iii
ABSTRACT.....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES.....	vii
LIST OF TABLES.....	viii
Chapter 1: Introduction.....	1
1.1 Cloud Computing.....	1
1.1.1 Characteristics of Cloud Computing.....	2
1.1.2 Service Models of Cloud Computing.....	3
1.1.3 Deployment of Cloud Computing.....	3
1.2 Datacenter.....	4
1.3 Virtualization.....	5
1.3.1 Hypervisor based platforms.....	5
1.3.2 OS-based platforms.....	6
1.4 Thesis Outline.....	7
Chapter 2: Literature Review.....	8
Chapter 3: Problem Statement.....	10
3.1 Problem Analysis.....	10
3.2 Objectives.....	11
Chapter 4: Cloud Infrastructure as a Service.....	12
4.1 CIAS Overview.....	12
4.1.1 Basic Overview.....	12
4.1.2 CIAS architecture and its components.....	14
4.1.3 Basic Work Flow of CIAS.....	16
4.2 Enabling Virtual Box to CIAS.....	17
4.2.1 Background study.....	17
4.2.2 Virtual Box Scenario.....	18
4.2.3 Technology Used.....	18

4.2.4 Objective.....	18
4.2.5 Brief Idea.....	18
4.2.6 Tools for setting up the environment.....	19
4.2.7 Working of Virtual Box API in CIAS.....	19
Chapter 5: Experimental Setup and Result Analysis.....	26
5.1 Dataset Description.....	26
5.2 Dataset Parameters.....	26
5.3 Flow of Research Methodology.....	27
5.4 Experimental Results.....	28
Chapter 6: Conclusion.....	42
6.1 Conclusion.....	42
References.....	43
Plagiarism Report.....	45

List of Figures

Figure 1.2: Classification of Virtualization platforms.....	5
Figure 4.2: CIAS Architecture	14
Figure 4.3: Screenshot of Virtual Box when Service is started.....	20
Figure 4.4: Screenshot of VM import to Virtual Box	21
Figure 4.5: Screenshot of getting List of VMs	23
Figure 4.6: Screenshot of VM Clone	24
Figure 5.1: Datasets.....	26
Figure 5.2: Research Methodology Flow.....	27
Figure 5.3: CPU capacity usage analysis for machines in FastStorage.....	28
Figure 5.4: Memory capacity usage analysis for machines in FastStorage.....	30
Figure 5.5: CPU capacity usage analysis for machines in RndMonth7.....	32
Figure 5.6: Memory capacity usage analysis for machines in RndMonth7.....	33
Figure 5.7: CPU capacity usage analysis for machines in RndMonth8.....	35
Figure 5.8: Memory capacity usage analysis for machines in RndMonth8.....	36
Figure 5.9: CPU capacity usage analysis for machines in RndMonth9.....	38
Figure 5.10: Memory capacity usage analysis for machines in RndMonth9.....	39

List of Tables

Table 5.1: Performance analysis with different datasets.....	41
--------------------------------------------------------------	----

Chapter 1

Introduction

1.1 Cloud Computing

Cloud computing gives a surrounding for sharing resources of different frameworks, application development platform, middleware development platform and applications. Models handle infrastructure services with platform framework, infrastructure framework with additional application services and much more [1]. This has been outlined in courses by expert enterprises, scholastics, professionals and companies. It is larger than usual pool of just usable and open virtual resources. These resources are reconfigured dynamically that regulates variable load (scaling), additionally allow for optimized resource utilization [2].

Amazon, Google, Yahoo and various web service providers Microsoft, Nokia, Samsung and many other companies have suggested their own distributed computing strategy. Cloud computing already has many advantages and in many business applications data is being migrating to public cloud [5].

According to NIST (National Institute of Standards and Technology) It is a model which enables convenient, on-request network accessibility to specific group of configured computing resources (servers, systems, networks and various other services) which can be quickly provisioned and with very less effort or interaction with service provider. It is basically based on some characteristics, cloud service models and deployment models [3].

Distributed computing experiences a loophole. As there are millions of individual accessing the cloud daily, there is massive activity in the servers that provide various services. Sometimes there is an extreme congestion in the network. For instance, while buffering recordings on YouTube there is delay in buffering despite the fact that there is a strong web connection or while booking a show ticket, the site recommends that there are too many requests at that specific minute. These are some of the practical examples of server failures, which demonstrate that since such huge numbers of requests are being sent at once, the server can't process all the data and hence, it crashes. This is an extreme issue and can make serious harm to the association that owns the cloud service. To design a

technique that decreases this overbearing burden on the servers the idea of load balancing is used. Load balancers are used to minimize the burden on the servers. Thus, a load balancer's role is to distribute the incoming traffic in a bunch of computing units with a specific end goal to keep up a smooth flow of data.

When there are thousands/millions of requests every second and just a couple of servers can keep an eye on them, load balancing system becomes a necessity for smooth execution of the system. Load balancing is implemented in both software and hardware side. In hardware load balancers, there are dedicated systems on chip gadgets, which are physically integrated in the cloud [13]. Hence it is not always feasible, since it turns out to be immensely costly on a large scale. Consequently, the software load balancing systems are implemented, which are calculations or algorithms coded into the framework. There are essentially two types of software load balancers: static and dynamic system. The software options are extremely cost efficient to execute.

1.1.1 Characteristics of Cloud Computing

Cloud computing can make user get service at anyplace anytime. So, basically it has several characteristics.

- On-demand Resource Provisioning - It empowers the customer to viably get to access services from the cloud more effectively without any human intervention.
- Resource Pooling - Multi-tenant model is used to serve multiple consumers based on computing resources with various virtual resources which are assigned according to the demand of consumer on dynamic basis. Resources contain processing, virtual and physical machines, network bandwidth, storage and memory.
- Rapid Elasticity - Services are provisioned rapidly and elastically.
- Measured Service – Frameworks that consequently control and minimize the resources that to be use by utilizing a metering capability (e.g., bandwidth, processing).

1.1.2 Service Models of Cloud Computing

Cloud computing service models are:

- Cloud Software as a Service (SaaS) – Cloud consumers which can utilize their applications on host environment that is accessible using internet from application clients. The most renowned vendors are Salesforce, Google Mail and Google Docs.
- Cloud Infrastructure as a Service (IaaS) – Ability that has been provided to the client so that it can install and execute the applications as per its need. Clients can straight forwardly utilize IT frameworks (storage, processing, networking and many other resources). It provides “Virtualization” so as to integrate resources in an impromptu way to meet developing those resources from cloud. Infrastructure cloud provider is Amazon.
- Cloud Platform as a Service (PaaS) – It provides platform that supports full “Lifecycle of software applications” and permits cloud clients to develop application and services. The fundamental distinction amongst SaaS and PaaS is that SaaS only hosts fully complete applications of cloud while PaaS is more advanced platform which hosts both finished and in-process cloud applications. Google AppEngine is a PaaS provider.

1.1.3 Deployment of Cloud Computing

Depending on cloud consumer requirement and availability Cloud computing can be characterize as four different models.

- Public Cloud – It refers to large infrastructure and high performance that has been operated by external company’s. Clients can access to the service victimization external interface offered by internet browsers. They share a standard infrastructure of cloud and don’t seem to intimate it. Public clouds are less secure but worthwhile in costs. Association that cannot manage the cost of huge IT investments and also the organization which do not have a lot of private data then public cloud is mainly preferred [4].
- Private Cloud – It mainly available to a particular organization, less demanding to oversee security, support and updates continuously gives extra management over the preparation and utilization. Basically refers to cloud infrastructure that owned

and managed by a single company, also used in a single network and not available for public use.

- Community Cloud – It refers to sharing of cloud infrastructures for specific networks by various clients. It is secured and managed by all the participating organization or by third party service provider.
- Hybrid Cloud – It is a combination of both private and public cloud. Some service run on company's private cloud while some services on external public cloud based on specific requirements.

1.2 Datacenter

Datacenter is a facility in which large number of server, networking equipment and storage units together with all non-computing hardware such as cooling plant and continuous power supplies are assembled as a result of basic environmental requirements, physical security needs and for the simplicity of support [7]. Datacenter vary in size from server rooms that contains small- to medium-sized organizations to server farms that run large scale cloud services.

While the number of conventional datacenter that host applications on a dedicated hardware decreases while the number of modern cloud datacenters that uses virtualization for effective management of applications is widely increasing. The Cisco Global Cloud index [6] states that 75% of every computational workload was kept running in cloud datacenters in 2015 and figures out that this rate will increase to 92% in the year 2020. In this thesis, we focus on cloud datacenter and will use the terms datacenter and cloud datacenter conversely from now on.

The economies of scale accessible – for example, decrease in the cost of power, software, and hardware – make the development of extensive datacenter financially attractive. The development of large-scale datacenter is likewise determined by the increased capabilities of cloud services, quick development of cloud computing as accepted worldview by which online services are given and additional amount of data being generated by modern applications. These datacenters as a rule have hundreds and thousands of servers that consumes huge amount of power. Thus energy consumption has also become an important concern because of its impact on operating expenses, capital costs and environmental sustainability.

1.3 Virtualization

Given the variety of working framework (OS) and their numerous versions, including the frequently particular configuration required to suit the extensive variety of famous applications, it has turn out to be progressively hard to set up and oversee modern computer systems. Because of these difficulties and few other important benefits that it offers, virtualization turned into a basic innovation for managing computer systems. Virtualization empowers a single computer to host multiple virtual machines (making use of different OS stacks) over a local OS. The advantages of virtualization incorporate enhanced resource utilization, dynamic resource allocation, and increased flexibility.

Virtualization has a long history, returning to work in the early 1960s by organizations like General Electric [8], Bell Labs [9], and IBM [11]. Virtualization innovation has developed since those early time-sharing frameworks and centralized computers to support virtualization. However, it was not until the mid-2000s that the innovation ended up across as x86 based frameworks started to support key virtualization guidelines in hardware [10]. An overview of virtualized platform that are common in production environment are stated as hypervisor (H)-based and operating system (OS)-based virtualization methods.

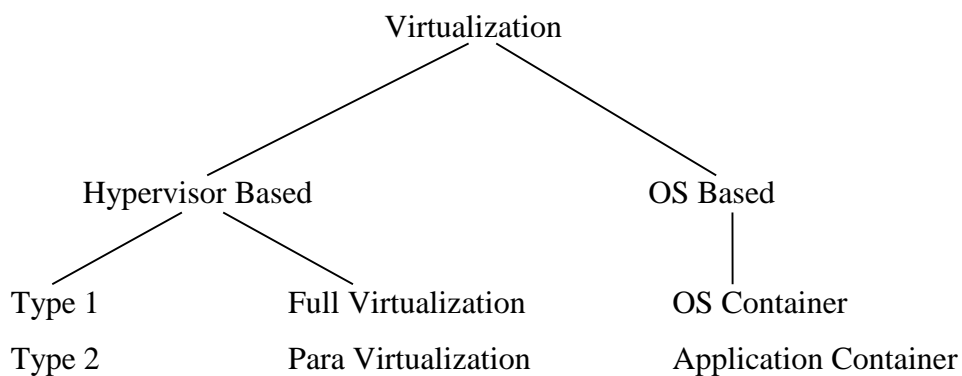


Figure 1.2. Classification of Virtualized platforms

1.3.1 Hypervisor-based platforms: H-based systems are the conventional virtualization framework in which a hypervisor is used to deal with the fundamental physical hardware. The hypervisor is in charge for making and managing with the execution of virtualized OS instances known as virtual machines (VMs) and assigning resources to the VMs. Hypervisor detach the VMs from the host framework and make possible to host VMs

running in different guest OS. H-based systems are generally slower than other virtualization framework because of the overhead brought up by the presence of a hypervisor and different guest OS kernels. These systems are basically classified as:

- **Type-1 hypervisors (native)** – Hypervisor that operate directly on system hardware. It provides high performance, availability and security because it operates as a layer designed to expose hardware resources to VMs, which reduces the complexity required to run the hypervisor itself. (e.g., XEN, Hyper-V)
- **Type-2 hypervisors (hosted)** – It operates on the top of host OS. It can be used mainly on systems where efficiency is low or where support for large number of Input output devices are important and can be provided by host system. (e.g., Oracle Virtual Box, VMware Workstation)
- **Full Virtualization** – It totally abstracts the guest System from the underlying hardware. The OS running over the virtualized equipment not changed at all.
- **Para-Virtualization** - It exhibits an interface to the guest OS that varies from the basic hardware. Accordingly, the kernel of the guest OS must be adjusted so as to work with the interface given by the hypervisor. Hardware virtualization uses virtualization highlights incorporated with the CPU of the host. The most recent ages of CPUs from Intel and AMD – Intel VT and AMD-V advances, respectively – offer support for hardware virtualization. This kind of virtualization gives the hardware extension required to run unmodified guest VMs and takes into account a good increment in VM execution [12].

1.3.2 OS based platforms: It is known as container based framework and don't use hypervisors. However, virtualizes resources at the OS level. Virtual instances created by encapsulating standard OS processes and their conditions. These instances are overseen by the basic OS kernel. More particularly, the Container Engine (CE) plays the same role as the hypervisor in an H-based system that manages image and containers while associating with the underlying OS kernel for resource allocation and management. OS-based virtualization acquires less overhead than H-based virtualization as because there is no intermediate layer of software (as the hypervisor) and a single OS is in run. Hence, sharing the host kernel too has weakness like decreased isolation among instances and inability to host different guest OS.

