Task Scheduling In Cloud Computing

A Dissertation submitted in partial fulfillment of the requirement for theAward of degree of

MASTER OF TECHNOLOGY

IN

INFORMATION SYSTEMS

Submitted By **Rahul Parashar**

(2K13/ISY/19)

Under the guidance of

Dr. N. S. RAGHAVA

Associate Professor

Department of Computer Science and Engineering

Delhi Technological University

Bawana Road, Delhi-110042

2013-2015

CERTIFICATE

This is to certify that the thesis entitled "**Task scheduling in cloud computing" submitted by Rahul Parashar (2k13/ISY/19)** to Delhi Technological University, Delhi for the award of the degree of **Master of Technology** is a bonafide record of research work carried out by him under my supervision.

The content of this Thesis, in full or in parts, have not been submitted to any other institute or University for the award of any degree or diploma.

Date:

Dr. N S Raghava
Project Guide
Associate Professor
Department of
Delhi Technological University, Delhi

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and **DELHI TECHNOLOGICAL UNIVERSITY**. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Dr. N.S. Raghava, *Project Guide*** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my special gratitude and thanks to **Dr. O.P. Verma *Head of Dept.*** for giving me such an opportunity to work on the project.

I would like to extend my gratitude towards my **parents** &**staff** of Delhi Technological University for their kind co-operation and encouragement which helped me in the completion of this project.

My thanks and appreciations also go to my **friends and colleagues** in developing the project and people who have willingly helped me out with their abilities.

Rahul Parashar
Roll No:2k13/ISY/19
Dept. of Information Technology
Delhi Technological University

ABSTRACT

Cloud computing is a way through which a user can access all types of resources from the location that is different from the location where his system presents. Basically cloud computing is a combination of grid computing, utility computing (Complete package of computer resources that can be used as a metered service) and autonomic computing (ability to do self management). Job scheduling is one of the most necessary tasks in a cloud computing environment because cost is associated with every resource based upon the time that is used by the user.

Generally such type of task scheduling algorithm that gives optimized solution in cloud computing. Also, the utilization of resources must be efficient, for that purpose, we have to build such type of job scheduling algorithm that does better utilization of resources. There have been various varieties of scheduling algorithm applied in a distributed computing system. By making certain changes, we can apply them in a cloud computing environment. The main goal of the task scheduling algorithm is to maximum resource allocation, gain high computation and the best system throughput. In our proposed task scheduling algorithm, tasks are submitted to the virtual machines by applying particle swarm optimization technique through which maximum number of tasks can be submitted which proves best utilization of resources.

In our algorithm, we vary the number of tasks for execution and virtual machine count for execution. This algorithm is implemented in JAVA platform using eclipse IDE. In our algorithm, we have few constraints that migration of tasks from one virtual machine to other is possible, arrival time of all the tasks are considered to be same. The swarm intelligence helps in convergence of solution finding in the big search space more quickly. Execution of the algorithm results an optimized schedule which provides maximum resource allocation possible in short span of time.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION To CLOUD COMPUTING

1.1 COMPUTING MODELS.

DESKTOP COMPUTING ----- Single independent and individual pc is used for computing. No need of internet or network connection is required for the computing purposes. Personal as well as professional work can be done using software like MS office suite, AutoCAD, Photoshop, illustrator, visual studio, CMS etc for engineers, doctors, programmers, artists etc.

CLIENT/SERVER COMPUTING ---- This type of computing is generally used in business organizations where databases are put on server machines and most of the application software are run on client machines. This model is used by different type of companies e.g. Banks, retail stores, Oil companies, automobiles companies each require its customized software like accounting software, distribution software, ERP etc for this model to work for their operations.

