Interference Aware Scheduling Of Tasks On Cloud

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

Master of Engineering

in

Computer Science and Engineering

Submitted By

Simran Setia

(801532050)

Under the supervision of:

Dr Rajesh Kumar

Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

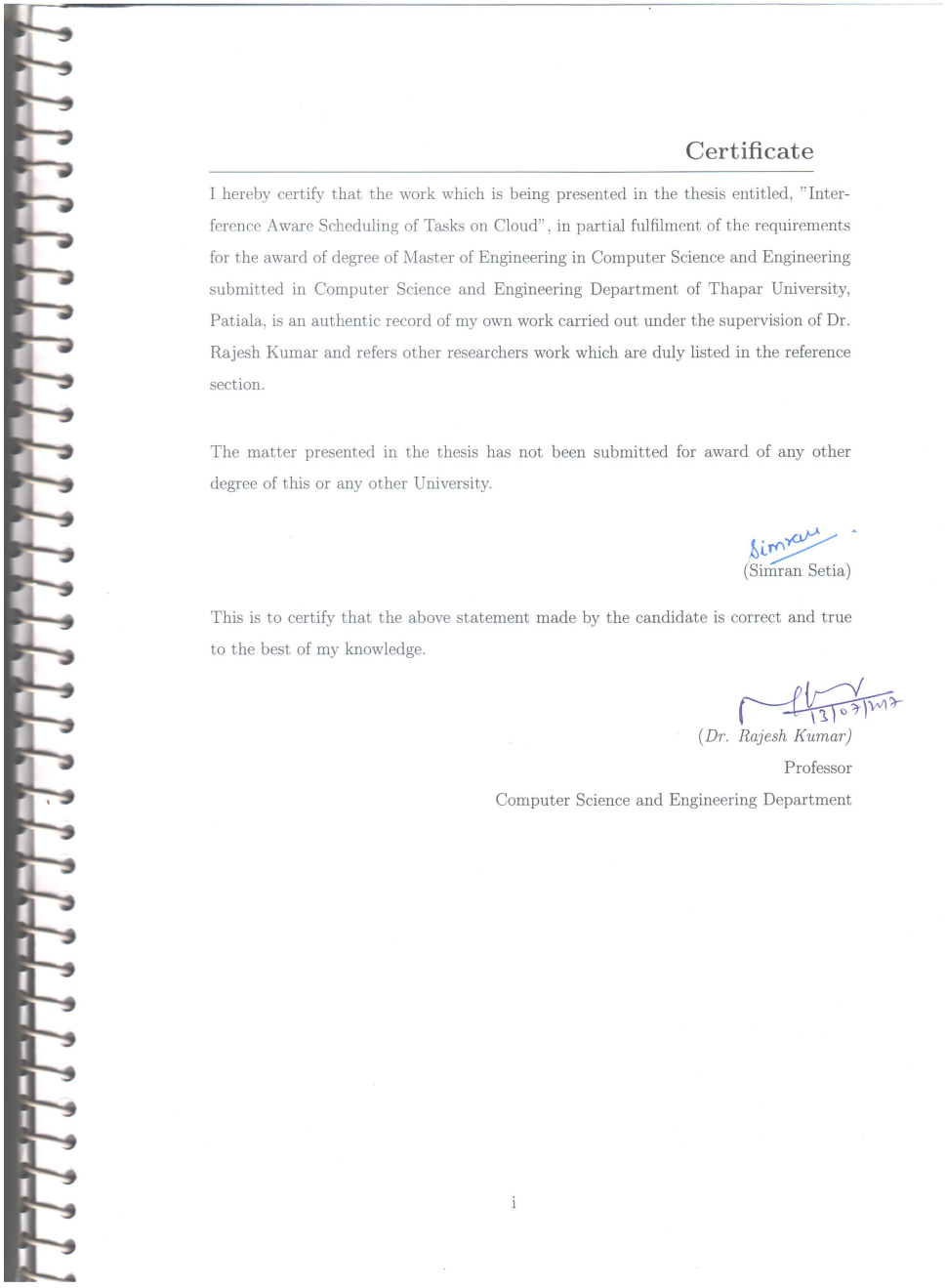
THAPAR UNIVERSITY

PATIALA-147001

July 2017

Certificate

002.png 002.png



Certificate

I hereby certify that the work which is being presented in the thesis entitled, "Interference Aware Scheduling of Tasks on Cloud", in partial fulfilment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Rajesh Kumar and refers other researchers 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.

Simran

(Simran Setia)

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

13/07/2017

(Dr. Rajesh Kumar)

Professor

Computer Science and Engineering Department

i

Acknowledgement

First, I would like to express my gratitude to my supervisor Dr. Rajesh Kumar for his invaluable advice and encouragement at every step of my M.E. program. This thesis would not have been possible without his guidance and support. And for this, I am truly grateful. He is a great mentor for my life as well.

I would like to express my gratitude to a PhD scholar Mr. Ashok Khunger. He has always been helpful with new technologies in carrying out the research work. I also thank my fellow ME colleague Ms. Swati Gupta for her help in reading and revising the thesis.

I am sincerely thankful to Dr. Maninder Singh Head, CSED and Dr. Ashutosh Mishra, PG Coordinator for their constant motivation. I would be failing in gratitude if I do not express my acknowledgement to Dr. S. S. Bhatia, Dean of Academic Affairs for making labs available and with good internet facility which made access to almost everything quite easily.

Finally, I would like to express my sincere and deep gratitude to my parents and family member for their love, encouragement, care and cooperation in helping to complete the thesis.

Simran Setia

Abstract

Cloud Computing has created a lot of buzz in the IT industry. The idea of utility based computing has led to a paradigm shift. This is due to the availability of dynamically available hardware components and freely accessible software tools at a very low cost. The beauty of cloud computing lies in its scalability. It allows its users to scale the computing resources whenever they find it necessary. Cloud Computing works on the idea of subscription based model. Users simply have to pay on pay per use basis. With no in-house management requirements for server, cloud computing helps to save substantial capital costs in any kind of business. Apart from being a cost efficient method for computing, every cloud offers unlimited storage to its users. So the users need not worry about the maintenance of physical storage devices. It is much easier to recover the data on a cloud as compare to physical storage devices. Therefore, Cloud computing has proved to be an asset to IT industry. large scale business, science and engineering applications provide a glimpse of cloud computing has to offer to its users. However, these large scale applications do present a lot of challenges to be dealt with. One of them is interference among various tasks scheduled on remote resources. The problem of interference among C.P.U. intensive, network intensive and memory intensive tasks can significantly degrade the performance of a cloud computing application. Here, we present a interference free scheduling algorithm of various tasks aiming for better performance of cloud computing application.

Contents

List of Figures

Chapter 1
Introduction

As evident from the commercial cloud platforms, Cloud Computing in a short span of time has gained in popularity as well as acceptance. The exponential increase in the data has led to wide popularity of Cloud Computing paradigm as the users find it difficult to store such large amount of information on physical servers. Cloud Computing provides the perfect solution to all such problems by virtualising the physical resources and let a third party take care of the information residing on a remote server. However, giving third party the right to take care of your information gives rise to security issues too. Apart from the issues faced by Cloud Computing users, there are many advantages that persuade users to use Cloud Computing platform to store the huge data. Cloud Computing enables on demand network access of computing resources like networks, storage, applications that can be provisioned on pay per use basis with no management overhead of remote resources provisioned. Cloud Computing is sum total of various technologies like Service Oriented Architecture and Virtualisation which satisfy the computing needs of the users while their data is stored on remote servers. In Service oriented architecture, various software and hardware services are provided to the users over a network. Virtualisation of Network, Servers, Storage, Data and Application has been made possible with the advent of Cloud Computing. However, there is a need to integrate the legacy applications with the cloud. For this, a number of integration tools are available in the market. The virtualisation of resources has been technologically made possible by Virtual Machine Manager, which separates virtualised environment from physical environment.

The increased use of cloud computing services has pressed the need for better performance of various cloud computing applications. The leading cloud service providers like Amazon Web Services, Microsoft Azure etc. provide a number of services like computation, storage to tons of applications. However, scheduling of tasks on different Virtual Machines has always been a problem because Virtual Machines have to accomodate diverse set of tasks in order to have fast execution and better resource utilisation[12]. As tasks may require to use same set of resources, it leads to interference among the tasks which follows poor performance

of the application . To overcome the issue of interference among the tasks, the intent of various sets of tasks must be noted.The classification can be done using various classification algorithms like Naive Bayes, Decision Tree, Fuzzy classification. Die to high accuracy of Random Forest, it has been used to classify the incoming tasks. Random Forest Classifier works as follows[13]:

- Random Forest Classifier begins by constructing a number of decision trees during the training phase.
- While in the testing phase,it outputs the class that is the mode of classes output by individual trees.

After distinguishing the tasks on the basis of their intent, we should make sure not to schedule tasks belonging to the same class on the same Virtual Machine. This would avert the problem of interference among the tasks. In this thesis, Random Forest Classifier has been used to classify the tasks. After the classification of tasks among C.P.U. intensive, Network intensive and Memory intensive, these are scheduled on different Virtual Machines so as to make sure an Interference free execution of application.

1.1 Introduction to Cloud Computing

Cloud Computing is the practice of delivering on demand computing resources based on the Internet[14]. Cloud Computing tend to share same set of resources by using a network of remote servers hosted by third party service provider. The services provided by Cloud computing are delivered over the internet, on pay-per-use basis. Applications and services are accessed via the Web and not via the physical storage devices. There are number of characteristics exhibited by Cloud Computing. Some of them are:

- *On Demand Self Service-* Any user can provision the services provided by the cloud service provider without actual human interaction. The services include storage, memory, network bandwidth and other platforms for the execution of cloud applications.

- *Resource Pooling*- In Cloud computing, a number of physical resources are pooled among various users. These resources are dynamically assigned or reassigned to different users according to demand of a particular user.

- *Broad Network Access*- Cloud services and applications can be accessed over the internet on a variety of platforms like laptops, tablets and smartphones.

- *Measured Service*- Cloud services are provided on pay-per-use business model. Resources are provisioned to the users on demand. The services and resources provisioned can be easily monitored and controlled by service provider and the user, imparting transparency to this technology.

- *Rapid Elasticity*- Elasticity refers to the degree of scalability provided by service provider. The dynamic provisioning and de-provisioning of services provide scalable solutions to the users. At the time of high workload, resources can be scaled in or out with minimal management.

1.1.1 Evolution Of Cloud Computing

Cloud computing has emerged as a result of the evolution of some of the existing technologies. The era of cloud computing started way back in 1999 with the advent of Salesforce.com. But cloud computing gained popularity with Amazon Web Services in 2002. Amazon's Elastic Compute Cloud (EC2) provides secure and scalable cloud computing environment for masses. The integration of Elastic Compute Cloud with Simple Storage Service (S3) makes it even more promising. Simple Storage Service allows the user to store any amount of information and retrieve it at any point of time. A series of developments took place before this era of cloud computing began. Some of those developments are:

- *Mainframe Computing*- Mainframe computers are used for bulk computations like management and calculation of census data. Modern mainframe computers emphasise on high throughput based on high volume input data. While both mainframe computing and cloud computing run on client server model, still there are certain points on which these both differ . In mainframe computing, user has total control over his data while in cloud computing third party service provider is given access for user's data.

- *Client-Server Model-* A client-server system is a networked and distributed computing comprising of client and server. A server can be viewed as a centralised entity to which every client passes a query. A client can be seen as basic entity or set of entities that share server. A server performs majority of the management tasks and a client only handles the interface between server and client. Cloud computing and client-server model are similar as cloud computing also involves a centralised data storage entity. Hence we can say that cloud computing uses client-server model to deliver services to its users.

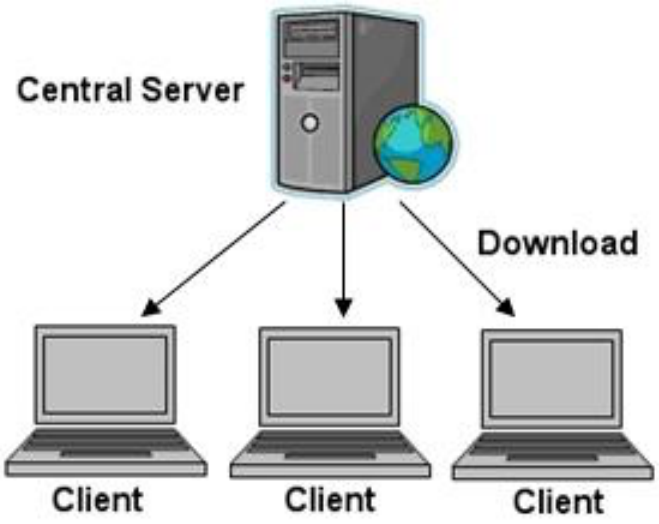


Figure 1.1: Client-Server Model[1]

- *Grid Computing-* Grid Computing is termed as a distributed computing in which various computer resources at different locations work together to achieve common goal. It is a special type of parallel computing comprising of network of loosely coupled clusters. Both cloud computing and grid computing are believed to be scalable. Grid computing is used to manage a number of short-term jobs while cloud computing is used to manage long term jobs. Cloud computing works on pay-per use basis while there's no such thing in grid computing.