OS-based virtualization encapsulates occurrence in containers of which there are two types- system containers and application containers. Platforms which use system

containers can run numerous processes in a single container and are intended to provide an entire runtime environment but with a more lightweight outlined compared to H-based platforms. System containers (e.g., Linux containers) give a similar environment to H-based virtualization but somehow without the overhead of having a different host OS kernel. Interestingly, application containers (e.g., Docker containers) are created to run a single process and are more helpful for deploying distributed applications as in micro service architecture.

1.4 Thesis Outline

Chapter 1 - This chapter briefly discuss related to cloud computing, datacenter and virtualization.

Chapter 2 - This chapter provides related work done in cloud computing.

Chapter 3 - This chapter discusses about the problem statement.

Chapter 4 - This chapter discuss about the Cloud Infrastructure as a Service approach. Virtual box enabled in CIAS for providing the environment for developers for debugging/testing.

Chapter 5 – This chapter discuss about the tools that are used for setting up the environment and their results obtained.

Chapter 6 – This chapter discuss about the conclusion from the results obtained in fifth chapter.

Chapter 2

Literature Review

This chapter summarizes the research work proposed in this thesis.

Ulta *et al.* [14] proposed that due to factors like resource sharing, virtualization and congestion in cloud computing platform lead to deterioration and variability of the performance. If all different kinds of workload including applications like scientific, e-governance, business are collocated in the same data center especially under the presence of variable network bandwidth distribution then it has been found that due to CPU bound variability, the maximum slowdown is 1.48 and which is 1.79 for the network bound variability. All this leads to unpredictability of big data application performance to be deployed in cloud which actually were designed for the homogeneous system.

Kargatzis *et al.* [17] describe that while selecting a solution for a cloud platform we have to stick to a particular solution provider and also to stick which act as hindrance to adopt service from different providers. So, for migrating homogeneously and heterogeneously the author proposes technical parameters based on virtualization engine used by them and also presents the result by analysing HTTP response time in heterogeneous and homogeneous systems. Virtual machine migration tool presents migration between OpenStack and also VMware platform which has achieved 2 millisecond response time for the HTTP request.

Prasad *et al.* [15] proposed an algorithm called PLW Scheduling Algorithm which can be used to efficiently schedule resources to the jobs in datacenter having varying workload. This solves the challenge to schedule jobs in the presence of limited resources in the cloud environment. In this, by considering factors like priority, job length and waiting time dynamically, the weights are assigned to the jobs. In this, more weighted tasks are executed before the others having less weight. Resources are allocated to the more weighted tasks first by the cloud service providers.

Magalhaes *et al.* [20] proposed to capture behavioural pattern of different user profile. It was an extension of cloudsim simulator. User profile type is considered in this model. GWD and GEV distributions are used for instruction arrivals. Generalized extreme and lamda distributions are used for session time.

Chen *et al.* [18] proposed population-based incremental learning (PBIL) algorithm for resource scheduling on the cloud. Firstly, the author established cloud computing resource scheduling and PBIL as an objective solution. Then further analysis is performed through simulations on CloudSim. The results proved that PBIL is faster, required less time for task completion, resource load balancing has been achieved, large-scale task resource scheduling and many more advantages.

Elghoneimy *et al.* [16] proposed the solution of resource allocation and scheduling problem in cloud computing environment. The researcher discusses various algorithms for the same and suggest improvement in schedulers. Author also suggests which alternative can be used and has presented surveys for the same.

Gmach *et al.* [21] proposed usage for capacity management that considers workload demand patterns characteristics. Synthetic workload generation is considered in the analysis of efficient usage of resource pools with large number of enterprise servers. A method to automatically find the pattern differences is introduced by the author which can generate synthetic trace to represent future behaviour of workloads.

Chapter 3

Problem Statement

This chapter contains the problem statement of this research work.

3.1 Problem Analysis

Cloud Infrastructure as Service (CIAS) is a collection of services, which has been used to automate the process of test failure reproduction. All the necessary information will be provided by the client application. These services will perform the required operations on database and virtual machines. CIAS is a single click VM infrastructure Environment for running CIT, Reproduction, Regression tests, etc. and lot of other tests. Thus, fundamentally the Issue in the organization was to run large number of tests for longer period of duration without any failure or delay. Organization has constrained VMs and they have to give certain VMs to various groups to run those tests. The VMs have been given to various teams and they run different tests onto that. As each group has given some restriction like for just two weeks they have been given that particular VM. Following two weeks that VM will consequently turn out to be free and that resource will be assigned to new member who is in queue. In the meantime, while running various tests in a specific team, some test fails and the team needs that VM for more number of days then it becomes difficult to continue with the running test in that allotted VM due to Company's restriction. In this way, VM limitation comes out to be a problem in the organisation and if there is any failure while running or debugging the tests then company has no choice to continue running those failed test in the same VM of the fact that VM is restricted to just 2 weeks for a particular individual group to run the test. If environment is delayed then definitely release will be late and if product is late, then lot of loss to the company so it's a big issue.

Business workloads on application are progressively hosted in virtual datacenter as Infrastructure cloud. Seeing how workloads on business application request and utilize the resources. This work basically focuses on study of workloads facilitated in cloud data centers. Two datasets have been taken which contain requested and actual used resources in data center. In this thesis, our investigation reveals insight into the workload of cloud

data centers facilitating business-critical workloads. Workload analysis in cloud data center has been performed.

3.2 Objectives

The following are primary objectives of this work:

1. To develop a virtualization schema for running and debugging test cases using virtual box.
2. To preserve the running status of de-allocated incomplete/failed VMs.
3. To prepare the workload analysis of cloud data center.

Cloud Infrastructure as a Service Approach

4.1 CIAS Overview

Cloud Infrastructure as Service (CIAS) is a collection of services, which has been used to automate the process of test failure reproduction. All the necessary information will be provided by the client application. These services will perform the required operations on database and virtual machines.

4.1.1 Basic Overview

Provide environment for reproducing test failures - Virtual machines have been used to reproduce model based test failures and improve productivity when performing failure analysis.

- 1) Automation for failure analysis - Test failures have been reproduced automatically using tools in virtual machines.
- 2) Reserve VM - User can reserve virtual machines. Any number of machines can be reserved based on the user's requirement. Whenever user sends a request to reserve a machine, service will search for a free VM in the database and reserves it for the user.
- 3) Recompose VM - Whenever user sends request to free a virtual machine (VM), recompose VM operation has been performed. It will delete the VM and create a new VM with same name. The following actions have been performed when the VM is in recomposing state:
 - i. Power OFF: VM should be powered off.
 - ii. Delete VM: VM should be deleted.
 - iii. Clone VM: a new VM has been cloned automatically from the parent VM, with the VM name remaining the same.
 - iv. Power ON: The newly created VM will be powered ON.
- 4) When all the above operations have been executed in sequence, the user will get a free VM. Newly created VMs are available to the user, which thereby resolves the issue in delay of assigning VMs.

- 5) Create Snapshot - Snapshot feature enables the user to preserve the state and data of virtual machine at any time. It shortens time to debug long running test failures once snapshot has been created.
 - i. Create Snapshot: This feature enables the user to create a snapshot. Multiple snapshot of a virtual machine can be created.
 - ii. Revert to Snapshot: User can revert to the current state of a machine to a snapshot. The VM is reverted to the state it was in, when the snapshot has been created. In addition, the state can be reverted as many times based on user desire.
 - iii. Remove snapshot: User can remove the snapshot any time from the virtual machine.
- 6) No delay in assigning VMs - Whenever user sends request to free VM, It will delete the VM and create a new VM with same name (Recompose VM). Therefore, there will be no delay in assigning VMs. In addition, whenever user requests for new VM, user will get a completely clean VM, which may be used for a new test reproduction.
- 7) Facilitates user to debug multiple issues on multiple virtual machines - Multiple virtual machines have been used to debug different issues. It facilitates the user to switch between machines instantly by using different RDP sessions to debug. By recreating failures on machine, user can debug with ease.
- 8) Auto-free virtual machines - Supports auto free of machines, which are not being used for stipulated time duration and not released by the user. Unused machine can be used for a new reproduction once it is auto-free.
- 9) Managing information for all virtual machines - The user gets the updated information for all the machines available such as status of the machines, which are free and reserved. Also, provide details on IP Address, Station name, Owner for allocated machines and station description. To have better understanding Figure 4.2 displays the CIAS architecture.
- 10) Auto notification to user regarding the reproduction through mails - On allocation of free virtual machines to run the test, user gets mail notification:
 - i. On setting up the environment.
 - ii. Before the test starts.
 - iii. After the test is completed.

- 11) Restful APIs - User can call restful APIs to perform various virtual machine operations such as reserving a VM, Create Snapshot, Delete all Snapshots, Revert Snapshot, Delete Snapshot, Get All VMs, Get Snapshot List and others.
- 12) Operations will be asynchronous in nature, which means multiple users can perform multiple operations in parallel.

4.1.2 CIAS Architecture and its Components

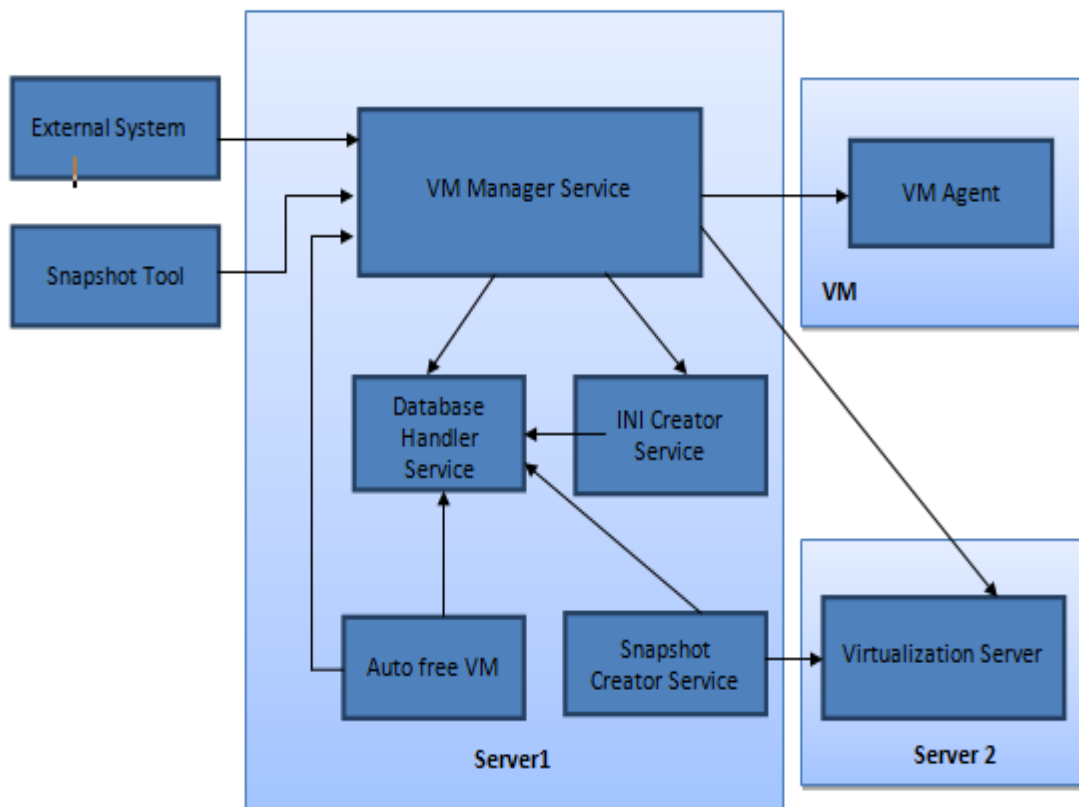


Figure 4.2. CIAS Architecture

CIAS includes the following components as shown in the Figure 4.2

1. Virtualization Server
2. VM Manager
3. VM Agent
4. Other Services

4.1.2.1 Virtualization Server:

Virtualization Server is a web service, which supports the following:

- a) It will manage multiple VM Operations supported by VMware like Delete VM, Create VM, Clone VM, Power On, Power Off, Create Snapshot, Revert Snapshot and Delete Snapshot etc.
- b) It will call V-sphere APIs for performing Operations.
- c) It can support one or more commands sequence by encapsulating in XML format.

4.1.2.2 VM Manager:

This service acts as an interface for Client application and other services. It is responsible for calling other services based on the request of the client. A client invokes the VM Manager by sending an xml message, and then waits for a corresponding xml response. It is a restful service and can take request from both web and windows application.

4.1.2.3 VM Agent:

The VM Agent has the following services to do with:

a) Windows Service:

- a. This service will enable auto login in the recomposed VM.
- b. The service will run as soon as the machine starts up (before login).
- c. User will never be asked to login to VM anymore.

b) VM Agent Service:

- a. This service should start-up as soon as user logs in.
- b. This service will provide all the functionalities for VM agent.
- c. Currently: Run a test, Update Registry, Update environment variable and run any command on VM.
- d. The service will register itself in database after startup.

4.1.2.4 Other Services:

- a) Database (DB) Handler Service - It has been used to handle the entire database for the cloud infrastructure. It typically contains the data records for managing various VM Centers, VM management pool, Free VMs, VMs reserved by the user.

- b) INI Creator Service - This service has been used to create Task.ini file. This file has been created, when the user reserves the VM. It creates the INI file with the VM name. Example: SWTest-1.ini
- c) Auto free VM Service - This service helps in releasing the unused VMs reserved by the user for a stipulated time.