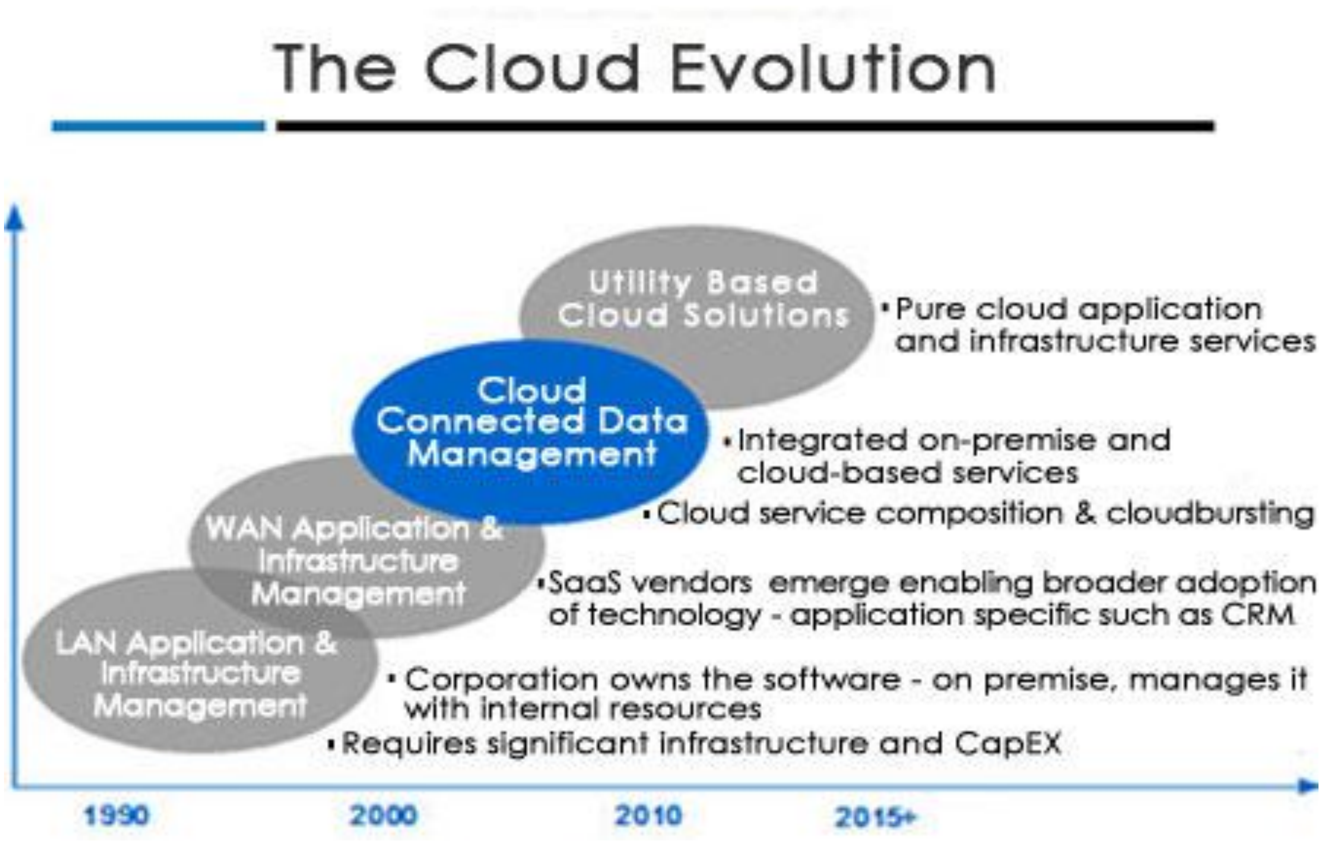
CLUSTER COMPUTING ---- In this type of computing a group of homogenous servers are used onto which loads are distributed. Servers are of one type only for example they can be just database servers which store and maintain company data.

GRID COMPUTING ---- It is the special case of client/server computing. It comprises of different heterogonous systems connected together to work as a whole. It is differ from cluster computing in a way that grid involves geographically distant computing nodes to be connected to work as a unit. Different application software and server software are integrated to work as a whole.

CLOUD COMPUTING ---- It is a combination of grid and cluster computing. Cloud can be called as a grid of clusters. A company that has a worldwide existence requires this type of infrastructure. For example Wall-mart, Toyota, TATA Group, City Bank etc. has their own clouds. Infrastructure maintenance of their cloud is done by other companies like IBM, Sun, HP, Intel, Cisco etc. and software companies that provide software for the working of cloud network are SAP, oracle apps, Microsoft dynamics etc.