- *Utility Computing-* Utility computing is metered services in which every customer is charged according to the specific usage of available computer resources. Cloud and Utility Computing work on pay-per use basis[15].

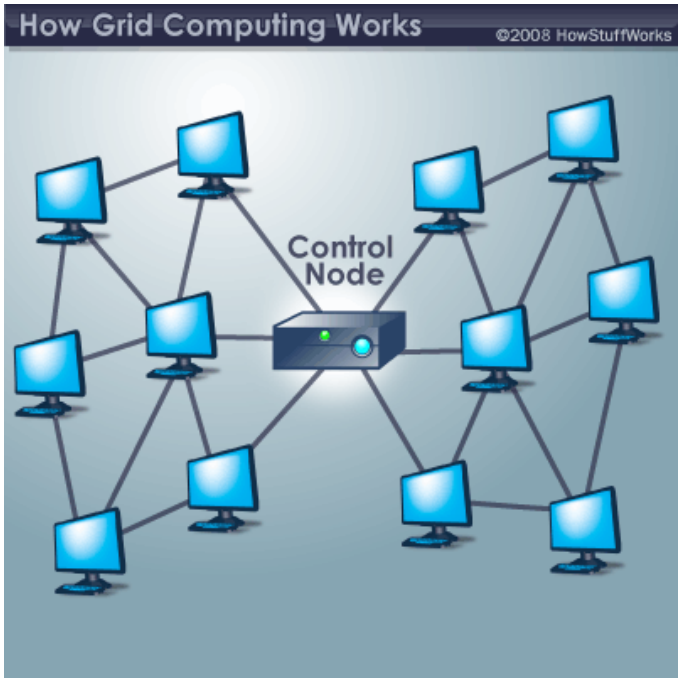


Figure 1.2: Grid Computing[2]

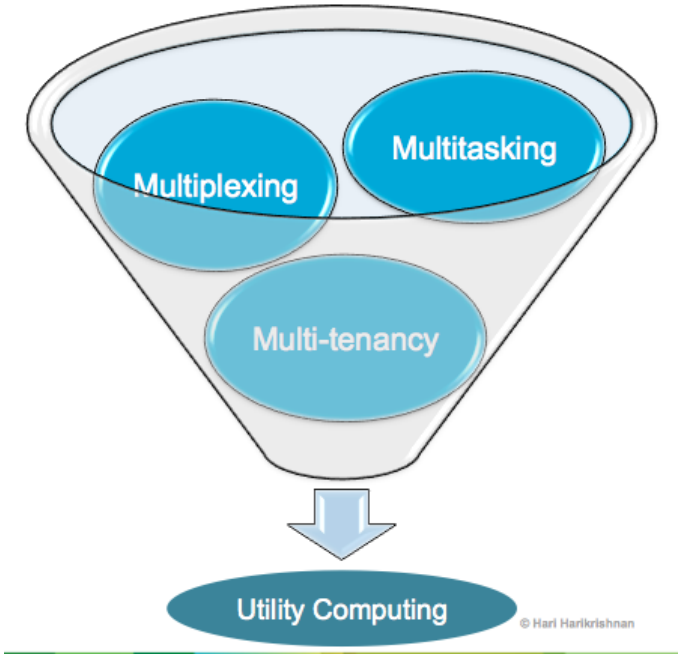


Figure 1.3: Utility Computing[3]

- *Autonomous Computing*- Autonomic Computing comprises of self-management characteristics of available networked computing resources[4]. Self-management characteristics make it even more robust to unpredictable changes. Cloud computing uses autonomic systems in order to achieve robust organisation providing its customers economies of scale.

Figure 1.4: Autonomous Computing[4]

1.1.2 Service Models

Cloud Computing advocates service-oriented architecture which states everything should be provided as a service. There are three standard service models in Cloud[14]:

- *Infrastructure as a Service*- The users are provided with computing infrastructure to realise the execution of their application. This may include storage, networks and other primary resources required to deploy and execute the cloud application. A hypervisor or Virtual Machine Manager is required which runs virtual machines as guests. Pool of resources are rapidly provisioned up and down according to the demand of a particular user. The user does not have to manage the underlying resources because that is being taken care of by the cloud service provider. However, the user has control

over the functioning of each resource provisioned. The examples include Amazon Web Services, Microsoft Azure, Google Compute Engine.

- *Platform as a Service*- In PaaS models, cloud service providers deliver web server, database or execution platform for the deployment of various applications. The users do not have to manage and control the underlying hardware and software. Software developers can easily develop software solutions without having to scale up and down every time in case of change of application's workload. The examples include Google App Engine. There are many advantages associated with PaaS models. Users can work on the same application even after being at geographically different areas. It also cuts application costs and coding time by providing the users with some already coded applications.

- *Software as a Service*- In SaaS model, users are provided with actual application software. Users do not manage the underlying infrastructure or the platform on which the software runs. Cloud service providers install the application on their physical server. So the maintenance of the application software is the responsibility of the service provider. The examples include Google Apps, Salesforce, Concur. The benefits of SaaS models include lower maintenance costs of the software, no scalability issues and no installation overhead. As the software is installed on cloud service provider, user does not have to care about the physical storage needed for the application and its subsequent upgrades.

1.1.3 Deployment Models

Cloud Service models are deployed in an environment which is mainly distinguished by proprietorship. The deployment model chosen depends solely on the requirements and structure of a particular organisation. In order to know which deployment model will be suitable, it is necessary to go through basic four deployment models[14]:

- *Public Cloud*- A public cloud is based on the delivery of cloud services to the general public in a uniform way without distinguishing over the control

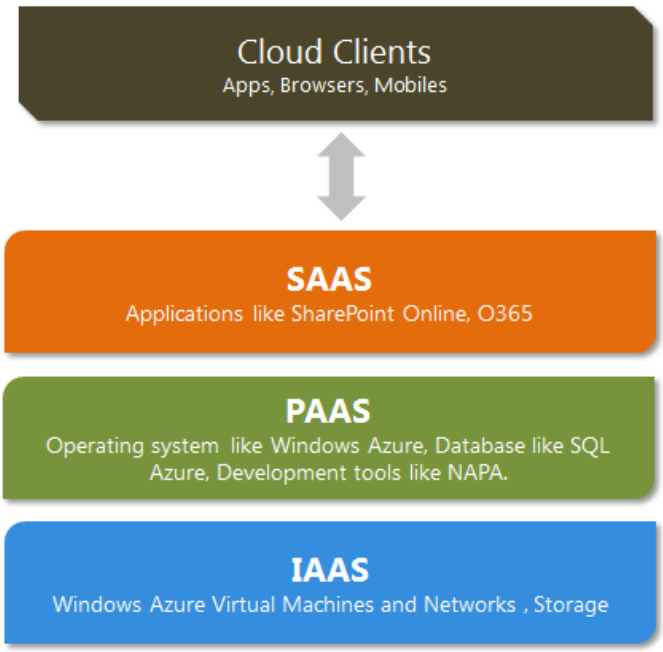


Figure 1.5: Cloud Service Models[5]

of applications. Cloud Service providers are responsible for imparting transparency in the system. All the users are provided with uniform services. No user is given special control of the underlying hardware or infrastructure. It is best suited for the application that is used by many users. As the cloud applications work on pay per use business model, the cost is shared by a number of users which makes it an economical deployment technique.

- *Private Cloud*- A private cloud is technically same as public cloud except the fact that the security measures are taken in private cloud to control the access to the underlying infrastructure or software. Every private cloud is implemented in cloud-based secure environment. It is not open to general public and belongs to particular organisation. According to the security measures implemented, every user is authorised to use particular set of applications. Before using any application, each user is authenticated and then allowed access to the application software. Organisations which run on time critical applications are best suited to apply private cloud as their deployment model.

- *Hybrid Cloud*- Sometimes, a suitable environment for a workload is provided partially by public cloud and partially by private cloud. In that case, hybrid cloud serves the organisation in the best possible way. It can be customised to a great extent by a user. Every Virtual Machine hosted is connected to either public or private cloud according to the choice of the owner of Hybrid Cloud. As we know cloud is a pool of multiple resources, the resources that are used by every cloud user in the organisation can be housed in public cloud while some critical resources which contains some sensitive data can be housed in private cloud. Hybrid cloud is way more flexible than any public or private cloud but one has to overlook other challenges like interface design for every application or high capital requirements.

- *Community Cloud*- A community cloud is built for targeted group of individuals which share common interests. It is best suitable for specific organisations which are working on a common project or for organisations which require centralised management of their joint ventures. All the users of community cloud share similar kind of interests and security concerns. As it is hosted by specific organisations, the cost is divided among them which makes it economical.

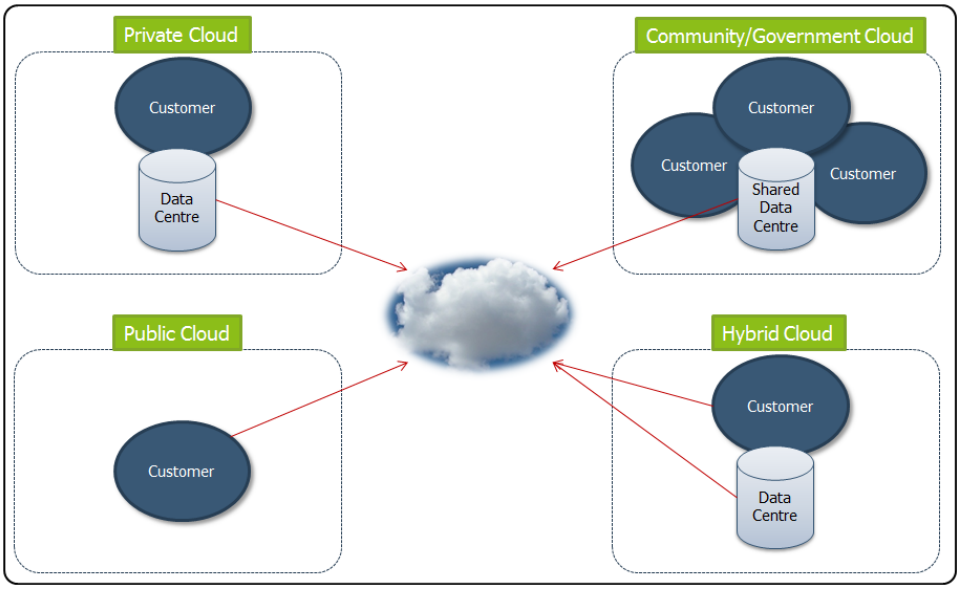


Figure 1.6: Cloud Deployment Models[6]

1.1.4 Virtualisation in Cloud Computing

Virtualisation is a kind of technology that enables to create multiple virtual instances of a single physical instance[16]. The instances may include any hardware platform, network devices or storage devices. It has led to a revolution in the area of Information Technology by reducing the hardware and software costs. As a single instance can be shared among several instances, it has provided economies of scale to many industries. Therefore, Virtualisation imparts flexibility to the computer hardware increasing its utilisation and efficiency.
Hypervisor, also known as Virtual Machine Manager, helps us to realise the concept of virtualisation. It acts as a middleware between Guest Operating System and System Hardware. The machine on which virtual machine runs is the host machine and the virtual machine is referred to as the guest machine which share the physical instance of underlying machine. However, every virtual machine is supposed to be independent and do not interfere with each other. There are basically two types of Hypervisor:

- *Type 1 Hypervisor*- These are often referred to as native or bare metal as they run directly on the system hardware without any operating system in between. Examples of Type 1 hypervisor are VMware ESXi, Citrix XenServer.

- *Type 2 Hypervisor*- Type 2 Hypervisor acts as a software within operating system of physical machine. As we can see from Figure 1.8., operating system runs as a third layer above the hardware and hypervisor runs as a second layer. Examples include Microsoft Hyper V, VMWare workstation.