4.1.3 Basic Work Flow of CIAS

4.1.3.1 Free VM Sequence:

- a) Client will send request to the VM Manager to free VM, as soon as the user frees the VM.
- b) VM manager will update the status to recompose in the database.
- c) VM Manager will call the **Recompose** of Virtualization Server. Recompose operation takes place. It will delete the VM and create a new VM with same name.
- d) VM Agent will update the status to free in the database, once the VM is up.

4.1.3.2 New VM request

- a) Client will send request to the VM Manager, whenever user request for a new VM.
- b) VM Manager will search for a free VM in the database, reserves it for the user and updates the status to 'Reserved' in the database.
- c) VM Manager calls the INI Creator Service and gets the INI file content.
- d) VM Manager Service will pass the INI file content to VM Agent, which is then responsible to start the test.

4.1.3.3 Create Snapshot

- a) Snapshot Creator Service will create snapshots.
- b) This service parses the incoming xml.
- c) This service gets the V-Center Name and credentials from DB Handler.
- d) This service will call **Create Snapshot** of virtualization Server.

4.1.3.4 Auto Free VM

- a) Auto free VM Service will poll for VMs, which are inactive for a stipulated time, on a regular basis in database.
- b) This Service will call **Free VM** of VM Manager for all the inactive VMs.
- c) Inactive VMs shall be freed automatically.

4.2 Enabling Virtual Box to CIAS

4.2.1 Background study

4.2.1.1 Virtual Box

Virtual Box is completely free and open source virtualized software which was developed by Sun Microsystems which is now a subsidiary of Oracle Incorporation. Virtual Box is a cross-platform virtualized application that can be introduced on PCs running Windows, Linux, Mac OS, and Solaris. Most present day PCs are powerful enough to run whole working OS inside main Operating system, which implies virtual machines are common place today than ever. It's a machine emulator and virtualizer. Oracle Virtual Box is definitely the best as compared to other virtualization softwares with features include rich, high performance, user friendly and many more. It's accessible for both 32-bit and 64-bit architectures.

4.2.1.2 API

An API is an application programming interface which is a medium of interaction between certain applications. Anything with a social login uses API, mobile App of a web based business site uses an API, even the promotions or ads that you see on the Internet uses APIs.

4.2.1.3 Restful APIs

A Restful API is an API that uses HTTP requests to PUT, POST, GET and Delete the data. Client will send request to a server element which is located at a URL and performs some operations with the methods mentioned. This provides a communication between the server and client where every request is independent of other request and contains information that requires to complete the request.

^The REST APIs support the following HTTP methods.

- **GET:** To retrieve the data.

- **POST:** To modify the data, perform a particular action

4.2.2 Virtual Box scenario

Limited number of Virtual machines in the organization gives an idea to develop new environment for the developers to do certain operations for testing using virtual box. The VM which is in VMware as VDI format, will be exported to one of the shared locations and using our tool that image will be imported to virtual box and continue testing the failed operation in the local machine.

4.2.3 Technology Used

ASP.NET C#, Microsoft SQL server, Oracle Virtual Box open source tool

4.2.4 Objective

Provide Virtual Box environment for running tests and debugging for the developers without affecting the delay. Also to maintain the continuation of in-completed or failed tests debugging in VM, When VM is de-allocated.

4.2.5 Brief Idea

The VM that has been allocated to the developers for their concerned tests/debugging will be exported in a shared location as OVF format (through provided API). After exporting has been done successfully, then import and perform various operations. Firstly, Installation of virtual box software has to be done to enable this virtual box with web server to remotely access the services of virtual box. Then through our API OVF template will be imported to the virtual box and start doing the operations. So, basically created the Virtual Box APIs:

- Export OVF (to shared location)
- Import OVF
- Get VM List
- Start VM
- Remove VM
- Take Snapshot VM
- Get All the Snapshots
- Clone VM
- Restore/Revert VM

- Stop VM

4.2.6 Tools for setting up the Environment

A brief introduction of tools which are used in designing and implementation of proposed framework is given below.

4.2.6.1 Microsoft Visual Studio 2012

It is an integrated development environment (IDE) from Microsoft and is used to create different programs. Additionally, it can develop sites, web services, mobile applications, and many more. It uses Microsoft software development platform, such as Windows Forms, Windows Presentation Foundation, Microsoft Silver light, Windows API, and Windows Store [13]. It can deliver both local and managed code.

4.2.6.2 Postman Client

It is an API development suite with great features that make the process of API development fast and effortless. The response that has achieved by our requests may likewise be seen in raw or somehow pretty forms. It already saves the previous API calls which saves our time.

4.2.7 Working of Virtual Box API in CIAS

In this section we present the implementation and performance analysis of the tool. It involves HTTP calls between virtual Box and APIs of VMware. It simply follows interactions between two platforms. There are many methods in which the Main API can be called by other code. Virtual Box accompanies a web service that maps almost the whole Main API. The service API is freely portrayed in a web service description file (in WSDL format), to write client programs that in any language can call the web service with a toolkit that understands WSDL. So, need to run these commands in command prompt to run the virtual box service. Virtual Box API is integrated with CIAS through Vbox WebServer. The service performs the authentication mechanism using the following command

```
VBoxManage setproperty webservauthlibrary vboxauthsimple
```

```
VBoxmanage internalcommands passwordhash "myPwd123"
```

```
VBoxmanage      setextradata      global      "VBoxAuthSimple/users/myUser"  
"2f751aa21cc2e20fad563d76c24ddfaf165de186d28d59ad42bf4f625921a115"
```

Vboxwebsrv.exe --host 0.0.0.0.

Figure 4.3 shows that service has been started and running. Now user can perform the operations according to the requirement.

```
C:\Users\User>VBoxManage setproperty webservauthlibrary vboxauthsimple
C:\Users\User>VBoxmanage internalcommands passwordhash "myPwd123"
Password hash: 2f751aa21cc2e20fad563d76c24ddfaf165de186d28d59ad42bf4f625921a115
C:\Users\User>VBoxmanage setextradata global "VBoxAuthSimple/users/myUser" "2f751aa21cc2e20fad563d76c24ddfaf165de186d28d59ad42bf4f625921a115"
C:\Users\User>Vboxwebsrv.exe --host 0.0.0.0
Oracle VM VirtualBox web service Version 5.2.10
(C) 2007-2018 Oracle Corporation
All rights reserved.
VirtualBox web service 5.2.10 r122406 win.amd64 (Apr 27 2018 10:39:07) release log
00:00:00.000497 main Log opened 2018-06-04T18:54:14.119371000Z
00:00:00.000497 main Build Type: release
00:00:00.000497 main OS Product: Windows 10
00:00:00.000497 main OS Release: 10.0.17134
00:00:00.000497 main OS Service Pack:
00:00:00.022953 main DMI Product Name: Lenovo IdeaPad U310 Touch
00:00:00.036929 main DMI Product Version: Lenovo IdeaPad U310 Touch
00:00:00.036929 main Host RAM: 3954MB (3.8GB) total, 1140MB available
00:00:00.036929 main Executable: C:\Program Files\Oracle\VirtualBox\VBoxWebSrv.exe
00:00:00.036929 main Process ID: 4176
00:00:00.036929 main Package type: WINDOWS_64BITS_GENERIC
00:00:00.069367 SQPmp Socket connection successful: host = 0.0.0.0, port = 18083, master socket = 896
00:05:05.071748 Watchdog Statistics: 0 webssessions, 0 references
00:05:23.758608 SQPmp Request 1 on socket 932 queued for processing (1 items on Q)
00:05:23.761681 SQW01 New SOAP thread started
00:05:23.763180 SQW01 Processing connection from IP=127.0.0.1 socket=932 (0 out of 1 threads idle)
00:05:23.781216 SQW01 External authentication library is 'vboxauthsimple'
00:05:23.787630 SQW01 Access for user 'myUser' granted
00:05:23.869757 SQPmp Request 2 on socket 712 queued for processing (1 items on Q)
00:05:23.872369 SQW01 Processing connection from IP=127.0.0.1 socket=712 (0 out of 1 threads idle)
00:05:23.892330 SQPmp Request 3 on socket 712 queued for processing (1 items on Q)
00:05:23.895061 SQW01 Processing connection from IP=127.0.0.1 socket=712 (0 out of 1 threads idle)
00:10:00.202804 Watchdog Statistics: 1 webssessions, 2 references
```

Figure 4.3. Virtual Box service has been started

After authentication has been done then developer can perform certain operations. We will do it by using POSTMAN client

4.2.7.1 Export Image

Firstly, user has to export the image to some shared location so that the image can be imported to the local machine using our provided API.

4.2.7.2 Import OVF

This API will import the image to the Virtual Box so that user can directly use the image and continue the failed operation accordingly. Figure 4.4 shows the screenshot of successfully import of OVF (CFJenkins) template to the virtual box.

URL: <http://localhost:52292/Service1.svc/ImportVMs?ip=localhost&username=sakshi&pwd=sakshi08&OVFPath=C:/WD/CFJenkins/CFJenkins.OVF>

Action Performed: Post

Response Type: Json

Request Type: Json

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:62443/Service1.svc/ImportVMs?ip=localhost&username=sakshi&pwd=sakshi08&OVFPath=C:/WD/CFjenkins/CFjenkins.OVF
- Parameters:**

Key	Value
<input checked="" type="checkbox"/> ip	localhost
<input checked="" type="checkbox"/> username	sakshi
<input checked="" type="checkbox"/> pwd	sakshi08
<input checked="" type="checkbox"/> OVFPath	C:/WD/CFjenkins/CFjenkins.OVF
New key	Value
- Authorization:** TYPE: Inherit auth from parent. Note: This request is not inheriting any authorization.
- Body:** Pretty, Raw, Preview, JSON (selected), and a refresh icon.
- Body Content:** A single line of JSON: `1 {}`

Figure 4.4. VM has been successfully imported

4.2.7.3 Start VM

This API will start a VM, user should provide the VMName which he/she wants to start.

URL: http://localhost:52292/Service1.svc/StartVM?ip=localhost&name=ubuntu

Action Performed: Post

Response Type: Json

Request Type: Json

Response Body:

This API returns the below response if the request is successful.

```
{
  "StartVMResult {
    "StatusCode": 0,
    "StatusMessage": "METHOD SUCCEEDED"
  }
}
```

4.2.7.4 Remove VM

This API removes a snapshot from a virtual machine. It also allows the user to remove multiple snapshots at a time from a VM.

URL:http://localhost:52292/Service1.svc/DeleteSnapshot?ip=localhost&vm=ubuntu&ss
name=newSnapshotUbuntuu

Action Performed: Delete

Response Type: Json

Request Type: Json

Response Body:

This API returns the below response if the request is successful.

```
{  
  "DeleteSnapshotResult": {  
    "StatusCode": 0,  
    "StatusMessage": "METHOD SUCCEEDED"  
  }  
}
```

4.2.7.5 Get all VM List

This API will give the list of all the VMs present in the Virtual Box. Figure 4.5 displays the screenshot of successfully getting the list of all the VMs present in the virtual box.

URL:http://localhost:52292/Service1.svc/GetVMs?ip=localhost

Action Performed: Get

Response Type: Json

Request Type: Json

Response Body:

This API returns the below response if the request is successful.

```
{  
  "GetVMsResult {  
    "StatusCode": 0,  
    "StatusMessage": "METHOD SUCCEEDED"  
  }  
}
```

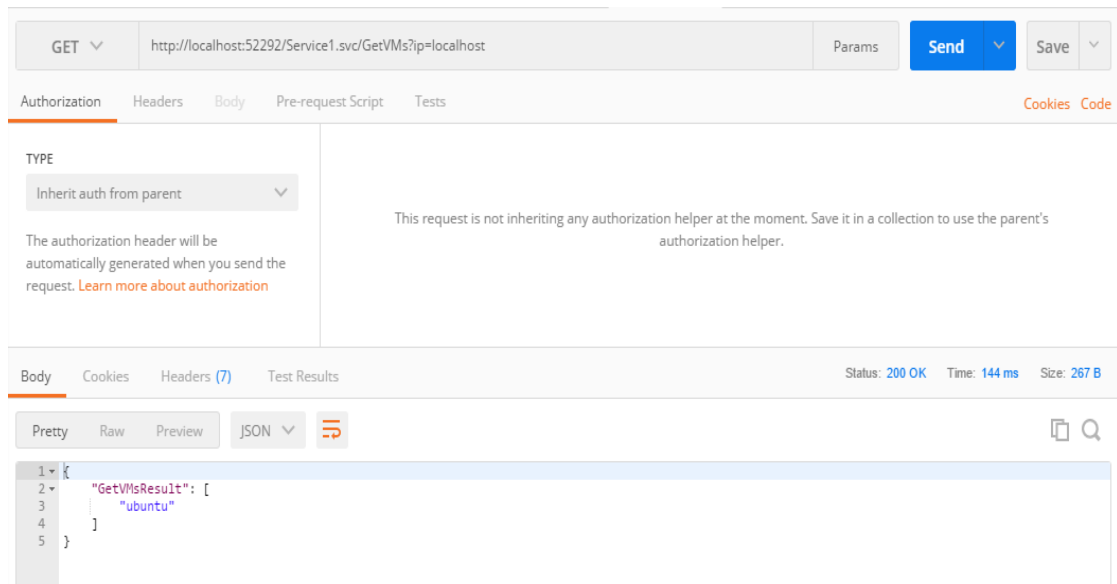


Figure 4.5. Successfully getting all the VM List

4.2.7.6 Take Snapshot VM

This API avails the user to create snapshots for a VM.