1.2 EVOLUTION OF CLOUD COMPUTING

Before getting popular, cloud computing existed but in the form of grid computing and cluster computing. It is because of the big companies like Google, Amazon, eBay cloud computing has become famous. Earlier these companies provide free internet services like email by Google, Blogs by blogger, Chat by yahoo, social networking by Face book etc so that their website contents had become larger by the users. In the course of time their servers become overloaded therefore they needed to increase their servers in numbers and as widespread more of the users were connected to their websites they had to spread their companies worldwide. In this way few servers become clusters and few clusters become grid. This cluster of grid over internet is now known as cloud. Companies had decided to provide the services provided by cloud over internet to different users. In this way it has become popular but it this type of computing had been used previously. Now a day's Cloud computing has become a hotspot technology in the world as it provides a shared pool of hardware and software resources on demand over Internet.



The Cloud Evolution
Utility Based Cloud Solutions
• Pure cloud application and infrastructure services
Cloud Connected Data Management
• Integrated on-premise and cloud-based services
• Cloud service composition & cloudbursting
WAN Application & Infrastructure Management
• SaaS vendors emerge enabling broader adoption of technology - application specific such as CRM
LAN Application & Infrastructure Management
• Corporation owns the software - on premise, manages it with internal resources
• Requires significant infrastructure and CapEX
1990
2000
2010
2015+

Fig. 1.1 Evolution of cloud computing

1.3 CLOUD

Cloud is a group of thousand of computers that appears like a single giant computer that is running personal computer, cell phone or any other device that is capable of using Internet and that can share resources to provide services on demand to the end users. The services can be storage for example. Cloud can provide 1GB or 2 GB storage capacity to the users that can be shrinking or expand on demand basis. Besides cloud, end user cannot get such type of facility as it cannot separate 2GB from a 80 GB hard disk. The services provided by the cloud are highly scalable, reliable,flexible, sustainable etc. [1] Various dynamic services are provided by the cloud computing environment with high scalability and reliability. According to the cloud service provider and the user of the services the perspective of the cloud changes.