The different types of virtualisation that can be implemented are:

- *Hardware Virtualisation*- Hardware Virtualisation is the virtualisation of hardware platforms required to run the operating systems or any other application. It hides the various characteristics of hardware platforms and present a abstract view instead. As to realise virtualisation, hypervisor or vIrtual machine manager is needed. Hardware Virtualisation also installs a hypervisor which acts as an intermediate layer between software and underlying hardware. Hardware Virtualisation is of following types:

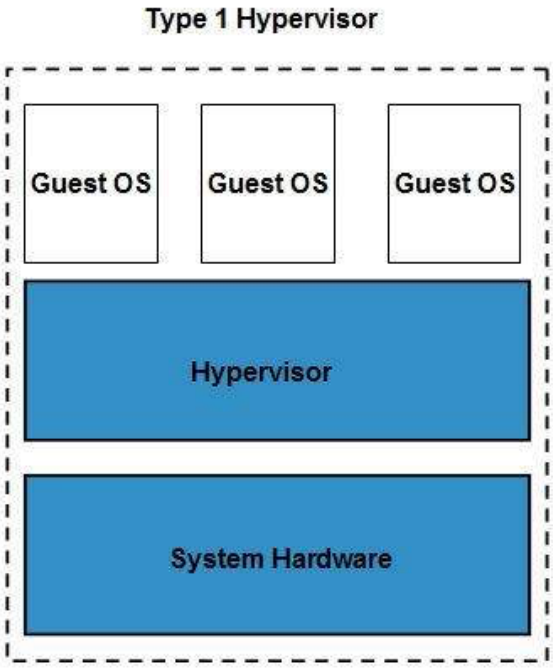


Figure 1.7: Type 1 Hypervisor[7]

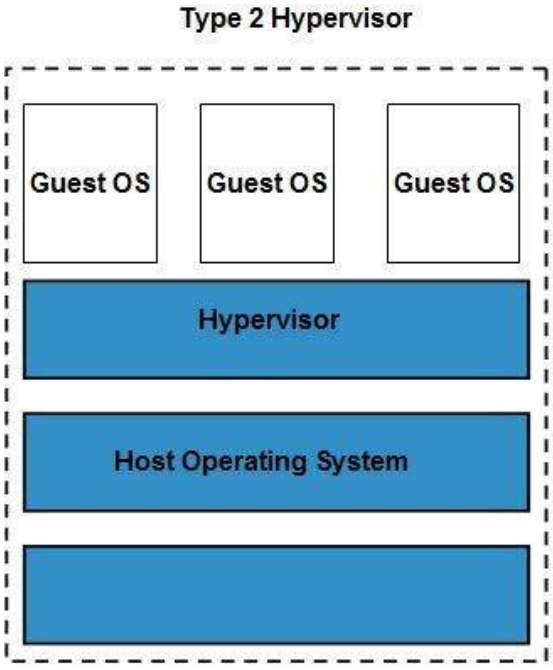


Figure 1.8: Type 2 Hypervisor[7]

– *Full Virtualisation*- In Full Virtualisation, guest software runs on unmodified host. The guest software is unaware of the underlying hardware. It is called so as the guest software is completely independent of the underlying hardware. It can prove to be very helpful if we want to combine existing systems and newer systems. However, if we need a slight variation in the underlying system, this technique does not prove to be useful.

– *Paravirtualisation*- If we want some changes in the underlying hardware, then paravirtualisation allows for an interface that is different from the underlying hardware. The advantage of using paravirtualisation is that we can run modified softwares on the underlying hardware. This allows us to run multiple operating systems on single hardware.

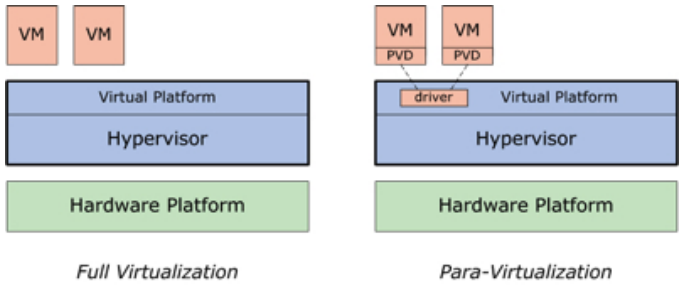


Figure 1.9: Full Virtualisation vs Paravirtualisation[8]

- *Application Virtualisation*- An application is installed on a remote server, instead of the hard drive of local server. The ability to use the hard drive of a remote centralised server enhances the storage capabilities of each server using the virtualised application.

- *Operating System Virtualisation*- OS Virtualisation allows running different operating systems or several instances of same operating system on a single machine. It saves the overhead of multiple hardware platforms for very OS the user needs.

- *Network Virtualisation*- It combines various network resources and transform them into a single software entity. It involves virtual management of various IPs through Network Interface Cards or routing tables.

- *Storage Virtualisation*- This is the most common type of virtualisation used. It has led to negligible storage costs. The user does not know exactly where the data is but he can access it whenever he needs it. The third party service provider is responsible for management and recovery of data in case of data loss.

1.2 Task Scheduling Algorithms on Cloud

There are many task scheduling algorithms predefined in cloud computing environment. The task scheduling algorithm chosen affects the performance and throughput of the cloud application. So it becomes necessary to choose the right task scheduling algorithm according to a particular application. As stated by Lakshay Malhotra et al.[17], the task scheduling algorithms can be classified into two main groups:

- *Batch Mode Heuristic Scheduling Algorithms*- This is a static approach in scheduling tasks. Tasks are collected over a period of time. After collection of tasks, scheduling algorithm starts scheduling the tasks according to a definite policy.

- *Online Mode Heuristic Algorithms*- It follows a dynamic approach for scheduling of tasks. Tasks are scheduled as they arrive in the system. This scheduling algorithm may not be as good as batch mode heuristic algorithms but it is the most appropriate way to schedule the tasks.

1.2.1 Batch Mode Heuristic Scheduling Algorithms

Batch Mode Heuristic Scheduling Algorithms start after a fixed period of time when all the tasks are collected and queued. Some of batch mode heuristic scheduling algorithms are:

- *First Come First Serve*- Tasks are scheduled as they arrive in the cloud application. It has high response time but may lead to starvation in case any of the tasks has high running time.

- *Shortest Job First-* According to the running time of various tasks collected, the task with minimum running time is scheduled first. Cloud application with shortest job first has high throughput but it may lead to starvation for longer duration tasks.

- *Round Robin-* In Round Robin Scheduling Algorithm, tasks are scheduled according to FIFO manner but are given limited period of time to execute called time quantum. After time quantum of each task expires, then next task in the queue is scheduled. This approach has high response time.

- *Min-Min Algorithm-* In Min-Min Algorithm, prerequisite is the completion time of every task. Among all the tasks, the task with minimum completion time is chosen and added to meta-task group. The task with minimum completion time among all the tasks in meta-task group is scheduled next on any given Virtual Machine.

- *Max-Min Algorithm-* Max-Min Algorithm is analogous to Min-Min Algorithm. In Max-Min Algorithm, the task with maximum completion time is chosen from meta-task group.

1.2.2 Online Mode Heuristic Scheduling Algorithms

Most-fit task scheduling algorithm is one of the examples of online mode heuristic scheduling algorithm.

- *Most-fit Task Scheduling Algorithm-* The most fit task among all the tasks in the ready queue is scheduled on the virtual machine first. The most-fit task is chosen according to some parameters.

1.2.3 Other Scheduling Algorithms

- *Resource Aware Scheduling Algorithm-* According to Pinal Salot[18], in Resource Aware Scheduling Algorithm two types of resources are identified. One is the resources required by each task and the other one is the resources present with each virtual machine. The number of tasks that can be run on each Virtual Machine can be inferred from the number of resources present

with each Virtual Machine. The data necessary for Resource Aware Scheduling Algorithm can be collected from the execution traces of each task. The dependencies observed from each task can be used to find out the resources required by each possible task.

- *Reliable Scheduling Distributed in Cloud Computing*- As proposed by Pinal Salot[18], scheduling in distributed system leads to balancing of workload among the various virtual machines. In Reliable Scheduling Distributed, each task is further divided into subtasks. Each of the subtask is then scheduled on any given Virtual Machine according to request and acknowledge time of that particular task. It proves to be a very good scheduling technique in case of load balancing.

- *Priority Based Service Scheduling*- In Priority Based Service Scheduling, priority is calculated according to tolerable delay and service cost [18]. The tasks available are assigned to each of the queues. These queues are admitted to each of the virtual machine according to given priority. This scheme guarantees high QoS.

- *Cost-Based Scheduling Algorithm*- Vijayalakshmi A. Lepakshi et al.[17] have proposed an algorithm which strives to reduce the communication costs by calculating the resource costs and computation cost of each task. A group of tasks is identified by constructing a Directed Acyclic Graph (DAG). A DAG is constructed to minimise the communication between the tasks. A particular group of tasks with minimum communication requirements is scheduled on a Virtual Machine according to given resource requirements.

1.3 Introduction to CloudSim

CloudSim is a framework for modelling and simulation of cloud computing applications and services. It is very beneficial for cloud computing users to test their applications and services in an environment similar to that of cloud in an efficient and controllable manner[9]. CloudSim helps in creation of multiple hosts with each host comprising of multiple Virtual Machines. CloudSim platform provides mod-

elling of Cloudlets, Datacenter brokers and Virtual Machines at the same place. Separate modules are also available for Virtual Machine and cloudlet scheduling. It even provides the flexibility to switch between space-shared and time-shared scheduling policies.

1.3.1 CloudSim Architecture

The main components present in CloudSim Toolkit are:

- *Cloudlet*- A cloudlet is a task assigned by the user to the cloud application. The information encapsulated in cloudlet comprises of cloudlet id, cloudlet length, cloudlet file size and cloudlet output file size.
- *Data Centres*- Data centres are an amalgamation of data storage facilities, communication facilities and servers. The infrastructure services provided by cloud service provider are encapsulated in a data centre.
- *Regions*- The geographical regions are called Regions. These are the geographical regions where cloud service providers provide cloud services. Six Regions correspond to six continents in the world.
- *Hosts*- It models the virtualised instances of physical resources such as network, memory.
- *Service Broker*- Service Broker is responsible for allocating data centre to serve particular user or group of users.
- *Virtual Machine Manager Allocation Policy*- It is responsible for allocating Virtual Machines to hosts.
- *Virtual Machine Scheduler*- It models scheduling policy for allocating processor cores to Virtual Machines.

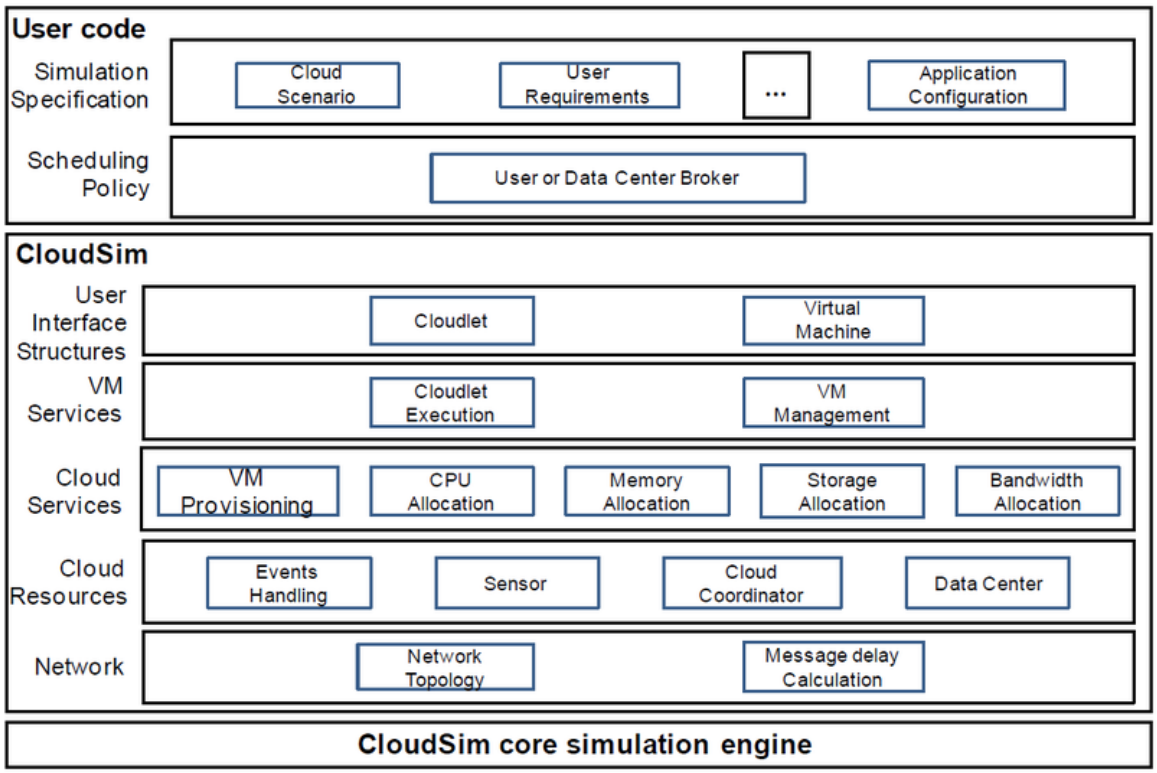


Figure 1.10: CloudSim Architecture[9]

Chapter 2
Literature Review

Cloud application is a cloud-based application. A cloud application is managed by third party service provider. Cloud Applications provide a number of benefits to its users. Cloud Applications are believed to be scalable. Third party service provider can scale up or down the services according to user requirements. Cloud applications do not need a physical server by virtue of virtualisation. User can download the application from anywhere. This means a cloud application does not consume any space in the physical hard-drive. In a cloud, a datacenter can be shared by a number of cloud applications leading to low costs. keeping in mind the above stated benefits, cloud application has become a major breakthrough in this technological era. There are a number of cloud applications which we use in our daily life like DropBox, Skype, GoogleDocs, MicrosoftOfficeLive. All these services can be delivered to the end user through Software As A Service(SaaS). Apart from SaaS, Platform As A Service(PaaS) and Infrastructure As A Service(IaaS) applications like Google App Engine, Amazon EC2 etc are available for the users. The performance of a cloud application depends on a number of factors. Ideally, a task must not be affected by co-located tasks. But It can be adversely affected by the interference caused by co-located tasks on the same virtual machine. Recently, various methodologies have been proposed to limit the effect of interference in data-intensive applications. Some of the proposed methodologies are:

2.1 Task And Resource Allocation Control(TRACON) Framework

Tasks being scheduled on the same virtual machine experience interference from co-located tasks. So it becomes necessary to design a scheduling algorithm which will implement an interference free policy. TRACON is a novel Task and Resource Allocation Control Framework proposed by R.C. Chiang et al.[19]. It comprises of three major components:

- *Interference Prediction Model*- It uses statistical Machine Learning techniques to infer the performance of application observed from different Virtual Machines.