URL: http://localhost:52292/Service1.svc/TakeSnapshot?ip=localhost&vm=ubuntu&ssname=newSnapshotUbuntu&ssdesc=newSnapshotIsCreated

Action Performed: Get

Response Type: Json

Request Type: Json

Response Body:

This API returns the below response, if the request is successful.

```

{
  "TakeSnapshotResult": {
    "StatusCode": 0,
    "StatusMessage": "METHOD SUCCEEDED"
  }
}

```

4.2.7.7 Clone VM

This API will create a clone of a particular VM. Figure 4.6 shows the screenshot of successful cloning of an image to the virtual box.

URL: http://localhost:52292/Service1.svc/CloneVMs?ip=localhost&vm=ubuntu

Action Performed: Post

Response Type: Json

Request Type: Json

Response Body:

Returns the below response, if the request is successful.

```
{
  "RevertSnapshotResult": {
  "StatusCode": 0,
  "StatusMessage": "METHOD SUCCEEDED"
}
```

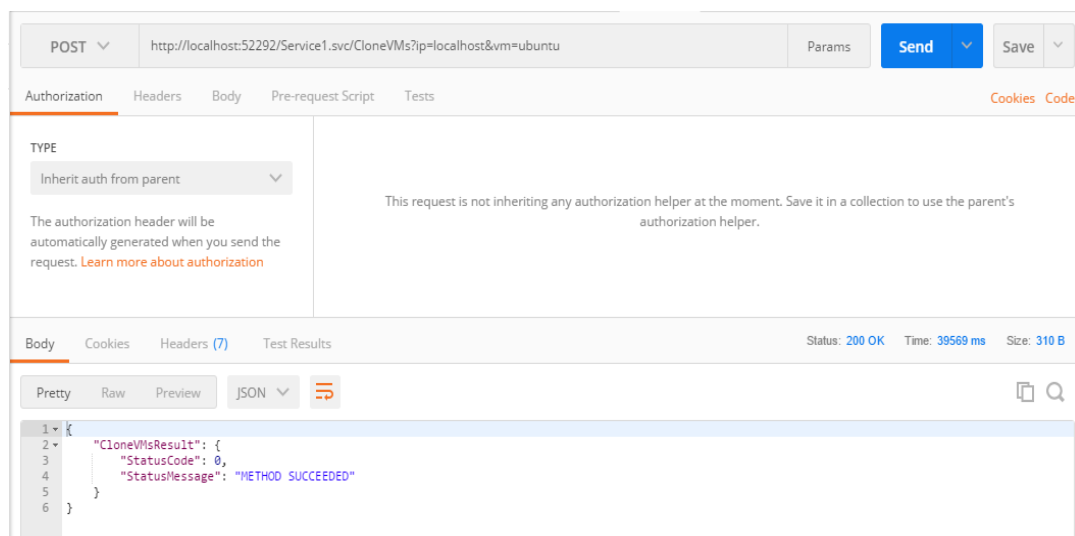


Figure 4.6. Clone has been successfully created

4.2.7.8 Restore/Revert VM

This API assists the user to revert the current state of the machine to a snapshot. The virtual machine is reverted to the state when the specified snapshot has been created.

URL: `http://localhost:52292/Service1.svc/RevertSnapshot?ip=localhost&vm=ubuntu&sname=newSnapshotUbuntu`

Action Performed: Post

Response Type: Json

Request Type: Json

Response Body:

Returns the below response, if the request is successful.

```
{
  "RevertSnapshotResult": {
```

```
"StatusCode": 0,  
"StatusMessage": "METHOD SUCCEEDED"  
} }
```

4.2.7.9 Get All the Snapshots

This API removes all the snapshots associated in the Virtual Box.

URL: <http://localhost:52292/Service1.svc/GetAllSnapshots?ip=localhost&vm=ubuntu>

Action Performed: Get

Response Type: Json

Request Type: Json

Response Body:

This API returns the below response if the request is successful.

```
{  
  "GetAllSnapshotsResult": [  
    "newSnapshotUbuntu",  
    "newSnapshotUbuntu1",  
    "newSnapshotUbuntu2",  
    "newSnapshotUbuntu3"  
  ]  
}
```

4.2.7.10 Stop VM

This API will stop VM; user should provide the VMName which he/she wants to stop.

URL: <http://localhost:52292/Service.svc/StopVM?ip=localhost&name=ubuntu> ubuntu

Action Performed: Post

Response Type: Json

Response Body:

This API returns the below response if the request is successful.

```
{  
  "StopVMResult {  
    "StatusCode": 0,  
    "StatusMessage": "METHOD SUCCEEDED"  
  }  
}
```

5.1 Dataset Description

The fast adoption of cloud data center leads to certain variation in workload structure and as an outcome in design and operation. The dataset contain the metrics from distributed data center from Bitbrains [22]. It is a service provider which specializes in business computation and managed hosting for enterprises. Cloud utilities can be accessed by billion people. Figure 4.7 shows the dataset.

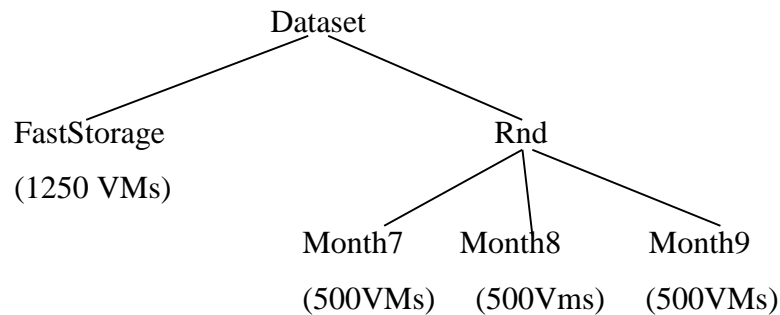


Figure 5.1. Dataset

FastStorage dataset contains 1250 virtual machines which are connected to fast storage area network (SAN) storage devices while Rnd dataset is distributed into 3 sub-categories which contains 500 Virtual Machines each that are connected to fast SAN or much slower network area storage devices [23].

5.2 Dataset Parameters

Our basic configuration relied on analysing the performance of datacenter with respect to CPU and memory for the above dataset.

For each dataset, it contains 11 parameters per VM

- Timestamp in milliseconds,
- Number of CPU cores,
- CPU capacity provisioned in MHz,
- CPU usage in MHz,

- CPU usage in percentage,
- Memory capacity provisioned in KB,
- Memory usage in KB,
- Disk read throughput in KB per sec,
- Disk write throughput in KB per sec,
- Network received throughput in KB per sec,
- Network transmitted throughput in KB per sec.

5.3 Flow of Research Methodology

In order to analyse the workload at datacentre. Figure 5.2 depicts the overall flow

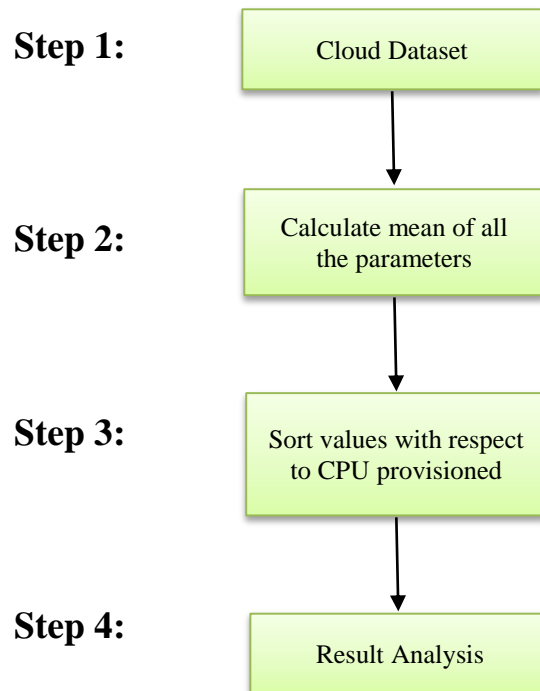


Figure 5.2 Research methodology flow

In Step 1, cloud dataset refers the following datasets- FastStorage, Rnd month7 (data collected in month july), Rnd month8, Rnd month9. In Step 2, average of all the parameters for individual VMs have been calculated. This step is performed to analyse the average usage of various resources. In Step 3, values have been sorted in descending

order with respect to CPU provisioned. This step is performed in order to compare the CPU usage against CPU provisioned as per decreasing CPU capacity.

In the result analysis step, the relationship between CPU and memory usage are analysed and it is tried to find out that how much CPU and memory are actually used in the cloud data center. In this phase, load factor of all the virtual machines are calculated.

5.4 Experimental Results

This section discusses the result analysis that basically focuses on study of workloads facilitated in cloud data centers.

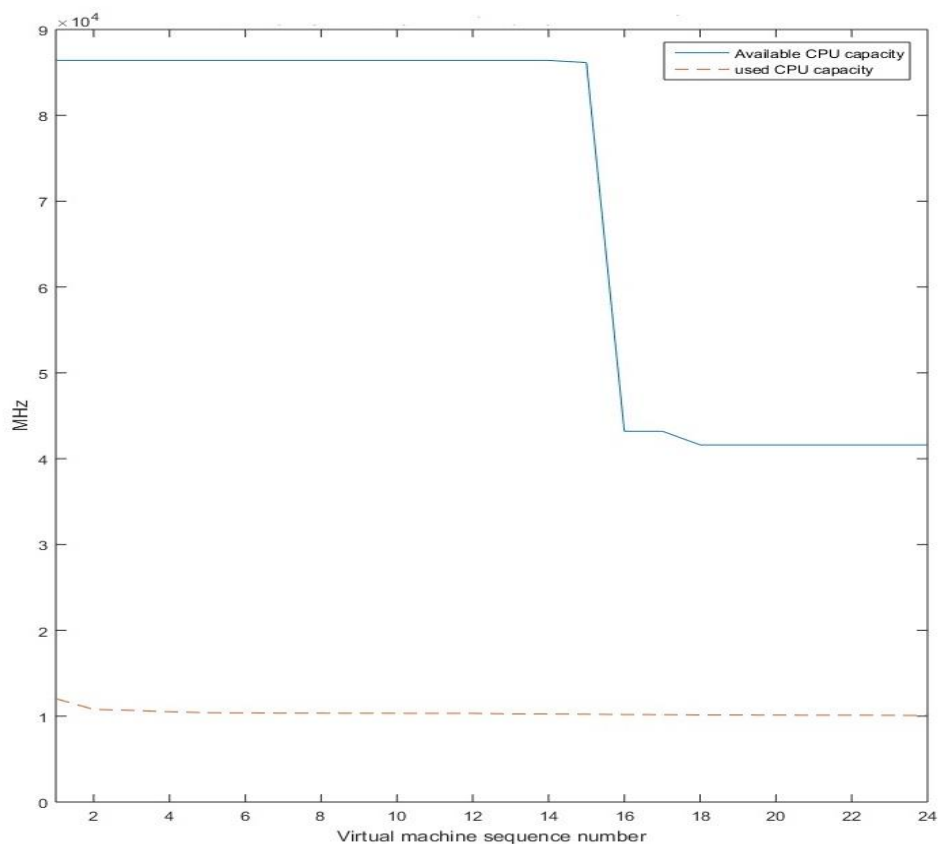


Figure 5.3(a) CPU capacity usage analysis for high capacity machines in FastStorage dataset

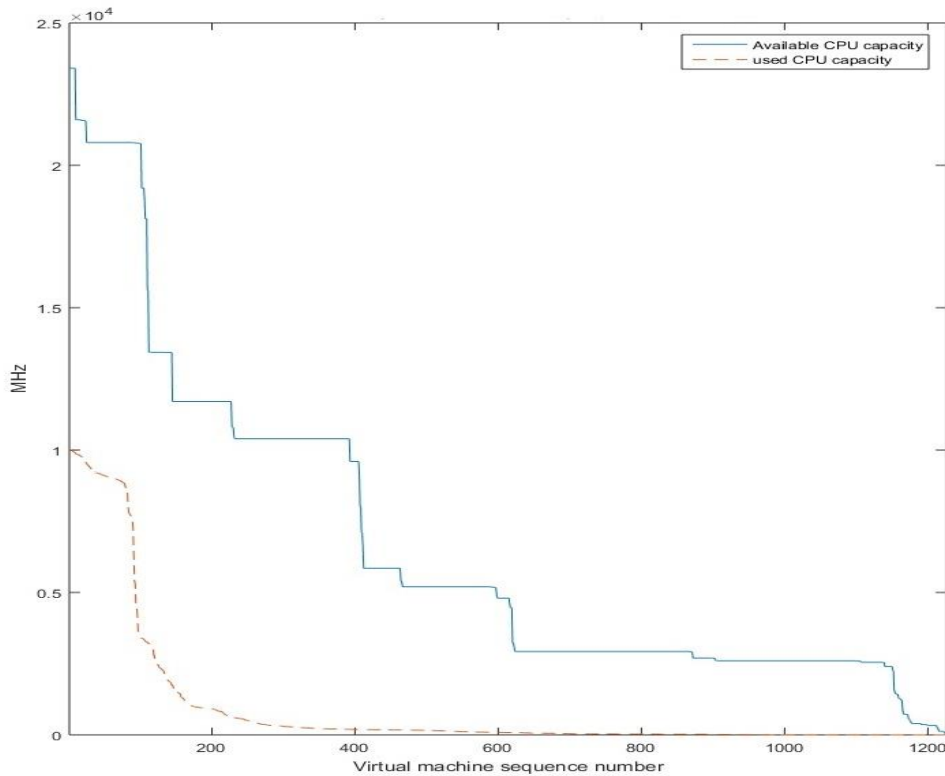


Figure 5.3(b) CPU capacity usage analysis for medium to low capacity machines in FastStorage dataset

In Figure 5.3, x-axis in the graph represents sequence of virtual machines and y-axis represents CPU capacity in MHz. The result from experiment shows that the available CPU Capacity and CPU utilization for the given dataset FastStorage decreases with the increase in number of VMs. The graph follows a particular pattern in both the cases when the number of VMs is increased. Though Figure 5.3(a) and Figure 5.3(b) follow similar trends in CPU utilization, it is observed the difference in CPU utilization values. Available CPU capacity reaches maximum value to 8.6 MHz while the maximum CPU usage reaches maximum value to only 1.1 MHz for FastStorage dataset. In Figure 5.3(a), available CPU capacity of VMs is maximum at 8.6MHz and it is constant till 14 VMs. After that, available CPU capacity is reduced to 4.2MHz and is constant till 24 VMs, while maximum CPU usage is only 1.1MHz and is constant from 1 VM to 24 VMs. According to Figure 5.3(b), with an increase in number of VMs available CPU capacity is continuously decreasing from 2.3MHz to 0MHz and CPU usage is also decreasing from 1MHz to 0MHz. From 25 VMs to 300 VMs CPU usage capacity is reducing from 1MHz to 0.1MHz. After 300 VMs till 1250 VMs, it is gradually leading to 0. For most (over 75%) VMs, the CPU usage is less than 0.1MHz. Moreover, only 20% of VMs have more

CPU usage between 0.1MHz-2MHz. These observations suggest that for most of the VMs the CPU usage capacity is often low. Hence, there is wastage of more CPU resources.