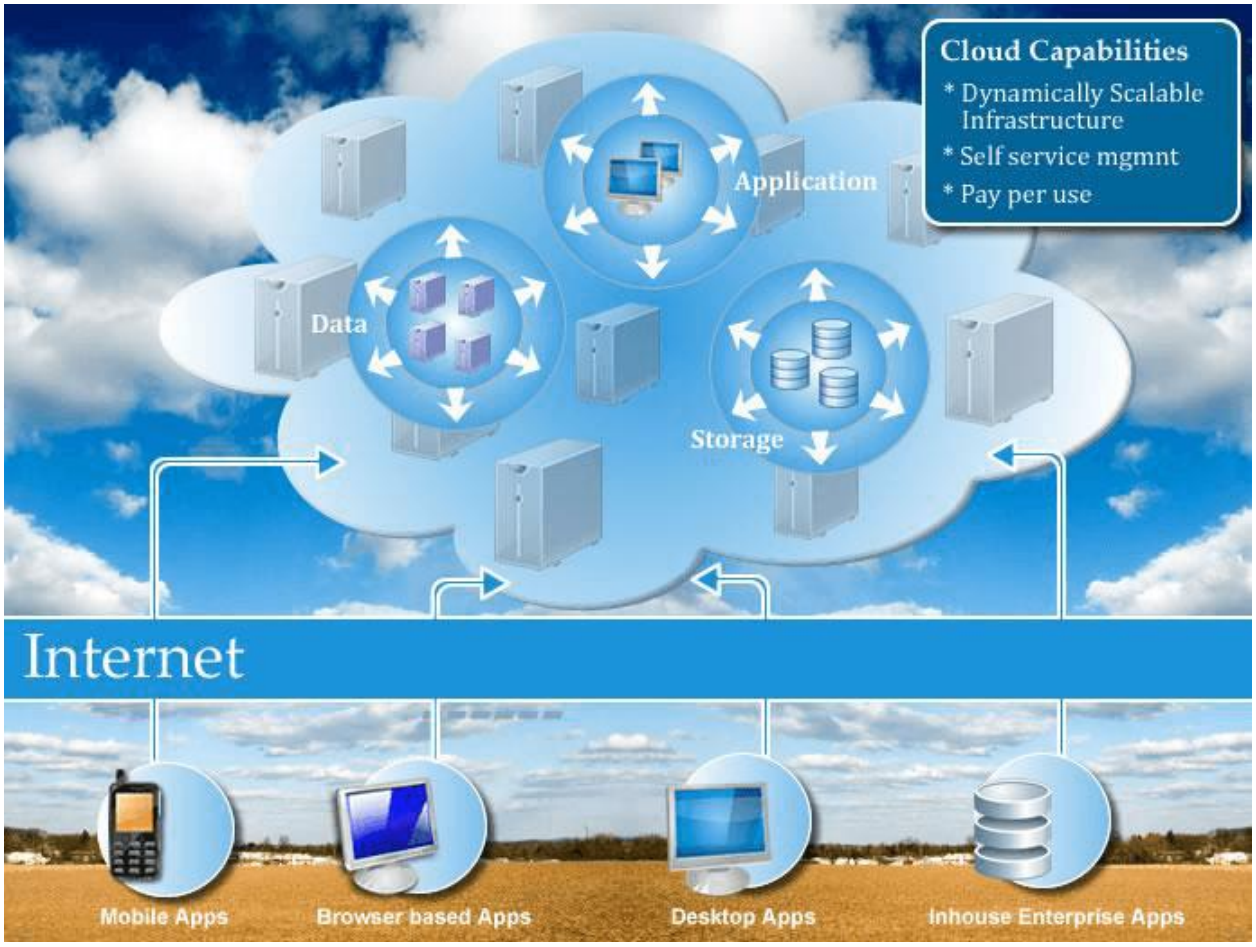


Fig. 1.2 Conceptual view of cloud computing

CLOUD CHARACTERISTICS:-

HIGHLY SCALABLE

A growing company requires its infrastructure to be maintained at a low cost. Cloud provides such companies that infrastructure which can be expand or shrink without any extra cost incur to the company. These overheads are handled by the cloud service providers so that company can only focus in its business areas. For them cloud is like a black box with the demanded services which can be easily shrink or expanded.

ON DEMAND CAPABILITIES

A business will secure cloud-hosting services through a cloud host provider which could be your usual software vendor. You can change your services online. You have the power to add or delete a service from your service list. Typically, you are billed with a monthly subscription or a pay-for-what-you-use scenario. Terms of subscriptions and payments will vary with each software provider.

1.3.1 PRIVATE CLOUD

Private cloud is one which is used within an organization i.e. access is limited to employee of the organization only. It is mainly used for security and privacy only and to solve this purpose access rights are given to authorize people only. Anyone can create his own private cloud by using certain platform which is available online. Private cloud is of following type:-

1. **ON-PREMISE:** They are also called as internal clouds. Cloud is hosted within own datacenter of the company.
2. **EXTERNALLY HOSTED**: It refers to the cloud services which are hosted by third party external to the company's data center. Cloud provider provides a isolated services through which high level of privacy and secrecy can be obtained.

1.3.2 PUBLIC CLOUD

All the resources are public and shared by the users. No dedicated resource is used to create this type of cloud. Users can easily access theses type of services through internet. Public cloud is connected to public Internet for anyone to leverage it. The services offered on public clouds may be free or offered on pay per usage model. Advantage of Public cloud over private cloud is that they are cheaper than the latter but offer less secure services than private cloud.Examples of public clouds are:

1. **Amazon Elastic Compute cloud.**
2. **IBM Blue cloud.**
3. **Sun cloud.**
4. **Google AppEngine.**
5. **Windows Azure Services Platform.**
6. **Citibank cloud.**