- *Interference Aware Scheduler*- It utilises the Interference Prediction Model for effective resource management.

- *Resource Monitor*- It monitors performance of the application at run-time.

It uses machine learning models to predict the interference on a Virtual Machine. It represents a dynamic model which tends to adapt according to the application's performance. It is primarily designed for data-intensive cloud applications.It only considers the concurrent data-intensive applications to curb the effect of interference. TRACON has resulted in significant performance improvements leading to better resource utilisation. It aims to improve both the application performance and Resource Utilisation for the virtualised Data-centres. But it only takes into account the effect of CPU and I/O utilisation on various data-intensive applications. As cloud applications run on a network, it must take into account the tasks that are more of network oriented like e-mail.

2.2 QoS Aware Clouds(Q-Clouds)

An alternative approach to handle the problem of interference among tasks was proposed by Nathuji et al.[20]. It advocates the use of QoS-aware clouds. QoS-aware clouds present a dynamic approach to overcome the challenges imposed by interference. Q-Cloud servers allocate resources to various applications according to workload Service Level Agreement which is a contract between a service provider and end user. Resource partitioning plays an important role in the implementation of Q-clouds. Page colouring can be used for resource partitioning. But page colouring is a NP complete problem. However, due to the running complexity of page colouring, this approach does not seem to be a promising one. If each task has a dedicated Virtual Machine, then there would be no interference among the tasks. However, the point of developing platform like cloud computing would be lost. The advantages of cloud platform is resource sharing which leads to

scalable solutions. So a dedicated Virtual Machine is not the solution to deal with interference. Q-Clouds offer performance oriented solutions to curb interference. The main components of a Q-Cloud are as follows:

- *Cloud Scheduler*- Cloud Scheduler determines the placement of tasks submitted by customers on cloud servers. The decision of scheduling tasks on different Virtual Machines is based on resource requirements of various tasks.

- *Interference Mitigation Control*- Interference Mitigation Control helps in bringing the performance of mitigated Virtual Machines to the same level as that of other Virtual Machines that experience no interference.

- *Resource Efficiency Control*- The main function of Resource Efficiency Control is to define new achievable QoS levels when previous levels have been achieved.

2.3 iAware Scheduler

There are basically two types of performance interference that must be overlooked, one on the migration source server and other on the destination server due to Virtual Machine migration as stated by F.Xu et al.[21]. It has been observed that on the destination server,migrated Virtual Machines and other running Virtual Machines undergo performance degradation due to migration interference. This might be due to mismatch between the demand of resources by co-located Virtual Machines and supply of shared resources. An estimation model, iAware has been designed based on various key factors like estimated network interference, estimated C.P.U. utilisation, Migration time. So basically there are two types of interference that must be taken care of. These are:

- *Co-location Interference*- This kind of interference is only observed among the tasks scheduled on the same Virtual Machine.

- *Migration interference*- Due to load balancing, some of the Virtual Machines migrate to different Physical Machines. So there may be an interference among the Virtual Machines hosted on same Physical Machine.

While Co-location Interference is believed to be a function of network I/O interference, C.P.U. interference and memory interference. A demand and supply model is realised based on the above assumptions. Migration interference can also be calculated using this model. Ratio of resource requirements and demands is calculated based on the interrupts received by a particular Virtual Machine hosted by a Physical Machine. A high ratio indicates the demand is more than the supply and the system is not performing well. This kind of model may also be considered in case of Virtual Machine migration. Demand-Supply ratio will be calculated on new Physical Machine in that case.

2.4 MIMP

MIMP(Minimum Interference Maximum Productivity) is a deadline and interference aware scheduler implemented in Hadoop given by Wei Zhang et al.[22]. It enhances the performance of Virtual Machine Manager's scheduler as well as the performance of hadoop job scheduler. It is designed to run on hybrid cluster which is a combination of virtual and physical nodes. It associates priority with each kind of task. Interactive web applications have high priority and hadoop tasks have lower priority as compare to other tasks. MIMP is believed to have two components:

- *Minimal Interference C.P.U. Scheduler*- It allocates C.P.U. scheduler to tasks according to their priority. It prevents lower priority tasks like hadoop tasks to get scheduled when scheduler is allocated to some higher priority tasks.

- *Maximal Productivity Job Scheduler*- It takes into account deadline aware scheduling. Its main function is to assign the available resources to available tasks such that all the tasks get completed before their respective deadline.

The new scheduling algorithm defined in MIMP is enhanced version of EDF(Earliest Deadline First). In EDF, deadline of every task is a prerequisite. The task with minimum deadline limit is scheduled first, followed by the next task with next minimum deadline limit. The new scheduling algorithm designed in MIMP takes into

account the progress of each task along with the deadline. This algorithm makes sure that the given task meets its deadline requirements along with maximum progress without causing the data nodes in hadoop to get overloaded.

2.5 Paragon

Paragon Scheduler designed by C. Delimitrou et al.[12], is a scalable scheduler which can be used for large data-centres. Thousands of applications run on large data-centres. The workload to be scheduled on different Virtual Machines is unknown to the scheduler. It uses analytical techniques like collaborative filtering to classify an incoming application. The classification of incoming workload may help to schedule the applications in such a manner such that there is no interference among the workload scheduled. While scheduling following points must be taken care of:

- *Hardware Heterogeneity*- In a data-center physical servers are periodically changed due to wear and tear. They must be replaced for the better performance of the application. So at any point, a data-center may host many hardware configurations of a physical server at same point of time.

- *Colocation Interference*- Interference arises due to scheduling of multiple workloads on the same physical server. It may deteriorate the performance of the application significantly.

2.5.1 Collaborative Filtering

Collaborative Filtering has been used to classify the incoming workload in Paragon. It is a popular technique already publicised by Netflix. Recommender systems widely use Collaborative Filtering to recommend new products to the users.
Collaborative Filtering uses Singular Value Decomposition(SVD) and PQ reconstruction. The input is a sparse matrix with rows as users and columns as movies. Each value in the matrix represents rating given by the user to a movie. The challenge is to calculate the other values in the sparse matrix so that new movies can be recommended to existing users.

In Singular Value Decomposition, a $m\times$n matrix is decomposed to U and V matrices. Every $m\times$n matrix is factorised to $U\sum$V* where U is a $m\times$m unitary matrix, Σ is a $m\times$n matrix having non-negative real numbers on its diagonal, V is a $n\times$n unitary matrix. For PQ reconstruction,

- Q=U
- $\mathbf{P}^\top = \sum.\mathbf{V}^\top$

2.5.2 Server Selection

After successful classification of the incoming workload, the major challenge is to select the right server for scheduling the workload. If we make a greedy selection and choose the server that finishes the given workload in minimum amount of time, then it may lead to interference among the co-located workload. However, if we schedule the workload on a different server keeping in mind the interference among the co-located workload, then we may end up scheduling the workload on a server that generates less throughput and degrades the performance of the application significantly. Therefore, keeping in mind the trade-off between performance degradation and interference, the scheduling of workload must be done carefully. There are two methods proposed for candidate servers. They are:

- *Greedy Server Selection*- It is a fast and easy method that searches for the optimal server after the examination of every server state.
- *Statistical Framework for Server Selection*- Greedy Server Selection may not be optimal for large data-centres comprising of thousands of servers. It uses cryptographic hash functions and samples a number of servers. Instead of examining every server, only few servers are examined.

2.6 Classification

Classification is the process of assigning each instance a class label present in the given data. First of all, the proposed classification algorithm tries to discover the relationships among the attributes to build the classifier model. A target

attribute is defined during this phase. After the classifier model is built, it is used to classify the instances to particular class according to target attribute class label. Classification process in data mining can be divided into two phases:

- *Training phase*- During this phase, a classifier model is built with the help of training data according to their class labels. Data can be trained through a number of ways like cross-validation, percentage split etc.
- *Testing phase*- Once training of the classifier model is done, it is used to classify the remaining instances to their respective classes.

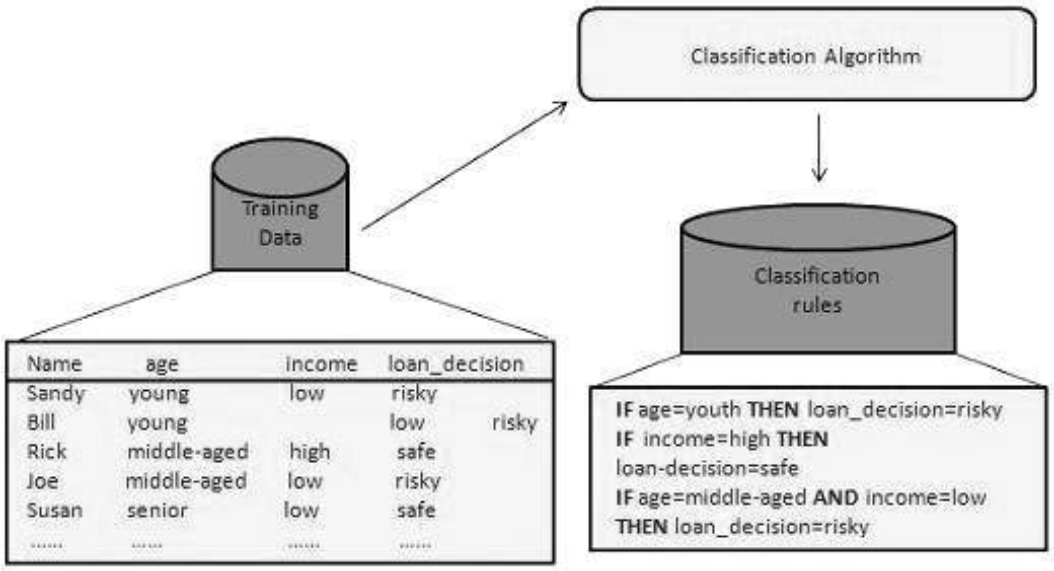


Figure 2.1: Training Phase[10]

There are a number of classification algorithms available. Random Forest has proved to be an effective way of classifying the available data with high accuracy.

2.6.1 Random Forest Classification

As stated by A Liaw and M Wiener, Random Forest works as an ensemble method. It starts with the known classification technique of Decision Tree to predict class labels. In random Forest classification algorithm, a number of decision trees are constructed based on the given data. A particular instance may have been assigned different classes in different trees. This may lead to discrepancies among the resulting class labels. But instead of drawing inference from a single decision tree,

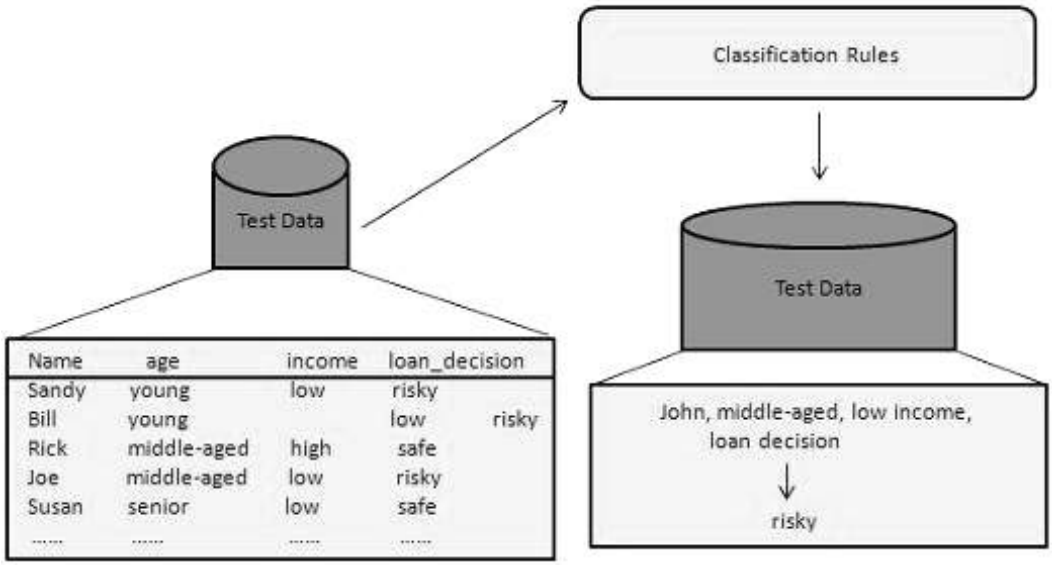


Figure 2.2: Testing Phase[10]

all the trees are used for predicting class labels. If a particular instance is classified under a particular class label by majority of the trees, then it would be classified under that particular class. This is how Random Forest has proved to be one of the most accurate classification algorithm.