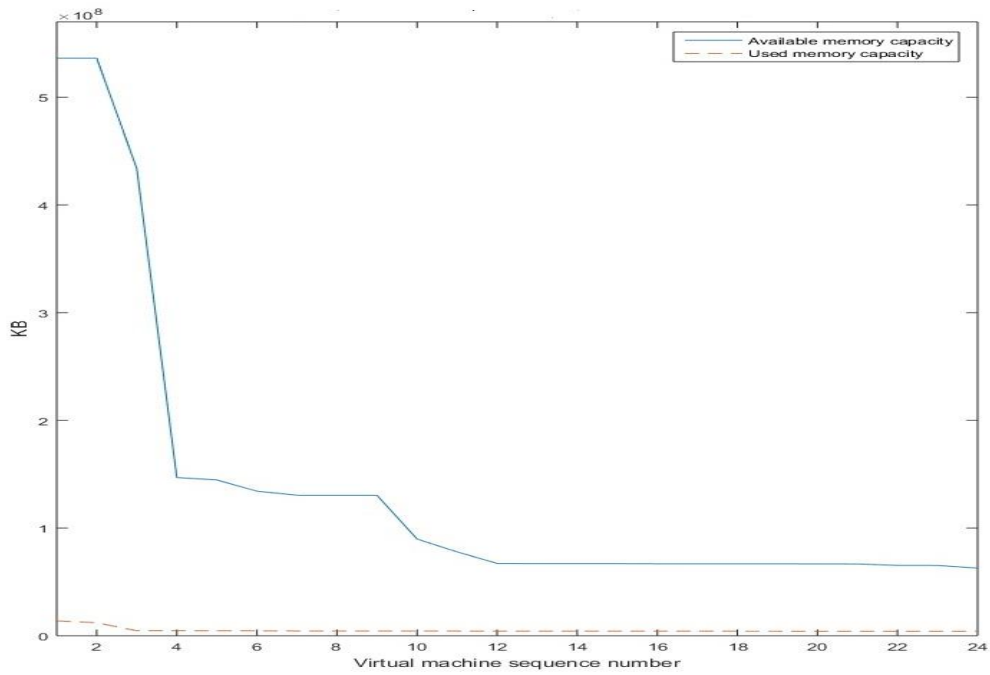


Figure 5.4(a) Memory capacity usage analysis for high capacity machines in FastStorage dataset

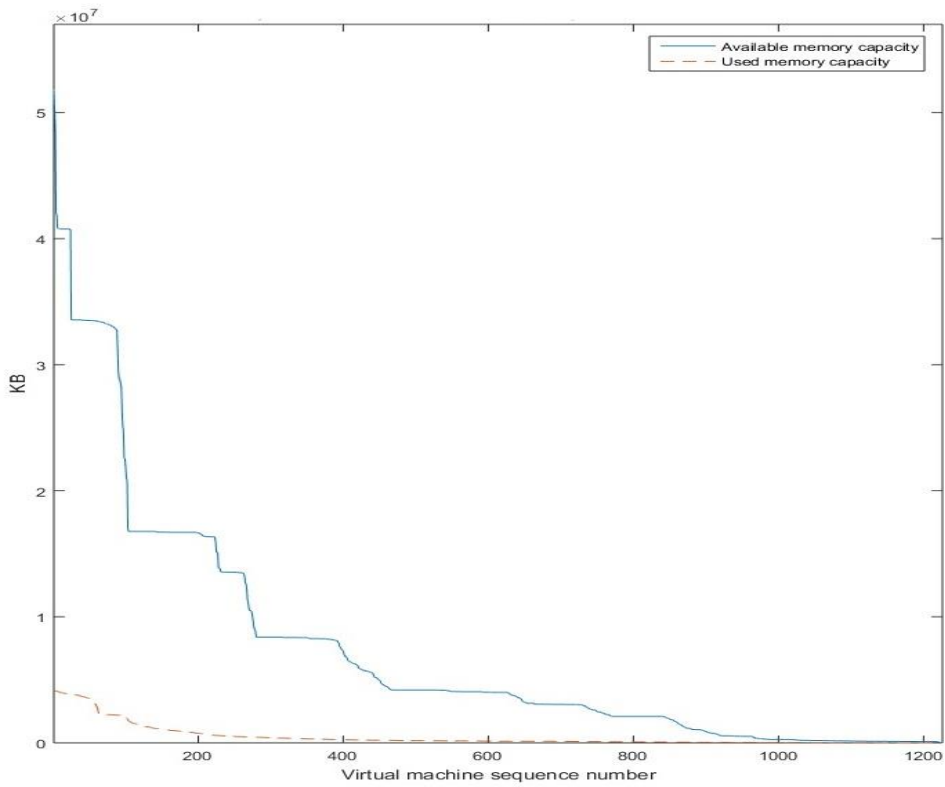


Figure 5.4(b) Memory capacity usage analysis for medium to low capacity machines in FastStorage dataset

In Figure 5.4, x-axis in the graph represents sequence of virtual machines and y-axis represents memory capacity in KB. The result from experiment shows that the available memory capacity and memory utilization for the given dataset FastStorage, decreases with the increase in number of VMs. The graph follows a particular pattern in both the cases when the number of VMs is increased. Though Figure 5.4(a) and Figure 5.4(b) follow similar trends in memory utilization, it is observed the difference in memory utilization values. Available memory capacity reaches maximum value to 5.2KB while the maximum memory usage reaches maximum value to only 0.1KB for FastStorage dataset. In Figure 5.4(a), available memory capacity of VMs is maximum at 5.2KB and it is constant till 3 VMs. After that, available memory capacity is reduced to 1.2KB and is constant till 9 VMs while maximum memory usage is only 0.1KB till 3 VMS. After that the value is abruptly decreased to 1.5KB and remain constant till 10 VMS. After that it again decreased to 0.6KB and remain constant till 24 VMs. According to Figure 5.4(b), with an increase in number of VMs available memory capacity is continuously decreasing from 2.3MHz to 0MHz and memory usage is also decreasing from 1MHz to 0MHz. From 25 VMs to 300 VMs memory usage capacity is reducing from 1MHz to 0.1MHz. After 200 VMs till 1250 VMs, it is gradually leading to 0. Over 80% of the memory resource is not being utilized. For most (over 80%) VMs, the memory usage is less than 0.1KB. These observations suggest that for most of the VMs the memory usage capacity is often low. Hence, there is wastage of more memory resources.

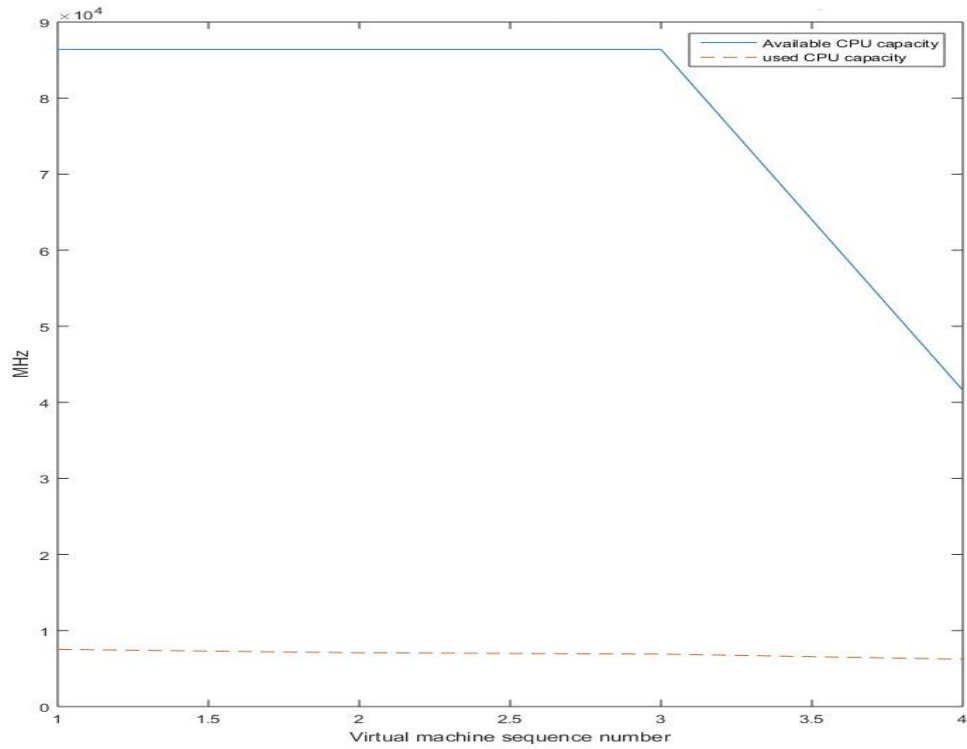


Figure 5.5(a) CPU capacity usage analysis for high capacity machines in Rnd month7 dataset

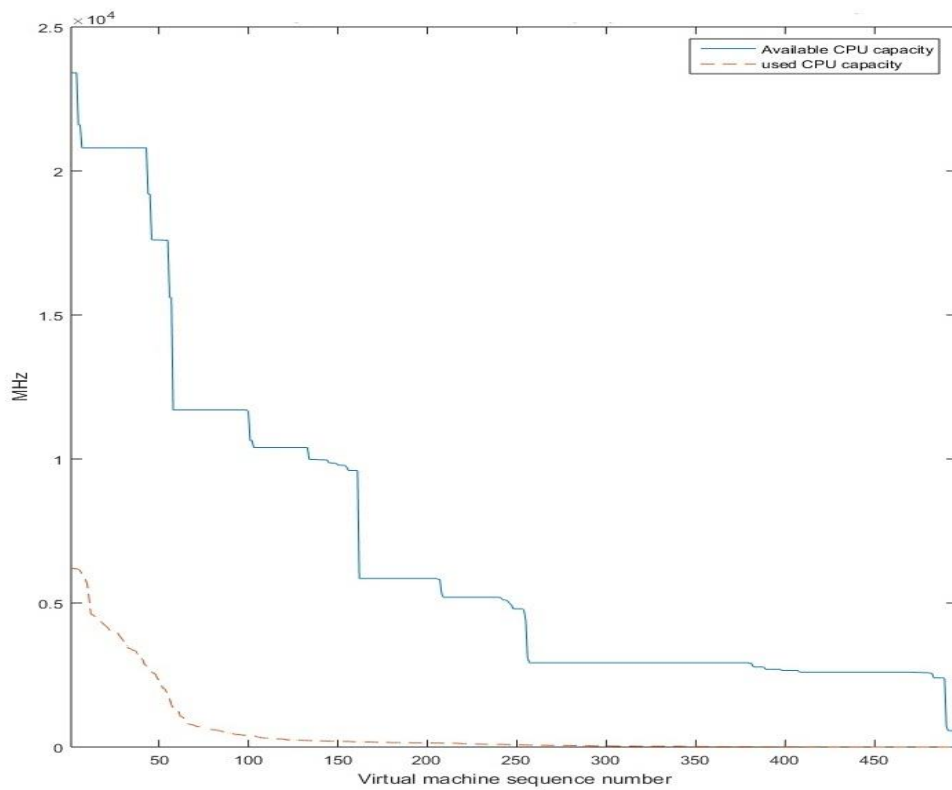


Figure 5.5(b) CPU capacity usage analysis for medium to low capacity machines in Rnd month7 dataset

In Figure 5.5, x-axis in the graph represents sequence of virtual machines and y-axis represents CPU capacity in MHz. Available CPU capacity reaches maximum value to 8.6MHz, while the maximum CPU usage reaches maximum value to only 0.7MHz for Rnd month7 dataset. In Figure 5.5(a), available CPU capacity of VMs is maximum at 8.6MHz and is constant till 3 VMs. After that, available CPU capacity is reduced to 4.1MHz and is constant till 4 VMs, while maximum CPU usage is only 0.7MHz and is constant from 1 VM to 4 VMs. According to Figure 5.5(b), with an increase in number of VMs available CPU capacity is continuously decreasing from 2.3MHz to 0MHz and CPU usage is also decreasing from 0.6MHz to 0MHz. From 5 VMs to 100 VMs CPU usage capacity is reducing from 0.6MHz to 0.1MHz. After 100 VMs till 500 VMs, the curve is leading to 0. For most (over 80%) VMs, the CPU usage is less than 0.1MHz. Moreover, only 20% of VMs have CPU usage between 0.1MHz-0.6MHz. These observations suggest that for most of the VMs the CPU usage capacity is often low. Hence, there is wastage of more CPU resources.