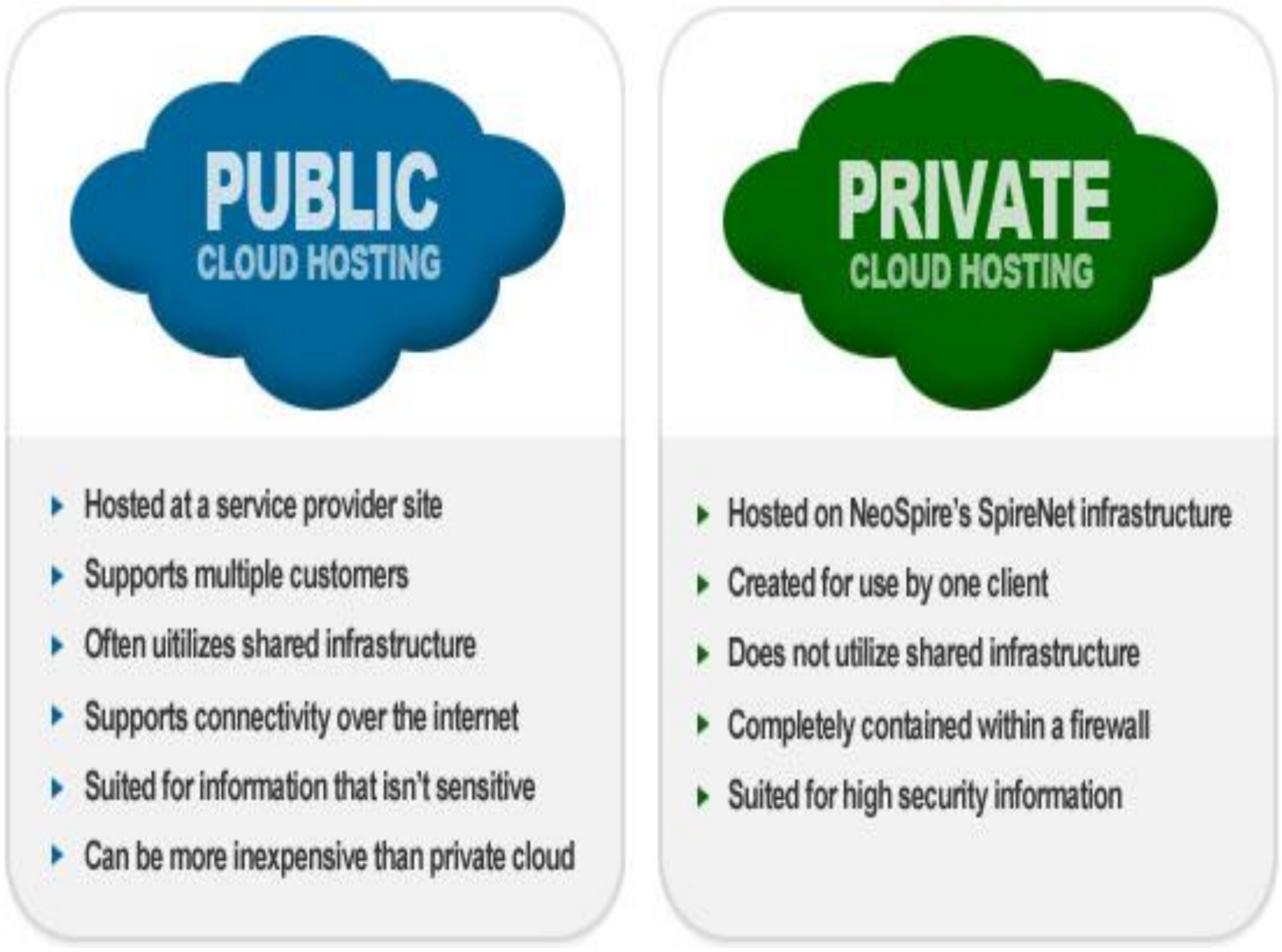

PUBLIC
CLOUD HOSTING
Hosted at a service provider site
Supports multiple customers
Often uitilizes shared infrastructure
Supports connectivity over the internet
Suited for information that isn't sensitive
Can be more inexpensive than private cloud
PRIVATE
CLOUD HOSTING
Hosted on NeoSpire's SpireNet infrastructure
Created for use by one client
Does not utilize shared infrastructure
Completely contained within a firewall
Suited for high security information

Fig 1.3 Public vs Private cloud

1.3.3 Hybrid Cloud

Hybrid Cloud is a cloud computing environment that provides hybrid of on premises, private and public cloud services with orchestration between the two platforms. It provides greater flexibility and by allowing load to move between public and private clouds.

1.3.4 Community Cloud

It is a cloud computing environment in which an infrastructure is shared among different organization from a common community with common goals. It can be managed internally or by a 3rd party. Also it can be hosted internally or externally. They are created to reduce cost and to maximize profits. As an example, many organizations require a same set of applications that reside on one set of cloud. So instead of giving them their own servers on cloud, a possible solution is to provide them a single server and let multiple customers connect to their environment and logically segment their sessions. In this way customers are using the same piece of hardware but still running their application – this makes it a community cloud.