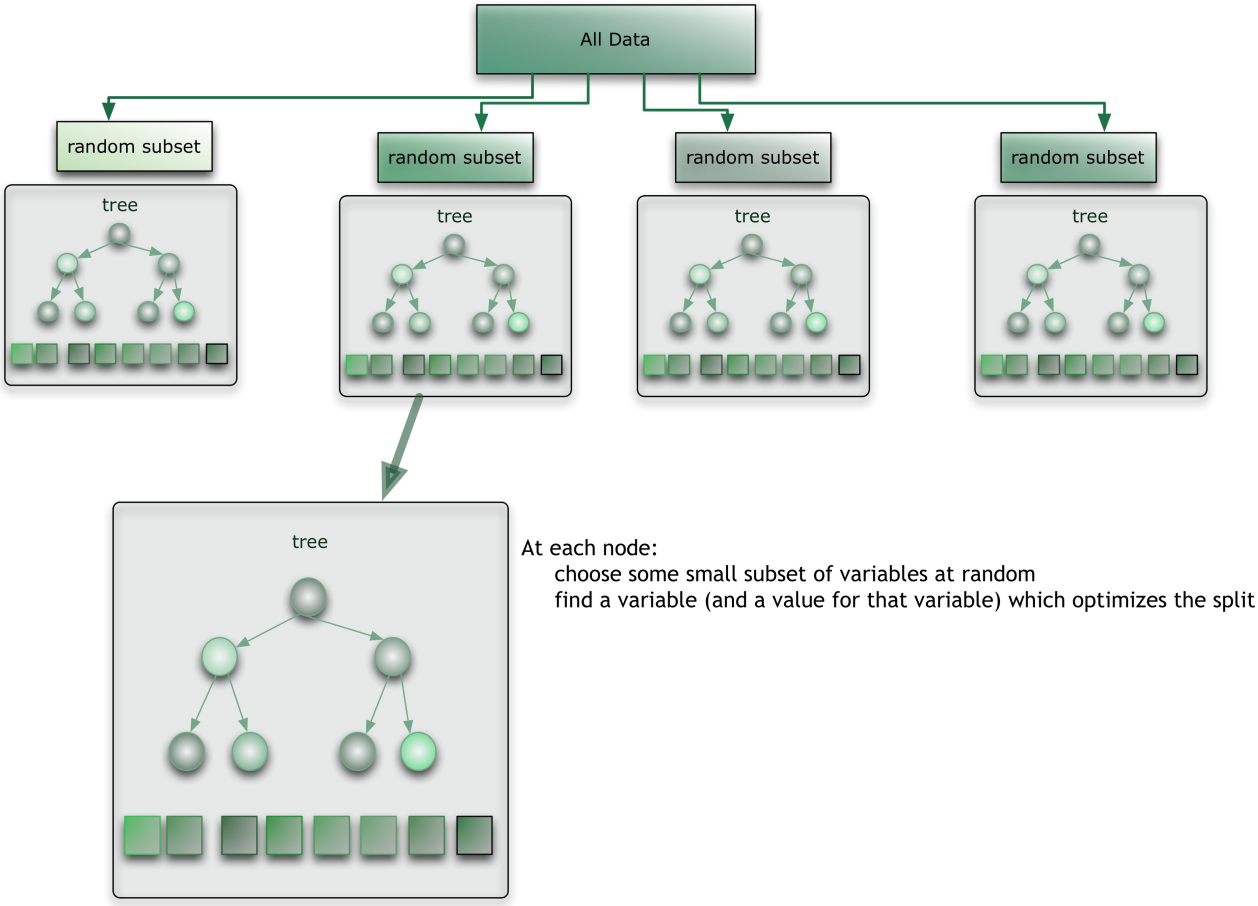


Figure 2.3: Random Forest Classification[11]

Chapter 3
Research Problem

3.1 Problem Statement

Now-a-days, from start-ups to large enterprises, every kind of business depends on the technologies provided by cloud computing in some way. Cloud computing has proved to be a game changer in the IT industry. It has also affected common people in some way. Any layman with no concern with the IT world uses a number of cloud applications like iCloud, Google drive, Skype etc. Keeping in mind the market created by cloud technologies, it has become necessary to achieve better performance with the cloud applications. Of all the challenges faced by cloud application, interference leads to significant performance issues. In this research work, we will consider the interference experienced among the co-located tasks on any Virtual Machine. First, all the incoming tasks need to be classified to their respective classes. The classes identified are:

- C.P.U. intensive tasks
- Memory intensive tasks
- Network intensive tasks

Random Forest would be used to classify the incoming tasks among the above stated classes as it is proved to be a highly accurate algorithm. Once classification of tasks is done, the challenge is to schedule the classified tasks on given Virtual Machines so as to eliminate any possibility of interference among them. Round-Robin would be used to schedule the tasks on their respective Virtual Machines. The tasks would be scheduled in a definite order on the consecutive Virtual Machines. For example, first C.P.U. intensive task would be scheduled on VM #1, first Memory intensive task would be scheduled on VM #2, first Network intensive task would be scheduled on VM #3 and this process goes on till all the tasks are scheduled. This kind of scheduling policy would ensure an interference free scheduling of tasks on Virtual Machines.

3.2 Research Gaps

A lot of research has been done to design an interference free scheduler. Various schedulers have been proposed. Some of them have recorded significant performance improvements. Some of the schedulers proposed by various researchers take account of the interference experienced among the co-located tasks and some of them take care of the interference experienced among the Virtual Machines. Till date, the other parameters like memory and network utilisation have not been considered. In this paper, we present a scheduler that will take into account the interference among the C.P.U. intensive tasks, Memory intensive tasks and Network intensive tasks on different applications. Further, the scheduling of various applications can be performed keeping in consideration the classification done earlier.

3.3 Research Objectives

In the light of above discussed problem statement, following set of objectives have been defined:

- To study various platforms available for developing a cloud application
- To study different classification techniques available to classify the incoming applications to their respective classes
- Analysis of various classification algorithms available for classification of tasks on Weka
- To classify the incoming tasks based on the best classification algorithm using Weka tool
- Execution of interference free scheduling paradigm based on the concept of Round-Robin scheduling algorithm on CloudSim
- To develop a research based interference free technique for scheduling tasks on a cloud platform using CloudSim ToolKit

3.4 Research Methodology

In this research work, CloudSim ToolKit and Weka are used to implement the interference aware scheduling. Weka is only used for classifying the incoming tasks.

- Create a workload file which consists of a list of cloudlets with the following attributes:
  - *Submission time*-The time(in ms) at which cloudlet or task is submitted to the cloud.
  - *Main Memory Required*-Main memory(in MB) required to accomplish the task.
  - *Storage*-The storage(in MB) required by each cloudlet to store data and execute successfully.
  - *Deadline*-The time(in ms) in which cloudlet should complete its execution.
- Create a data center with 200 hosts. Each host should have 4 cores of 10000 MIPS each, 32 GB RAM, 10 TB storage in CloudSim.
- Pick the data from the workload file and create cloudlets using Cloudlet class in CloudSim.
- Calculate the C.P.U. usage, memory usage and network usage of each cloudlet using getUtilizationCpu(), getUtilizationRam(), getUtilizationBw() functions respectively present in Cloudlet class
- Save the respective usages of each cloudlet in a csv file
- Use Weka to classify the above cloudlets. Train and test the data using Random Forest Classifier using Weka. Store the results in a file.
- Using the results from the file, create separate lists of cloudlets cpulist, ramlist and bwlist according to the classification done.

- Use Scheduling algorithm stated in section 4.4 in Round Robin fashion. The Scheduling Algorithm proposed would schedule in such a way so as to have interference free scheduling

Chapter 4
Implementation

In this section, the implementation details of the underlying research work would be discussed. CloudSim has been used to implement the proposed interference aware scheduling proposed.

4.1 Cloudlets

Cloudlet, in cloud computing, is regarded as a small-scale data center capable of supporting a cloud application. In CloudSim, cloudlet is a task submitted to cloud. Cloudlet class in CloudSim defines following parameters in its constructor:

- *Cloudlet Id*- It is the unique identification number associated with every cloudlet.
- *Cloudlet Length*- It defines the length of cloudlet in terms of number of instructions to be executed.
- *Number of processing elements*- In CloudSim, Pe class represents each processing element defined in terms of MIPS(Millions of Instructions per Second). pesNumber represents the number of processing elements present for completing the task.
- *Cloudlet File Size*- It represents size of input file of a task before execution on a cloud.
- *Cloudlet Output File Size*- It represents size of cloudlet file after execution on a cloud.
- *Utilization Model*- Utilization Model in CloudSim is defined for C.P.U., network and memory. It defines the percentage usage of each of C.P.U., memory and network.

The additional parameters used in this research work are Submission time, Main memory required, storage and deadline defined in section 3.4.
The foremost work is to define the parameters present in Cloudlet constructor for

each cloudlet. A workload file must be created for additional parameters. After defining all the parameters, Cloudlet class is used to create cloudlets. The user class must extend the Cloudlet class for creating cloudlets. UtilizationModelStochastic class is used to calculate C.P.U., memory and network utilization.

- *C.P.U. utilization*- It can be calculated with the help of getUtilizationModelCpu() function.
- *Memory utilization*- It can be calculated with the help of getUtilizationModelRam() function.
- *Network utilization*- It can be calculated with the help of getUtilizationModelBw() function.

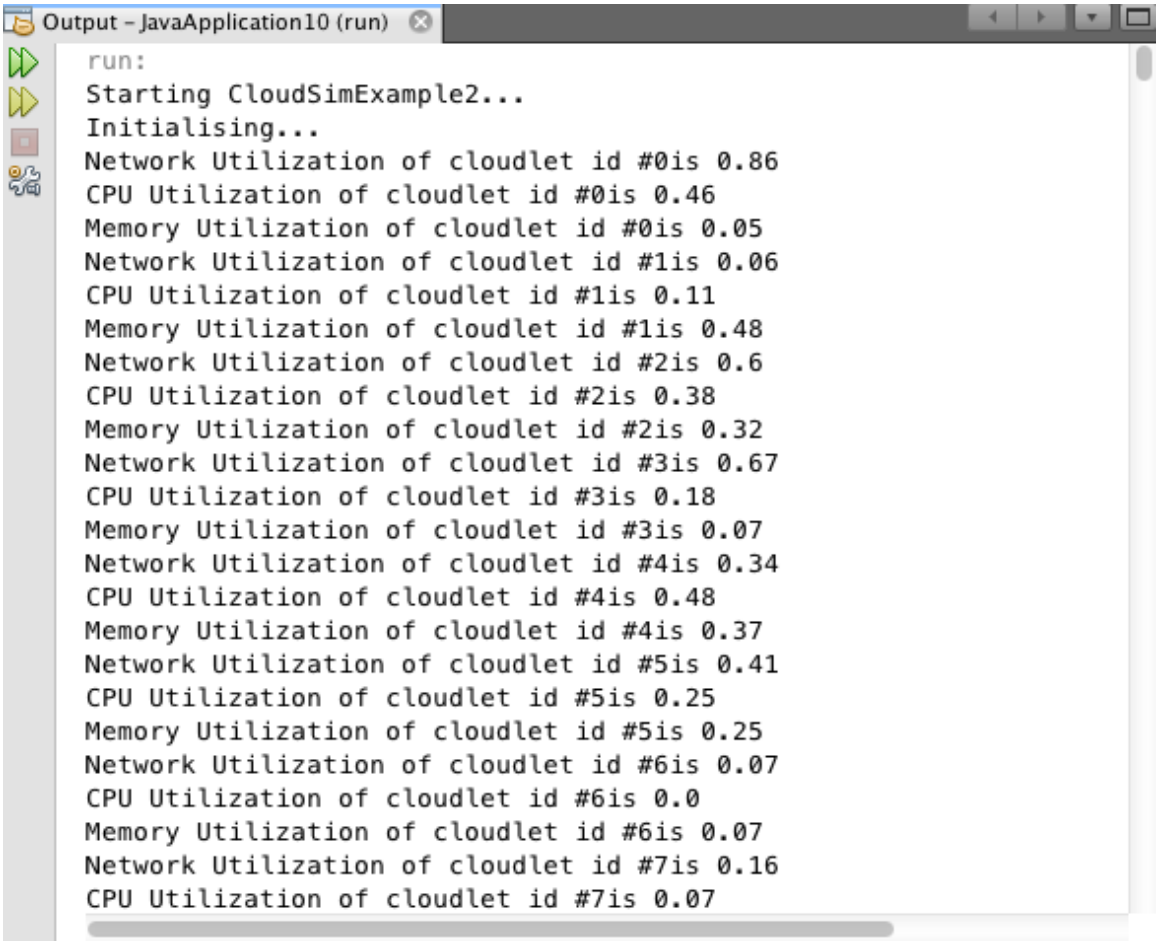


Figure 4.1: Utilization percentage of cloudlets

4.2 Datacenter and hosts

Datacenter is regarded as the actual hardware which stores all the data related to cloud application or cloud users having several virtualised machines. Datacenter class in CloudSim deals with Virtual Machine related queries. A cloud application may have a number of datacenters. Datacenter constructor defines the following parameters:

- *Name*- The identity of each datacenter.
- *Datacenter characteristics*- It defines the characteristic list of a datacenter. The following characteristics form a part of this list:
  - System Architecture
  - Operating System
  - Virtual Machine Manager
  - Host list
  - Time zone of the resource located
  - Cost of using processing in the resource
  - Cost of using memory in the resource
  - Cost of using network in the resource
- *VM Allocation Policy*- It represents the allocation policy used to assign hosts to the Virtual Machines present in the datacenter.
- *Storage List*- It defines set of storage devices used by the datacenter to store data.

A host in CloudSim hosts Virtual Machine associated to it. A number of hosts can be created in a datacenter. Each host must have 4 processing elements of 10000 MIPS each, 32 GB RAM, 10 TB storage. It can host a number of Virtual Machines. Virtual Machines can be created with the help of Vm class. Vm class defines following parameters:

- *VM Id*- Unique identification number of each Virtual Machine.

- *Broker Id*- It represents datacenter broker associated with each Virtual Machine which is responsible of assigning cloudlets to particular Virtual Machine.
- *MIPS*- MIPS(Millions of Instructions per Second) of each Virtual Machine
- *Number of Processing elements*- Number of processing elements associated with each Virtual Machine.
- *RAM*- Amount of RAM(Random Access Memory) required for Virtual Machine
- *Bandwidth*- Amount of bandwidth required for Virtual Machine in case of network applications.
- *Size*- Storage required to run Virtual Machine.
- *VMM*- Virtual Machine Manager used.
- *Cloudlet Scheduler*- It represents the scheduling policy used by the Virtual Machine. It can be classified to following types:
  - *Time Shared*- Each Virtual Machine in datacenter is assigned a set of tasks. Each task runs for a definite time quantum on the Virtual Machine followed by the next task.
  - *Space Shared*- The datacenter is divided into a number of Virtual Machines. Each Virtual Machine is allocated a task that runs to completion.

In this research work, only one datacenter is considered with 400 Virtual Machines and 200 hosts. Time Shared cloudlet scheduler is best suited for implementing a scheduling algorithm similar to Round Robin as each task in Round Robin scheduling algorithm is assigned a definite time quantum.

4.3 Classification

Random Forest has been chosen as the classification algorithm to classify the incoming tasks due to its high accuracy. A number of classification algorithms like

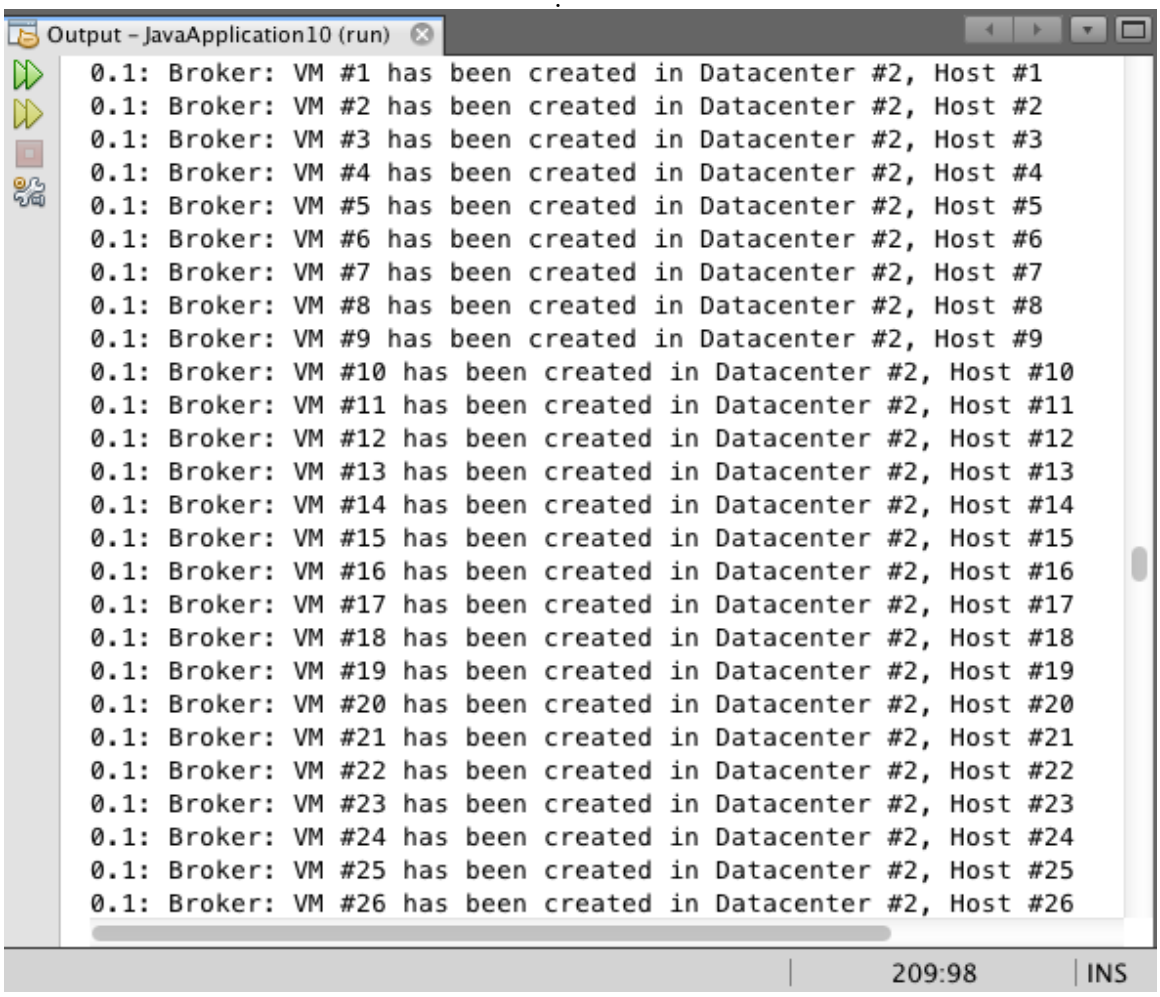


Figure 4.2: VM creation in CloudSim

fuzzy classification, Naive Bayes, Decision tree were considered. But the best among all was observed to be Random Forest with minimum error rate.

Weka has been used to implement Random Forest classification algorithm. Weka is a data mining software written in Java. It implements a number of machine learning classification algorithms. It provides a number of visualisation tools that aids user in understanding the machine algorithms better. Apart from classification algorithms, Weka also aids in regression, clustering and feature selection.

The very first step in Weka is data preprocessing. It includes storing the data to be classified in Comma Separated Value(CSV) or Attribute Relation File Format(ARFF). Apart from this, noise or any missing values should be removed from the dataset. So, the percentage usage of every cloudlet should be stored in a CSV file. There must not be any noise or missing values in the data. The data should be trained manually for some data instances so as to train the Random Forest classifier model. After loading the desired file in Weka, make sure the target class label attribute should be nominal. Now start the classification process and make sure values of evaluation parameters like accuracy, Minimum Error Rate(MER) of the running model should be as desired. The results should be stored in a CSV output file.

4.4 Interference Aware Scheduling

Interference Aware Scheduling is allocation of tasks in such a way so that two tasks with same intent do not hinder each other's work to completion. It helps in optimisation of the scheduling process. The classification of tasks among C.P.U. intensive, Memory intensive and Network intensive helps in realising the desired scheduling policy.

Different tasks must be scheduled on Virtual Machines in such a way so as to avoid interference among the tasks. Any number of tasks classified under same class must not be scheduled on same Virtual Machine at the same time. This will lead to interference among the tasks resulting in poor performance. As a result, Round Robin has been chosen as the scheduling policy. First of all, all C.P.U intensive tasks are scheduled on different Virtual Machines. Next Memory

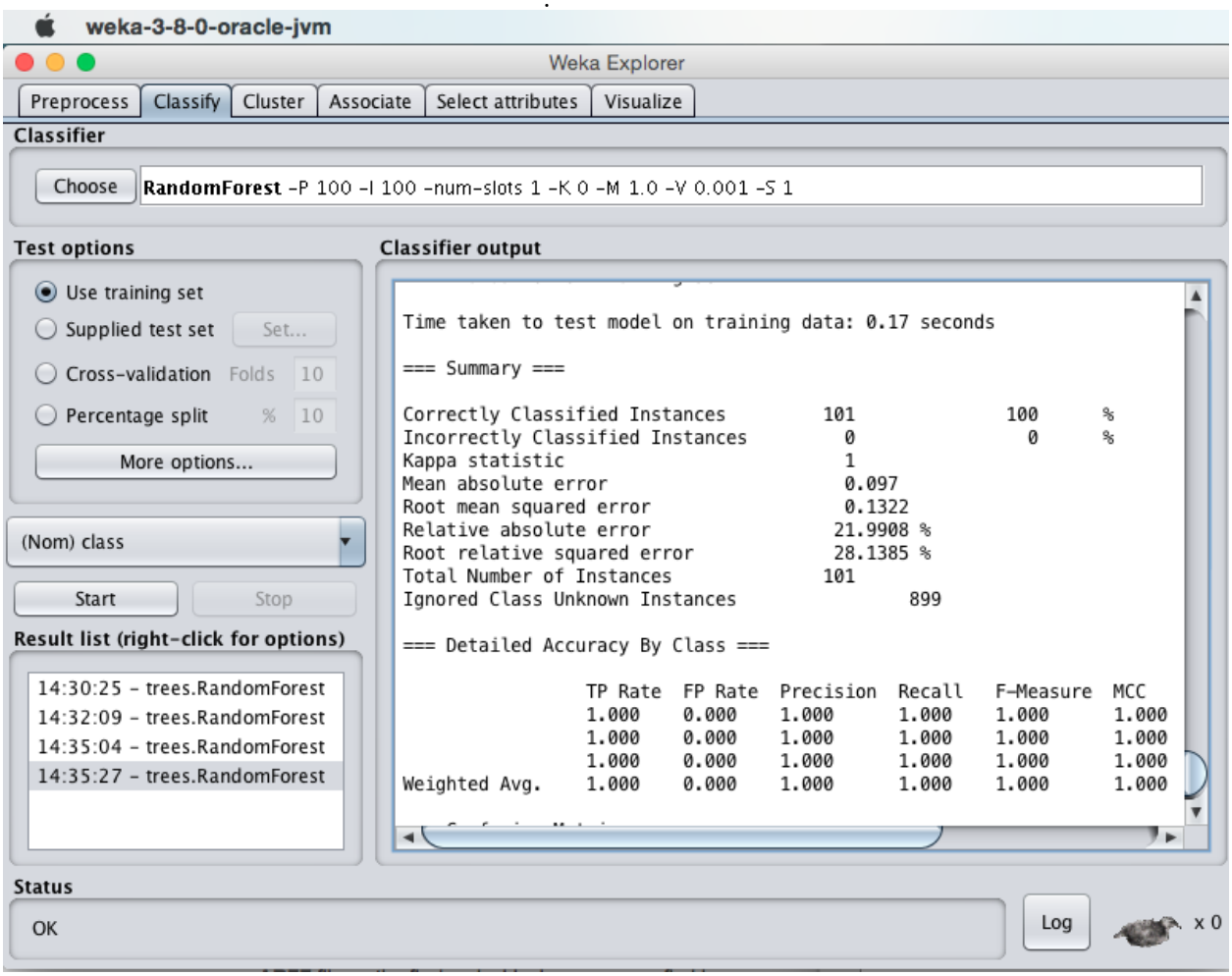


Figure 4.3: Evaluation Parameters of Random Forest

inst#	actual	predicted	prediction
1	1:B	1:B	0.78
2	2:A	2:A	0.92
3	3:C	3:C	0.98
4	3:C	3:C	0.93
5	2:A	2:A	0.85
6	1:B	1:B	0.78
7	3:C	3:C	0.73
8	1:B	1:B	0.78
9	3:C	3:C	0.96
10	3:C	3:C	0.87
11	2:A	2:A	0.79
12	1:B	1:B	0.9
13	2:A	2:A	0.97
14	2:A	2:A	0.8
15	2:A	2:A	0.98
16	2:A	2:A	0.99
17	2:A	2:A	0.83
18	3:C	3:C	0.89
19	1:B	1:B	0.85
20	3:C	3:C	0.84
21	3:C	3:C	0.96
22	3:C	3:C	0.89
23	2:A	2:A	0.93
24	1:B	1:B	0.68
25	2:A	2:A	0.76
26	1:B	1:B	0.62
27	3:C	3:C	0.8

Figure 4.4: Predicted Class Labels

intensive and then Network intensive are scheduled on different Virtual Machines. As number of Virtual Machines are less than number of cloudlets, when all of the virtual machines get exhausted the next cloudlet is scheduled on the very first Virtual Machine as in Round Robin fashion[23].