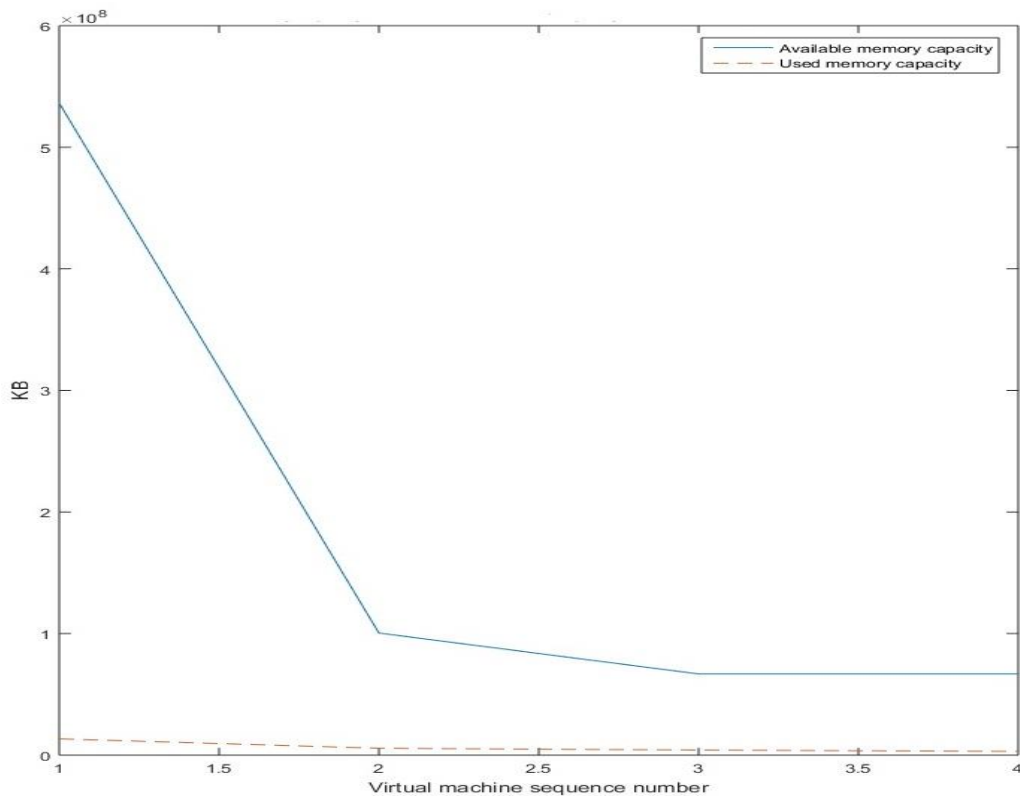


Figure 5.6(a) Memory capacity usage analysis for high capacity machines in Rnd month7 dataset

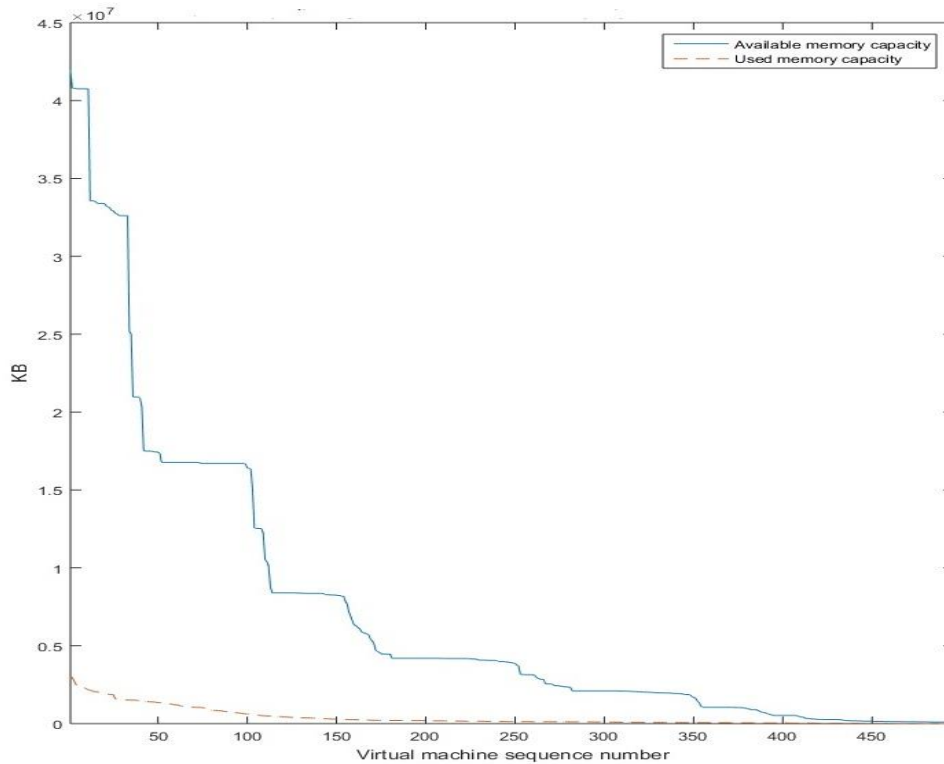


Figure 5.6(b) Memory capacity usage analysis for medium to low capacity machines in Rnd month7 dataset

In Figure 5.6, x-axis in the graph represents sequence of virtual machines and y-axis represents memory capacity in KB. Available memory capacity reaches maximum value to 5.3KB while the maximum memory usage reaches maximum value to only 0.1KB for Rnd month7 dataset. In Figure 5.6(a), available memory capacity of VMs is maximum at 5.3KB and is constant till 2 VMs. After that, available memory capacity is reduced to 1KB and is constant till 4 VMs, while maximum memory usage is only 0.1KB and is constant from 1 VM to 4 VMs. According to Figure 5.6(b), with an increase in number of VMs available memory capacity is continuously decreasing from 4.1KB to 0KB and memory usage is also decreasing from 0.3KB to 0KB. From 5 VMs to 100 VMs memory usage capacity is reducing from 0.3KB to 0.1KB. After 100 VMs till 500 VMs, the curve is leading to 0. Hence, there is a lot of memory wastage. Over 80% of the memory resource is not being utilized. For most (over 80%) VMs, the memory usage is less than 0.1KB. These observations suggest that for most of the VMs the memory usage capacity is often low. Hence, there is wastage of more memory resources.

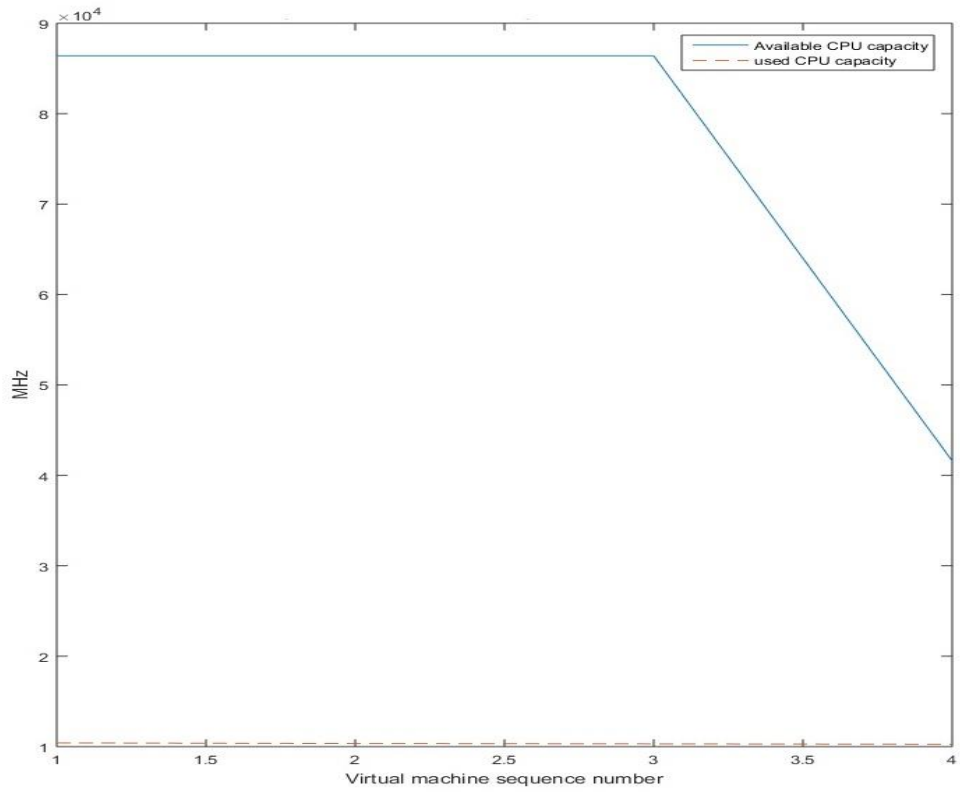


Figure 5.7(a) CPU capacity usage analysis for high capacity machines in Rnd month8 dataset

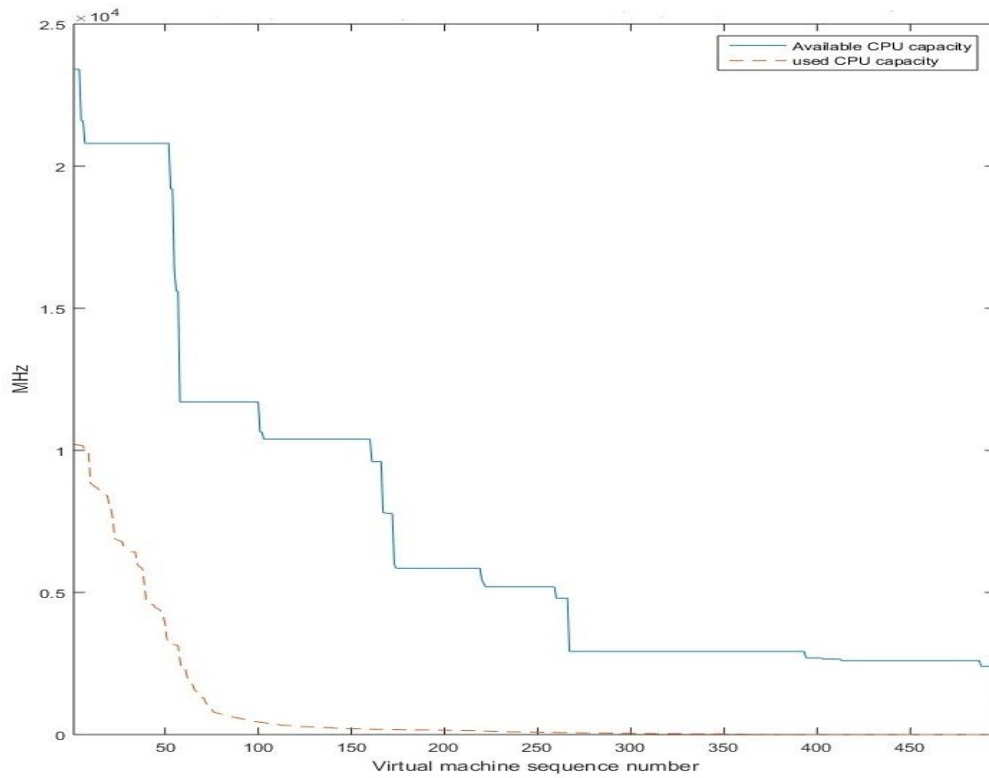


Figure 5.7(b) CPU capacity usage analysis for medium to low capacity machines in Rnd month8 dataset

In Figure 5.7, x-axis in the graph represents sequence of virtual machines and y-axis represents CPU capacity in MHz. Available CPU capacity reaches maximum value to 8.6MHz, while the CPU usage reaches maximum value to only 1MHz for Rnd month8 dataset. In Figure 5.7(a), available CPU capacity of VMs is maximum at 8.6MHz and is constant till 3 VMs. After that, available CPU capacity is reduced to 4.1MHz and is constant till 4 VMs, while maximum CPU usage is only 1MHz and is constant from 1 VM to 4 VMs. According to Figure 5.7(b), with an increase in number of VMs CPU capacity is continuously decreasing from 2.3MHz to 0MHz and CPU usage is also decreasing from 1MHz to 0MHz. From 5 VMs to 90 VMs CPU usage capacity is reducing from 1MHz to 0.1MHz. After 90 VMs till 500 VMs, it is approximate to 0. For most (over 80%) VMs, the CPU usage is less than 0.1MHz. Moreover, only 20% of VMs have more CPU usage between 0.1MHz-1MHz. These observations suggest that for most of the VMs the CPU usage capacity is often low. Hence, there is wastage of more CPU resources.