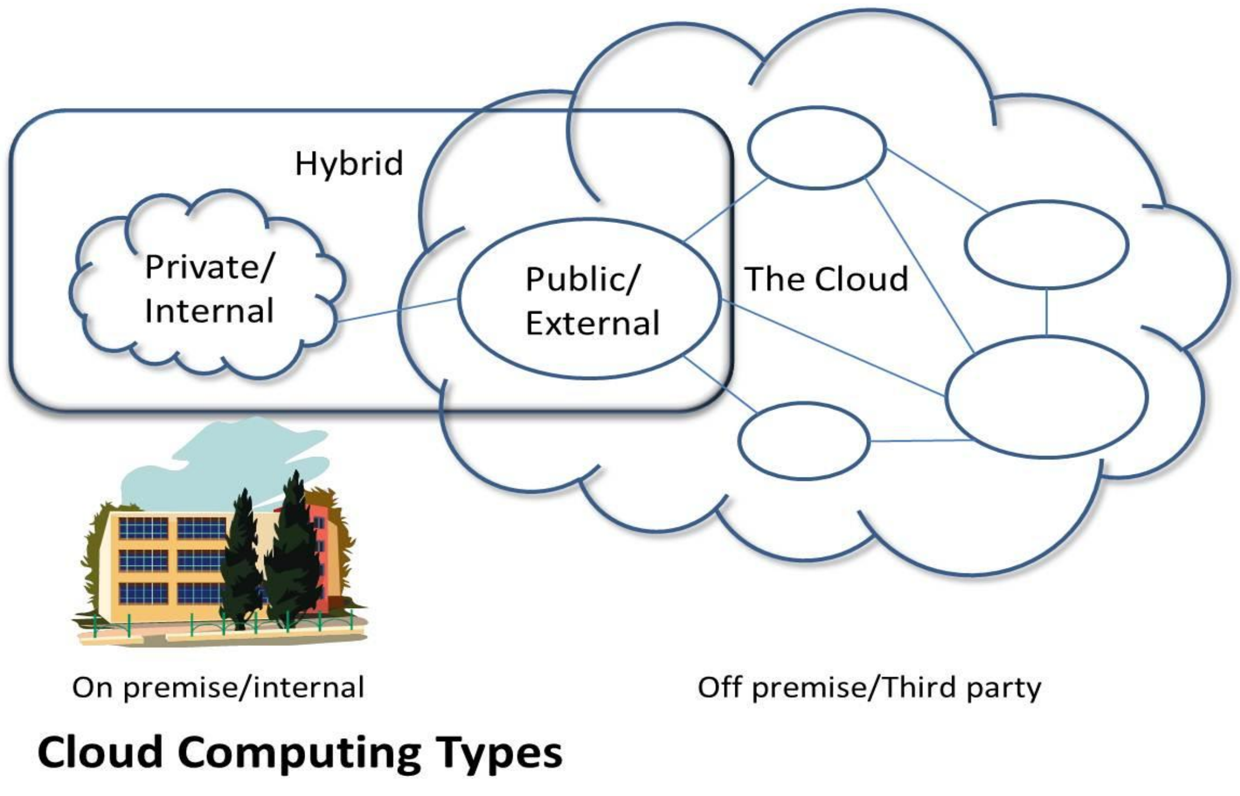


Fig. 1.4 Types of cloud computing

1.4 Virtualization

Virtualization is the process of creating a virtual version of something and that something includes computer hardware platform, operating systems, storage device, network resources etc. It is the key component of cloud computing. It can simply be the separation of operating system from the underlying hardware. Because of this separation different operating platforms can now share the same underlying hardware. Therefore it became easy to add another operating system or migrate it to different hardware. Hence the cost of buying expensive hardware devices drastically reduces and at the same time reliability increases. Virtualization also provide flexibility of having multiple instances of a application running together on multiple instances of different operating system which are sharing the same underlying hardware.