Round Robin works in a cyclic manner. It allocates tasks to VM #0 to VM #399 in one go since only 400 Virtual Machines have been used. Round Robin has been identified as Time-Shared Scheduling of cloudlets on cloud. In Time-Shared Scheduling Policy, Virtual Machine receives a time slice on each processing core during which it completes the execution of a particular cloudlet. As it is applied to both Virtual Machines and tasks, the processing power is shared by each Virtual Machine and each Virtual Machine is shared by number of task units[9]. Following algorithm has been proposed:

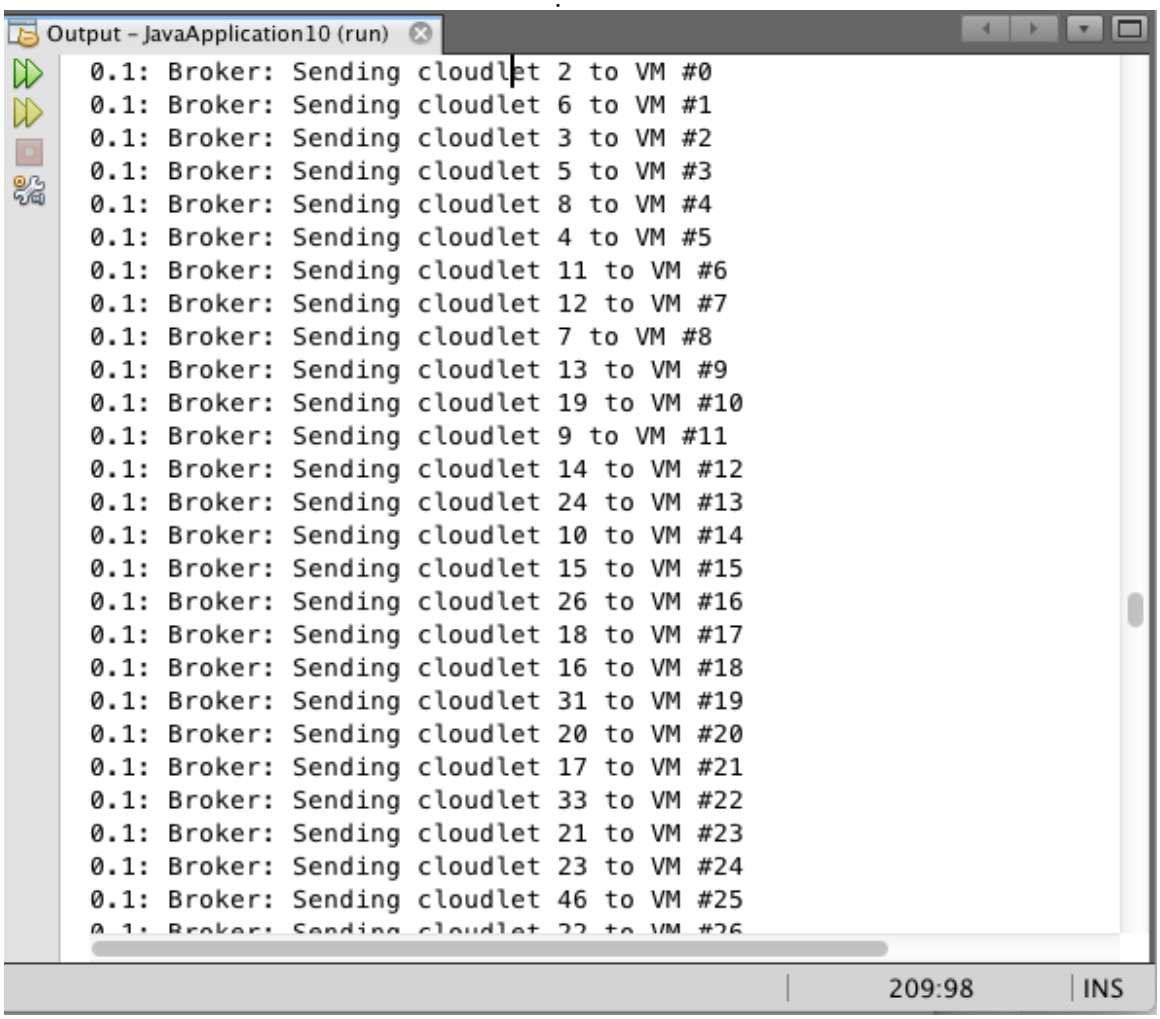


Figure 4.5: Cloudlet Scheduling

Algorithm 1 Interference Aware Scheduling Algorithm

1: **procedure** INTERFERENCE AWARE SCHEDULING
2: $i \leftarrow$ Task assigned to cloud
3: $cpu \leftarrow$ CPU utilisation of cloudlet %
4: $mem \leftarrow$ Memory utilisation of cloudlet %
5: $nw \leftarrow$ Network utilisation of cloudlet %
6: $cpulist \leftarrow$ list of cloudlets with high CPU utilisation
7: $memlist \leftarrow$ list of cloudlets with high Memory utilisation
8: $nwlist \leftarrow$ list of cloudlets with high Network utilisation
9: $nc \leftarrow$ *Number of cloudlets in cpulist*
10: $nm \leftarrow$ *Number of cloudlets in memlist*
11: $nn \leftarrow$ *Number of cloudlets in nwlist*
12: $nv \leftarrow$ *Number of Virtual Machines*
13: $j \leftarrow$ *Each virtual machine*
14: *top*:
15: **if** $cpu > mem$ & $cpu > nw$ **then** Add it to cpu list;
16: **else if** $mem > cpu$ & $mem > nw$ **then** Add it to memlist;
17: **else** Add it to nwlist;
18: **end if**
19: $i \leftarrow 0$
20: $j \leftarrow 0$
21: **for** $i \leftarrow 1, nc$ **do**
22: $i \leftarrow i + 1$
23: $j \leftarrow j + 1$
24: **if** $j \neq$nv **then**
25: $j \leftarrow (j + 1)\%nv$
26: **end if**
27: **end for**

```
28:     i ← 0
29:     for i ← 1, nm do
30:         i ← i + 1
31:         j ← j + 1
32:         if j ≠nv then
33:             j ← (j + 1)%nv
34:         end if
35:     end for
36:     i ← 0
37:     for i ← 1, nn do
38:         i ← i + 1
39:         j ← j + 1
40:         if j ≠nv then
41:             j ← (j + 1)%nv
42:         end if
43:     end for
44: end procedure
```

Chapter 5
Results and Discussions

As discussed above, Interference among the cloudlets lead to significant performance degradation as two cloudlets with high intent for a particular resource can hinder the progress of other cloudlet. Keeping in mind the evaluation results obtained from interference aware methodology, this approach has resulted in significant performance improvements.

Figure 5.1 summarises the results obtained when there is interference observed among the cloudlets. As evident from the Figure 5.1, the number of cloudlets finishing in 3000 seconds is more than that of interference aware methodology . Figure 5.2 summarises the results obtained when the interference aware scheduling is applied on different Cloudlets, while Figure 5.3 shows the comparison of finish time observed with two counter approaches and Figure 5.4 shows comparison of two approaches in terms of number of tasks completed in specific time interval. Figure 5.5 shows final output displayed on the CloudSim. The average finish time of cloudlets observed in interference aware scheduling of cloudlets is 1748 seconds where as the average finish time of cloudlets in conventional scheduling policy is 2600 seconds.

Clearly, interference leads to delayed execution of cloudlets. Instead if we apply interference aware scheduling proposed in section 4.4, cloudlets tend to complete their execution in a shorter span of time.