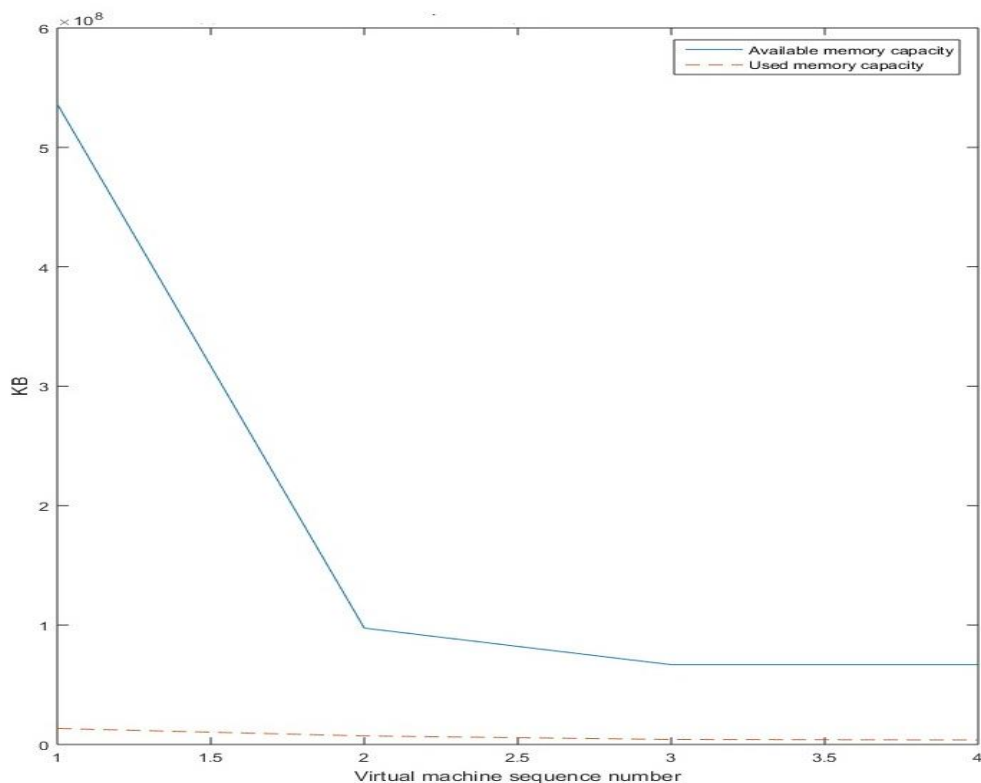


Figure 5.8(a) Memory capacity usage analysis for high capacity machines in Rnd month8 dataset

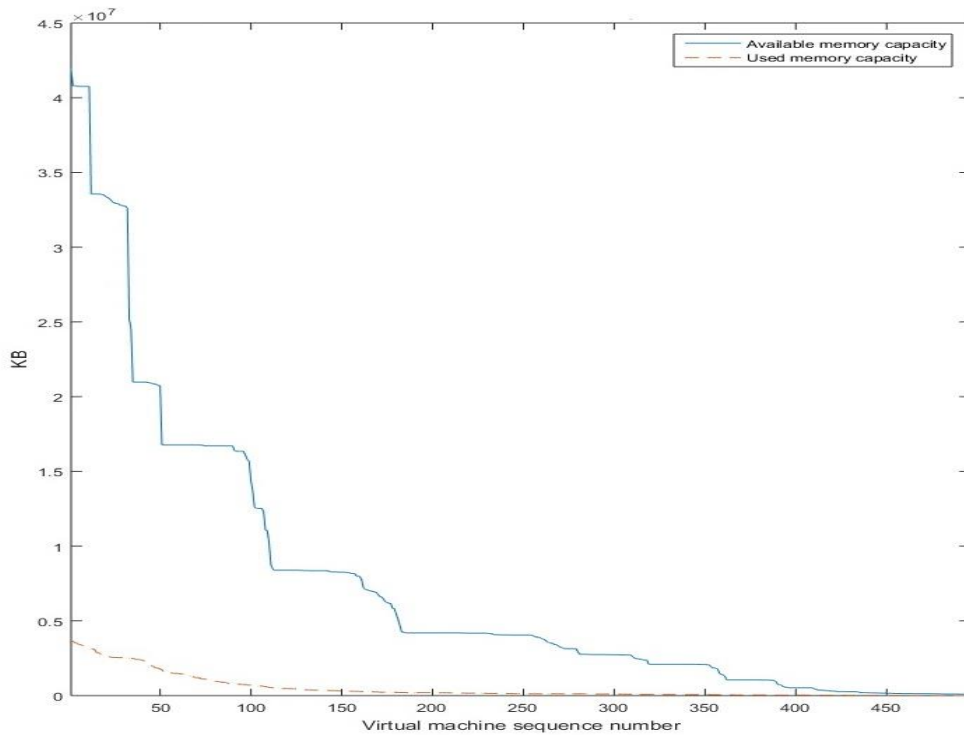


Figure 5.8(b) Memory capacity usage analysis for medium to low capacity machines in Rnd month8 dataset

In Figure 5.8, x-axis in the graph represents sequence of virtual machines and y-axis represents memory capacity in KB. Available memory capacity reaches maximum value to 5.2KB while the memory usage reaches maximum value to only 0.1KB for Rnd month8 dataset. In Figure 5.8(a), available memory capacity of VMs is maximum at 5.2KB and reduced to 1KB at 3 VMs. After that, available memory capacity is 1KB and is constant till 4 VMs while maximum memory usage is only 1KB and is constant from 1 VM to 4 VMs. According to Figure 5.8(b), with an increase in number of VMs available memory capacity is continuously decreasing from 4.1KB to 0KB and memory usage is also decreasing from 0.4KB to 0KB. From 5 VMs to 100 VMs memory usage capacity is reducing from 0.5KB to 0.1KB. After 100 VMs till 500 VMs, it is approximate to 0. For most (over 80%) VMs, the memory usage is less than 0.1KB. These observations suggest that for most of the VMs the memory usage capacity is often low. Hence, there is wastage of more memory resources.

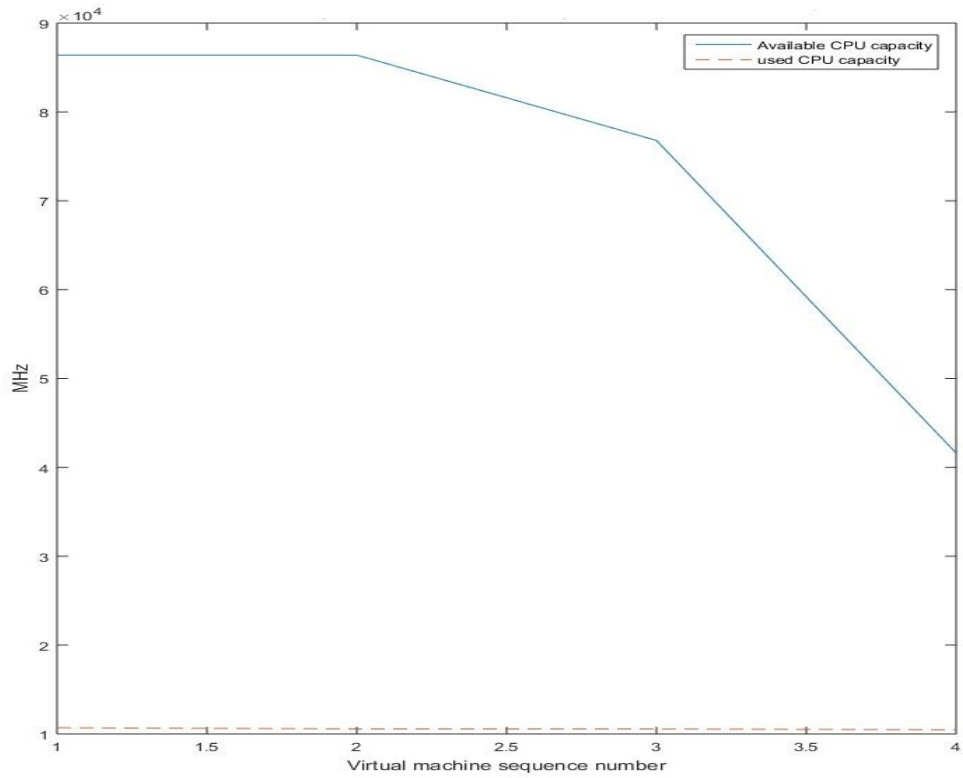


Figure 5.9(a) CPU capacity usage analysis for high capacity machines in Rnd month9 dataset

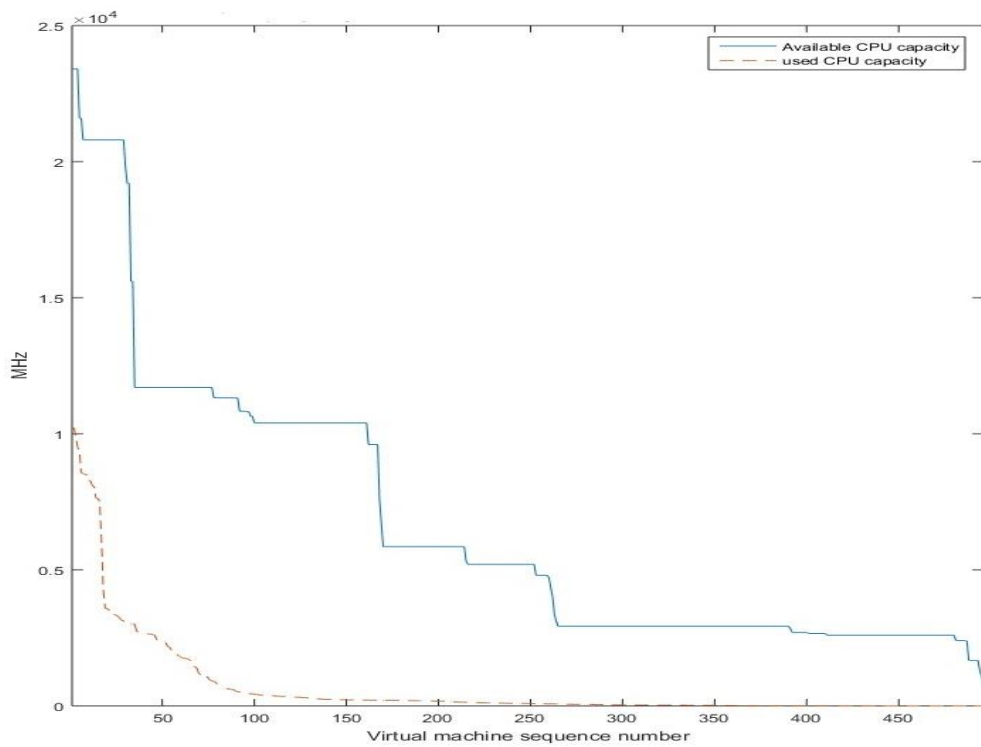


Figure 5.9(b) CPU capacity usage analysis for medium to low capacity machines in Rnd month9 dataset

In Figure 5.9, x-axis in the graph represents sequence of virtual machines and y-axis represents CPU capacity in MHz. Available CPU capacity reaches maximum value to 8.6MHz, while the CPU usage reaches maximum value to only 1MHz for Rnd month9 dataset. In Figure 5.9(a), available CPU capacity of VMs is maximum at 8.6MHz and it constant till 3 VMs. After that, available CPU capacity is reduced to 4.1MHz and is constant till 4 VMs while maximum CPU usage is only 1MHz and is constant from 1 VM to 4 VMs. According to Figure 5.9(b), with an increase in number of VMs available CPU capacity is continuously decreasing from 2.3MHz to 0MHz and CPU usage is also decreasing from 1MHz to 0MHz. From 5 VMs to 90 VMs CPU usage capacity is reducing from 1MHz to 0.1MHz. After 90 VMs till 500 VMs, it is approximate to 0. Hence, there is a lot of CPU wastage. For most (over 80%) VMs, the CPU usage is less than 0.1MHz. Moreover, only 20% of VMs have more CPU usage between 0.1MHz-0.6MHz. These observations suggest that for most of the VMs the CPU usage capacity is often low. Hence, there is wastage of more CPU resources.