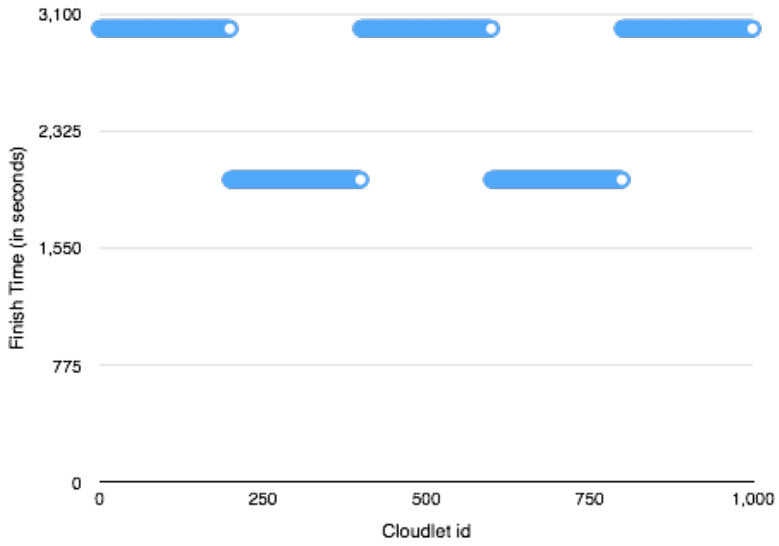


Figure 5.1: Performance Evaluation of Non-Interference Aware Scheduling

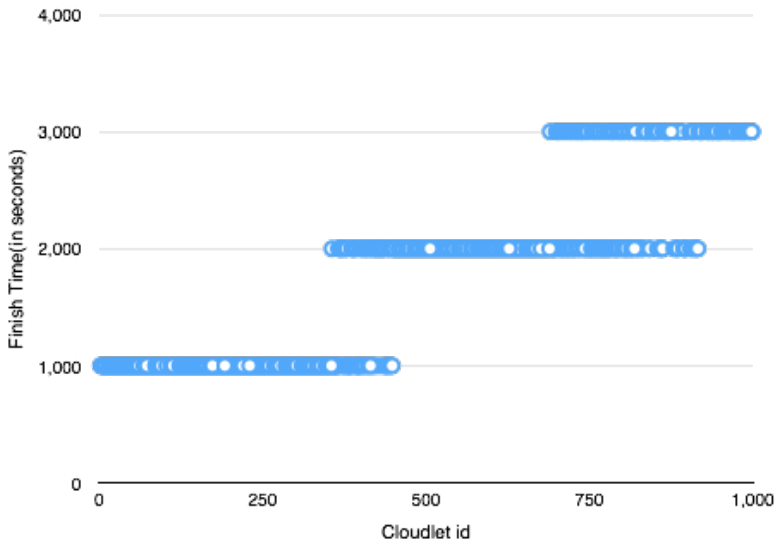


Figure 5.2: Performance Evaluation of Interference Aware Scheduling

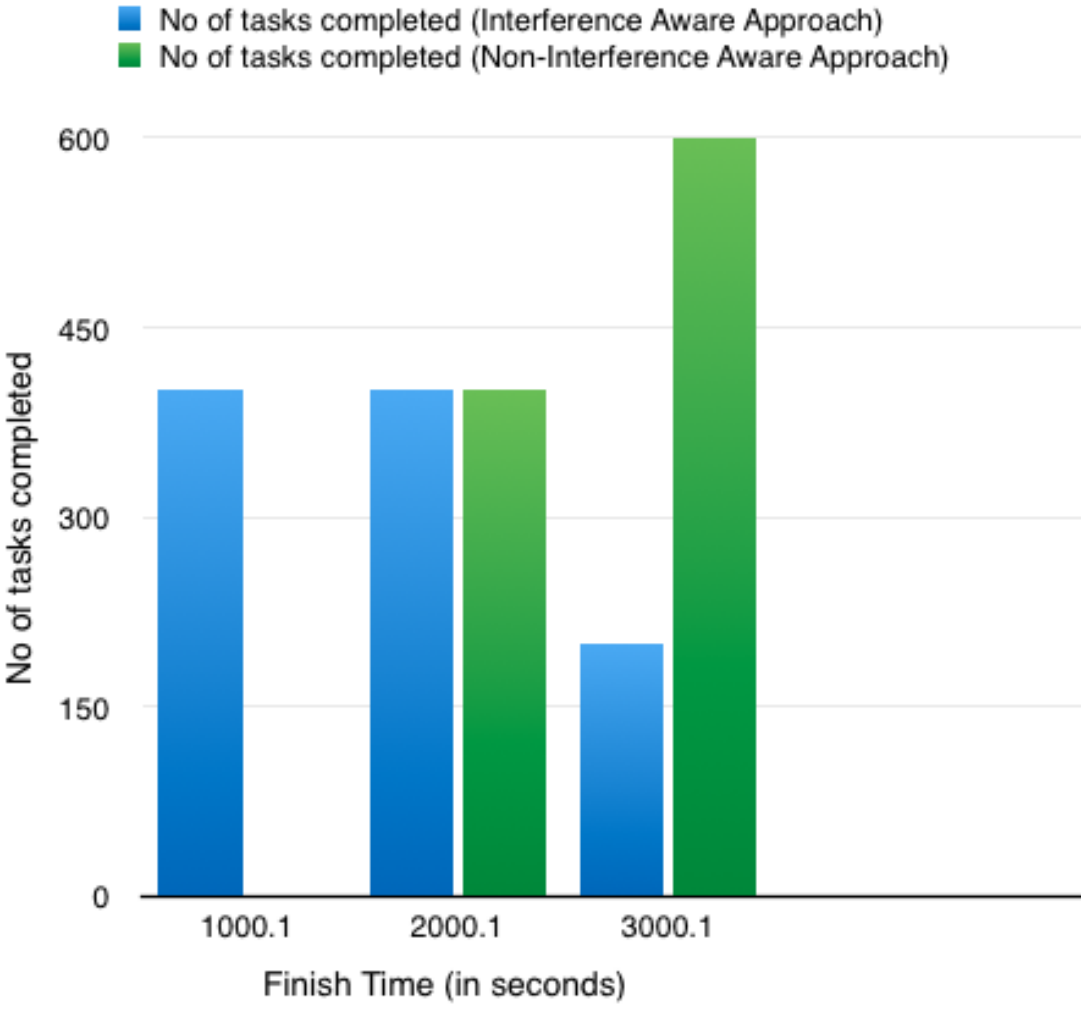


Figure 5.3: Comparison of the two approaches

cloudlet id	Vm id	Execution time	Start time	Finish time
2	0	1,000	0.1	1,000.1
177	200	1,000	0.1	1,000.1
6	1	1,000	0.1	1,000.1
193	201	1,000	0.1	1,000.1
3	2	1,000	0.1	1,000.1
231	202	1,000	0.1	1,000.1
5	3	1,000	0.1	1,000.1
180	203	1,000	0.1	1,000.1
8	4	1,000	0.1	1,000.1
196	204	1,000	0.1	1,000.1
4	5	1,000	0.1	1,000.1
233	205	1,000	0.1	1,000.1
11	6	1,000	0.1	1,000.1
181	206	1,000	0.1	1,000.1
12	7	1,000	0.1	1,000.1
197	207	1,000	0.1	1,000.1
7	8	1,000	0.1	1,000.1
237	208	1,000	0.1	1,000.1
13	9	1,000	0.1	1,000.1
183	209	1,000	0.1	1,000.1
19	10	1,000	0.1	1,000.1
198	210	1,000	0.1	1,000.1
9	11	1,000	0.1	1,000.1
239	211	1,000	0.1	1,000.1
14	12	1,000	0.1	1,000.1

Figure 5.4: Final Output

Chapter 6
Conclusion and Future Scope

6.1 Conclusion

In today's era, Cloud Computing has been embraced by every single organisation, from start-ups to global organisations. Cloud Computing provides scalable solutions to every kind of business demand. For start-ups, as the business grows cloud services required also grow. Moreover, Cloud Computing serves the best environment for innovation as it makes deployment of new ideas very easy and at reasonable costs. Being able to connect business all over the world, it makes flow of innovative ideas very easy. In case of data loss, it provides data back-up recovery solutions. Cloud Computing has proved to be a robust technology in every way. So the services provided by Cloud Computing are ideal for every kind of business demands. But with the increasing demand, there comes performance related issues too. One of them is interference among the tasks with same intent. In order to address the issue of interference, we need to identify the intent of various tasks and classify them accordingly. These tasks should be scheduled on Virtual Machine such that tasks belonging to same class are scheduled on different Virtual Machines. This will surely lead to an interference free scheduling of tasks on cloud.

6.2 Future Scope

The above stated scheduling technique can be made more robust by dynamic creation of tasks. Tasks can be created dynamically instead of the static approach. Each task when recognised by the system is immediately classified and scheduled on the desired Virtual Machine, keeping in mind the class of each task. In future work, we can also evaluate this approach for varied number of datacenter, Virtual Machines and hosts. This will help in judging the performance of the algorithm with interference aware approach. We can also take various parameters to judge the performance of cloud application with this approach like throughput of each Virtual Machine, running time of each task in the application etc.

References

[1] S. Wizdom, "How onedrive syncs between multiple clients and the onedrive server," https://answers.microsoft.com/en-us/onedrive/wiki/odoptions-oddesktop/how-onedrive-syncs-between-multiple-clients-and/aaa138d7-ade4-48c1-9d7c-c8b8755712db, January 2014.

[2] J. Strickland, "How grid computing works," http://computer.howstuffworks.com/grid-computing.htm, May 2017.

[3] H. Harikrishnan, "Utility computing: Point of arrival," http://harikrish.net/business/colocation-to-utility-computing/, July 2016.

[4] J. M. Willis, "Ibm and the history of autonomics," http://www.johnmwillis.com/ibm/ibm-and-the-history-of-autonomics/, March 2008.

[5] A. Kumawat, "Cloud service models (iaas, saas, paas) + how microsoft office 365, azure fit in," http://www.cmswire.com/cms/information-management/cloud-service-models-iaas-saas-paas-how-microsoft-office-365-azure-fit-in-021672.php, July 2013.

[6] N. Vold, "Cloud basics – deployment models," https://www.visma.com/blog/cloud-basics-deployment-models/, March 2012.

[7] A. Upadhyay, "Cloud computing concept," https://atalupadhyay.wordpress.com/this-is-my-new-page-of-wordpress/cloud-computing-concept/, July 2013.

[8] M. T. Jones, "Virtual linux: Platform and os linux virtualization," http://www.datamation.com/osrc/article.php/3879871/Virtual-Linux-Platform-and-OS-Linux-Virtualization.htm, May 2010.

[9] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, no. 1, pp. 23–50, 2011.

[10] K. Govindan, "Data mining and data warehouse," http://govindank97.blogspot.in, January 2017.

[11] D. Benyamin, "Random forest," http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics, November 2012.

[12] C. Delimitrou and C. Kozyrakis, "Paragon: Qos-aware scheduling for heterogeneous datacenters," in *ACM SIGPLAN Notices*, vol. 48, no. 4. ACM, 2013, pp. 77–88.

[13] M. Walker, "Random forest algorithm," http://www.datasciencecentral.com/profiles/blogs/random-forests-algorithm, May 2017.

[14] P. Mell and T. Grance, "The nist definition of cloud computing," 2011.

[15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[16] L. Malhotra, D. Agarwal, and A. Jaiswal, "Virtualization in cloud computing," *International Journal of Computer Science and Mobile Computing*, vol. 4, no. 2, 2014.

[17] V. A. Lepakshi and D. P. CSR, "A study on task scheduling algorithms in cloud computing," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 11, pp. 119–125, 2013.

[18] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," *International Journal of Research in Engineering and Technology*, vol. 2, no. 2, pp. 131–135, 2013.

[19] R. C. Chiang and H. H. Huang, "Tracon: Interference-aware scheduling for data-intensive applications in virtualized environments," in *TRACON: Interference-aware scheduling for data-intensive applications in virtualized environments*. ACM, 2011, pp. 47–50.

[20] R. Nathuji, A. Kansal, and A. Ghaffarkhah, "Q-clouds: managing performance interference effects for qos-aware clouds," in *Proceedings of the 5th European conference on Computer systems*. ACM, 2010, pp. 237–250.

[21] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, "iaware: Making live migration of virtual machines interference-aware in the cloud," in *IEEE Transactions on Computers*, vol. 63, no. 12. IEEE, 2014, pp. 3012–3025.

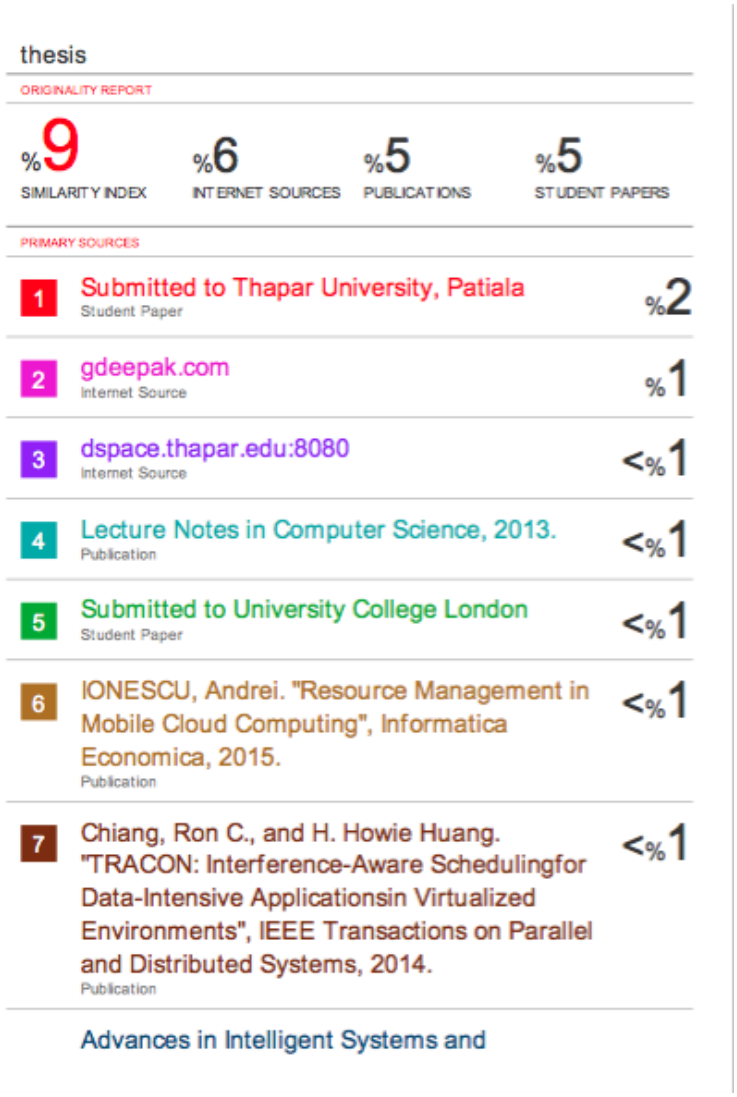
[22] W. Zhang, S. Rajasekaran, T. Wood, and M. Zhu, "Mimp: Deadline and interference aware scheduling of hadoop virtual machines," in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 2014, pp. 394–403.

[23] R. V. Rasmussen and M. A. Trick, "Round robin scheduling–a survey," *European Journal of Operational Research*, vol. 188, no. 3, pp. 617–636, 2008.

List Of Publications

- Accepted paper titled "Interference Aware Scheduling Of Tasks On Cloud" in 4th International Conference on "Signal Processing, Computing and Control (ISPCC-2017)", sponsored by IEEE to be held in Jaypee University Of Information Technology, Waknaghat on 21st September to 23rd September,2017.

Plagiarism Report



thesis

ORIGINALITY REPORT

%9	%6	%5	%5
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Thapar University, Patiala Student Paper	%2
2	gdeepak.com Internet Source	%1
3	dspace.thapar.edu:8080 Internet Source	<%1
4	Lecture Notes in Computer Science, 2013. Publication	<%1
5	Submitted to University College London Student Paper	<%1
6	IONESCU, Andrei. "Resource Management in Mobile Cloud Computing", Informatica Economica, 2015. Publication	<%1
7	Chiang, Ron C., and H. Howie Huang. "TRACON: Interference-Aware Schedulingfor Data-Intensive Applicationsin Virtualized Environments", IEEE Transactions on Parallel and Distributed Systems, 2014. Publication	<%1

Advances in Intelligent Systems and