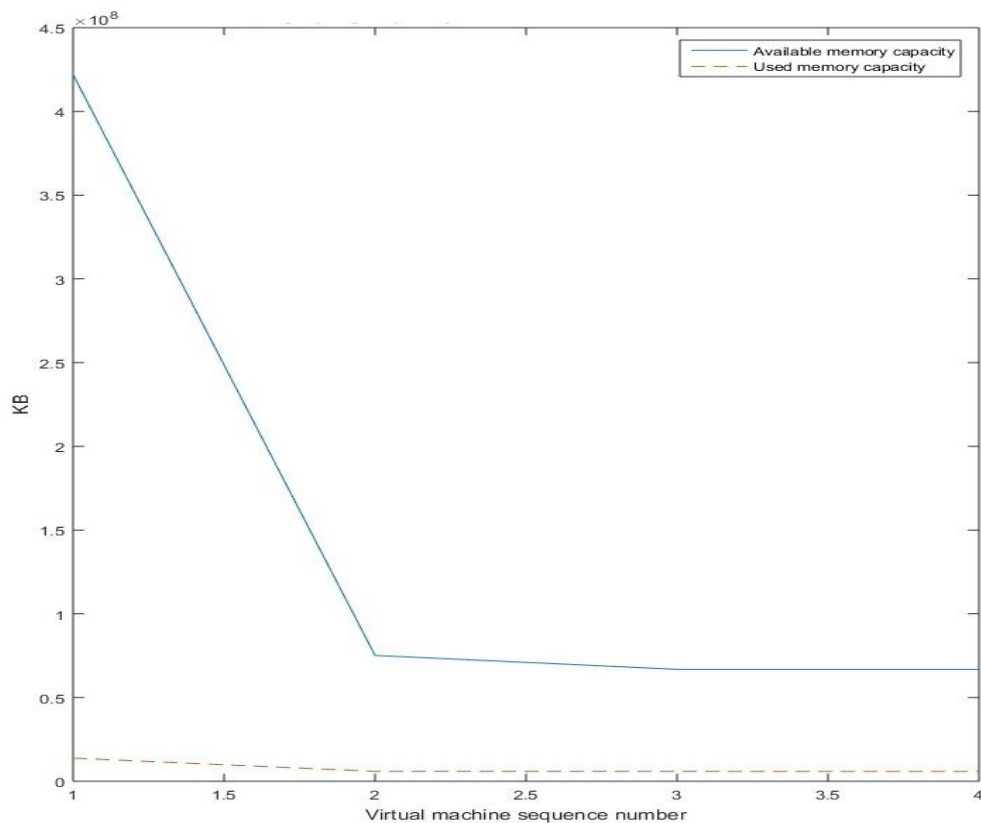


Figure 5.10(a) Memory capacity usage analysis for high capacity machines in Rnd month9 dataset

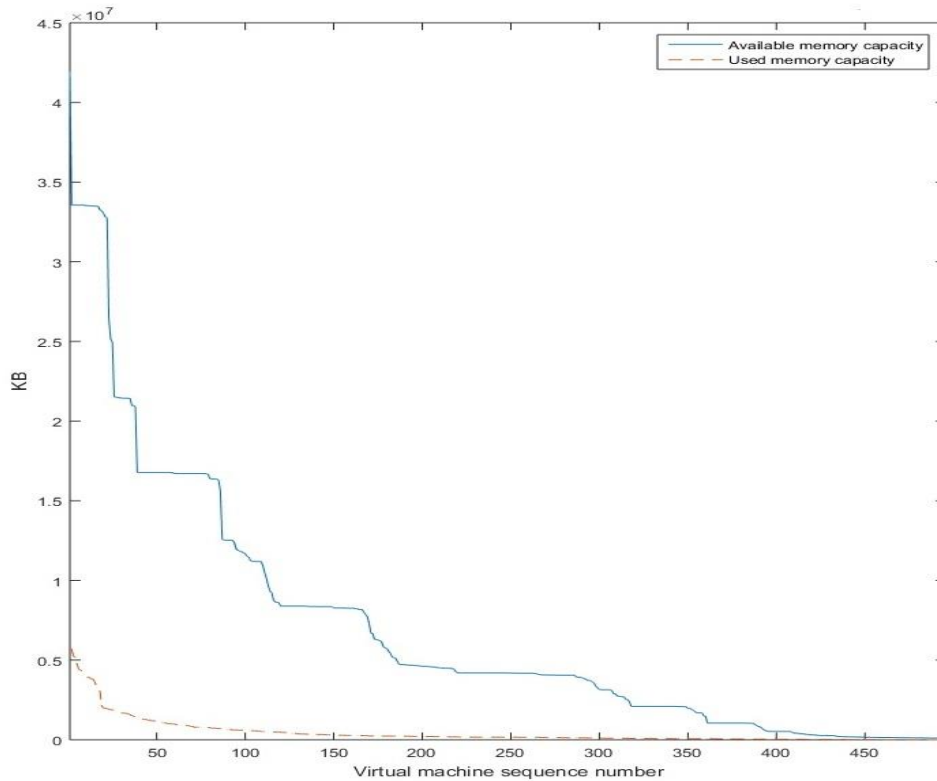


Figure 5.10(b) Memory capacity usage analysis for medium to low capacity machines in Rnd month9 dataset

In Figure 5.10, x-axis in the graph represents sequence of virtual machines and y-axis represents memory capacity in KB. Available memory capacity reaches maximum value to 4.2KB, while the memory usage reaches maximum value to only 0.1KB for Rnd month8 dataset. In Figure 5.10(a), available memory capacity of VMs is maximum at 4.2KB and reduced to 0.7KB at 2 VMs. After that, available memory capacity is 0.7KB and is constant till 4 VMs while maximum memory usage is only 0.1KB and is constant from 1 VM to 4 VMs. According to Figure 5.10(b), with an increase in number of VMs available memory capacity is continuously decreasing from 4.2KB to 0KB and memory usage is also decreasing from 0.6KB to 0KB. From 5 VMs to 100 VMs memory usage capacity is reducing from 0.6KB to 0.1KB. After 100 VMs till 500 VMs, it is approximate to 0. For most (over 85%) VMs, the memory usage is less than 0.1KB. These observations suggest that for most of the VMs the memory usage capacity is often low. Hence, there is wastage of more memory resources.

Table 5.1 Performance analysis with different datasets

VM CPU load factor	FastStorage	Rnd Month7	Rnd Month8	Rnd Month9
0	0.72	0.8	0.6	0.6
5	76.24	78.6	77	77.2
10	7.2	6.4	6.6	6.2
20	3.52	5.8	2.4	7.6
30	1.84	5.8	2.6	2.4
40	1.84	1.8	4.2	2.6
50	8	0.2	5.8	1.6
60	0.56	0.4	0.6	1.4
70	0	0.2	0.2	0.2
80	0.08	0	0	0.2
90	0	0	0	0
100	0	0	0	0

Table 5.1 shows CPU load factor of VMs. On the basis of above values, In FastStorage dataset out of 1250 VMs, 953 VMs have 5% CPU load, 100 VMs have 50% load, 90 VMs have 10% load, 44 VMs have 20% load, 23 VMs have 40% load, 20 VMs have 60% load, 9 VMs have 0% load and only 1 VM has 80% load.

In Rnd dataset out of 500 VMs, 388 VMs have 5% load, 27-32 VMs have 10 or 30% load, 12-14 VMs have 40 or 50% load, 4 VMs have 60% and 0% load each, only 1 VM has 70-80% load each.

It can be concluded that maximum number of VMs have 5% CPU load for both the mentioned datasets and very less number of VMs have 70-100% CPU load. There is only 1 VM who has 80% CPU load for the above dataset. As an example, suppose there is VM1 having 5% load, then 95% of its resources are free and VM2 having 80% load. VM1 is having 95% free resources and only 5% is being occupied by VM1. So, free resources of VM1 should be assigned to VM2 to make efficient use of resources. The VMs that are having 60-80% load can get the resources from the one having 5-20% load. This will help in utilizing the resources more efficiently.

This chapter summarizes the research work proposed in this thesis.

6.1 Conclusion

- The developers are provided an environment for test run as well as debugging. In users own machine the tests can run or debug for as long as user wants. Hence, the work provided in this thesis using virtual box in cloud infrastructure, saves significant time and effort in debugging thereby improves efficiency.
- This thesis emphasize on performance analysis for workload of cloud data center of business-critical applications. It is found that maximum number of VMs have less CPU load (only 5%) which is not efficient as CPU resource is not even getting utilized by 70%. There are very less or few number of VMs which are utilizing 80% CPU. To conclude, it is advised that VM rescheduling is required to utilize data center resources more optimally.

References

- [1] L. J. Zhang and Q. Zhou, "Cloud Computing Open Architecture," *IEEE International Conference on Web Services*, pp. 607-616, 2009.
- [2] S. Patidar, D. Rane and P. Jain, "A Survey Paper on Cloud Computing," *IEEE Second International Conference on Advanced Computing & Communication Technologies*, pp. 394-398, 2012.
- [3] P. Mell and T. Grance, "The NIST definition of Cloud Computing," NIST Special Publication 800-145, 2011.
- [4] Q. Zhang, L. Cheng and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7-18, 2010.
- [5] "Overview of Cloud Computing", [Online]. Available: <https://en.wikipedia.org/wiki/Cloudcomputing>. [Accessed 15 3 2018].
- [6] L. A. Barroso, J. Clidaras and U. Holzle, "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines," *Synthesis Lectures on Computer Architecture Morgan & Claypool Publishers*, vol. 8, no. 3, pp. 1-154, 2013.
- [7] "General electronics," [online]. Available: <https://www.britannica.com/topic/General-Electric>. [Accessed 16 6 2018].
- [8] "General Bell labs," [online]. Available: <https://www.bell-labs.com/>. [Accessed 17 6 2018].
- [9] G. Neiger, A. Santoni, F. Leung, D. Rodgers and R. Uhlig, "Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization," *Intel Technology Journal*, vol. 10, no. 3, pp. 167-177, 2006.
- [10] "IBM," [online]. Available: <https://www.ibm.com/us-en/>. [Accessed 17 6 2018].
- [11] W. Chen, H. Lu, L. Shen, Z. Wang, N. Xiao and D. Chen, "A Novel Hardware Assisted Full Virtualization Technique," *9th International Conference for Young Computer Scientists*, pp. 1292–1297, 2008.
- [12] Wikipedia, "Microsoft Visual Studio," Available: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio. [Accessed 12 2 2018].
- [13] S. Patel, R. Patel, H. Patel and S. Vahora, "CloudAnalyst: A Survey of Load Balancing Policies," *International Journal of Computer Applications*, vol. 117, no. 21, 2015.

- [14] A. Uta, and H. Obaseki, "A Performance Study of Big Data Workloads in Cloud Datacenters with Network Variability," *Companion of ACM/SPEC International Conference on Performance Engineering*, pp. 113-118, 2018.
- [15] B. H. Bhavani and H. S. Guruprasad, "Efficient Resource Scheduling under Varying workloads in Cloud Data Center," *International Journal of Emerging Engineering Science and Technology*, vol. 1, no. 1, pp. 40-45, 2015.
- [16] E. Elghoneimy, O. Bouhali and H. Alnuweiri, "Resource allocation and scheduling in cloud computing," *International Conference on Computing, Networking and Communications*, pp. 309-314, 2012.
- [17] D. Kargatzis, S. Sotiriadis and E. G. Petrakis, "Virtual machine migration in heterogeneous clouds: from openstack to VMWare," *38th IEEE Sarnoff Symposium*, pp. 1-6, 2017.
- [18] N. Chen, X. Fang and X. Wang, "A cloud computing resource scheduling scheme based on estimation of distribution algorithm," *Second International Conference on Systems and Informatics*, pp. 304-308, 2014.
- [19] S. Singh and I. Chana, "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges," *Journal of Grid Computing*, vol. 14, no. 2, pp. 217-264, 2016.
- [20] D. Magalhães, R.N. Calheiros, R. Buyya and D. G. Gomes, "Workload modeling for resource usage analysis and simulation in cloud computing," *Computers & Electrical Engineering*, vol. 47, pp. 69-81, 2015.
- [21] D. Gmach, J. Rolia, L. Cherkasova and A. Kemper, "Workload Analysis and Demand Prediction of Enterprise Data Center Applications," *10th IEEE International Symposium on Workload Characterization*, pp. 171-180, 2007.
- [22] "GWA-T-12 Bitbrains," [Online]. Available: <http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains>. [Accessed 17 6 2018].
- [23] S. Shen, V. Beek and A. Iosup, "Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters," *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 465-474, 2015.

Efficient Resource Scheduling Framework for Dynamic Load Management in Cloud Data Centers

ORIGINALITY REPORT

8%

SIMILARITY INDEX

5%

INTERNET SOURCES

4%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

www.ds.ewi.tudelft.nl

Internet Source

2%

2

Submitted to CSU, Long Beach

Student Paper

1%

3

G.S.A. Talukder, A.-M.K. Pathan. "QoSIP: a QoS aware IP routing protocol for multimedia data", 2006 8th International Conference Advanced Communication Technology, 2006

Publication

1%

4

download.virtualbox.org

Internet Source

<1%

5

Dionisio Machado Leite, Maycon L. M. Peixoto, Bruno G. Batista, Bruno T. Kuehne, Carlos H. G. Ferreira. "The influence of resource allocation on cloud computing performance", Proceedings of the Symposium on Applied Computing - SAC '17, 2017

Publication

<1%

Submitted to Vel Tech University

6	Student Paper	<1%
7	Ilullo.com Internet Source	<1%
8	gwa.ewi.tudelft.nl Internet Source	<1%
9	Submitted to The Robert Gordon University Student Paper	<1%
10	Submitted to selby college Student Paper	<1%
11	www.cloudbus.org Internet Source	<1%
12	Siqi Shen, Vincent van Beek, Alexandru Iosup. "Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters", 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2015 Publication	<1%
13	Submitted to Aston University Student Paper	<1%
14	boiling.seas.ucla.edu Internet Source	<1%
15	Andreas Kohne. "Cloud-Föderationen", Springer Nature, 2018 Publication	<1%

16 Phaphoom, Nattakarn, Xiaofeng Wang, and Pekka Abrahamsson. "Foundations and Technological Landscape of Cloud Computing", ISRN Software Engineering, 2013. <1%
Publication

17 oceanrep.geomar.de <1%
Internet Source

18 hero.lecturer.pens.ac.id <1%
Internet Source

19 docplayer.net <1%
Internet Source

20 dl.acm.org <1%
Internet Source

21 link.springer.com <1%
Internet Source

22 dea.lib.unideb.hu <1%
Internet Source

23 ijccer.org <1%
Internet Source

24 www.internationalscienceindex.org <1%
Internet Source

Exclude bibliography On