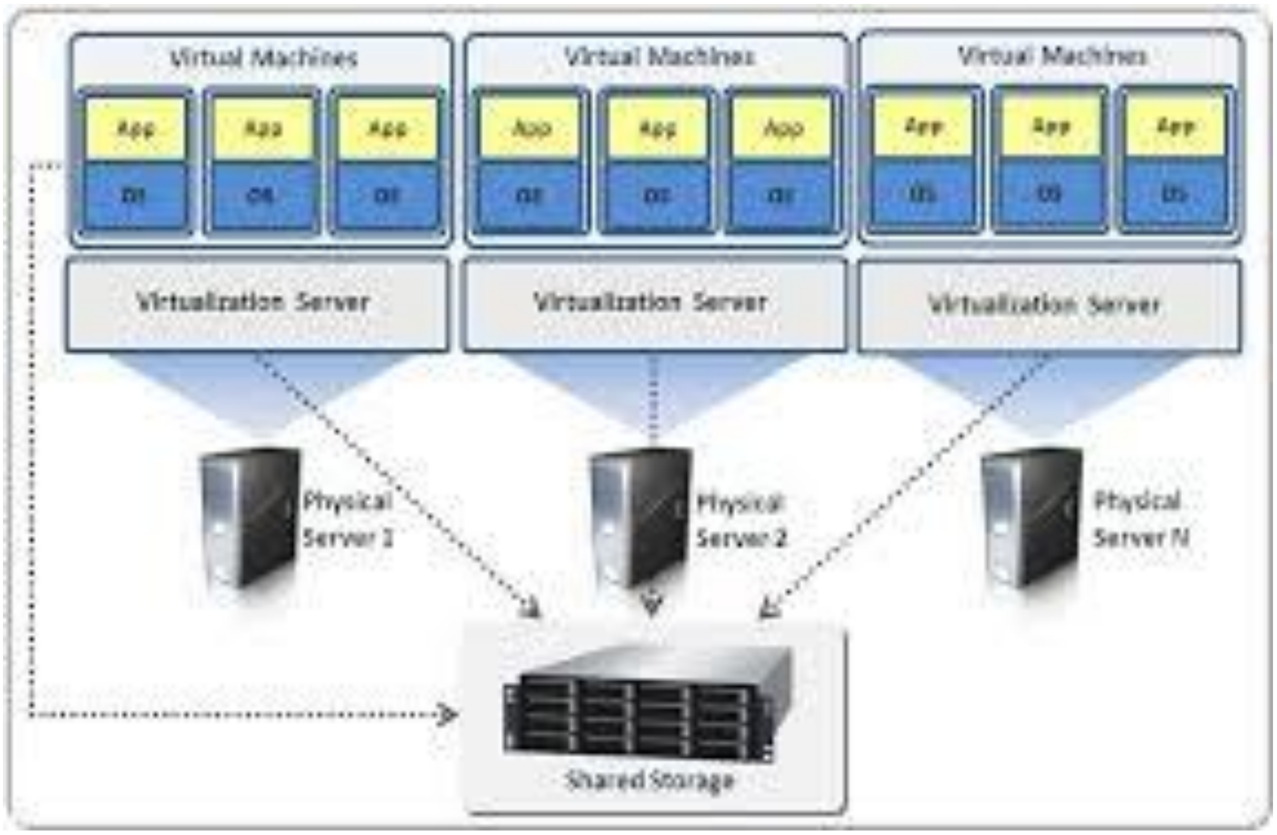


Fig 1.5 Virtualization

1.5 Cloud Computing

Basically, Cloud computing is the separation of application software from the operating system from the hardware that runs everything. To understand this, consider an example where a company needs email services. To start mailing services company need a bundle of hardware, operating system, say Window NT server installed onto that hardware and window mail software installed onto that operating system. All the components of the system are intractably linked. Any problem in any one of it can cause whole mailing services to crash. Either the problem of virus in application or in OS or the hardware failure results in the loss of service since each one is dependent on other. Same will happen with other type of services and the reason for the problem is same and that is the system components are highly linked with each other.

The idea to solve the above mentioned problem is to disconnect all the linked part i.e. disconnect applications from the operating system which in turn from the hardware. This can be achieved through virtual computing. Virtual computing put all the components of a system into its own container and if some component stops working for example if operating system crashes then virtual software automatically shift application running that particular OS to the similar operation system. Therefore no hindrance can be observed in the smooth functioning of the system and the services will never stops from delivering to the customers at any point of time.

Another example of cloud computing is accessing a web application from a desktop. A user can access to the web services using a web browser. It is cloud computing in a way that he can finish his work from anywhere and anytime. All he need is to log in to that service and he can start working from the point where he left last time. All the work is saved on the cloud.

Cloud computing has something to offer everyone whether it is a single user or a professional or an organization of different concerns and size.[12]For the purposes of scientific researches, a large number of resources and equipments for storage, computation can be done with the help of cloud computing. Especially for medium or small size company who cannot afford expensive network equipments, applications and competing in a very rivalry environment. Cloud computing provides more efficient productive environment for developers.

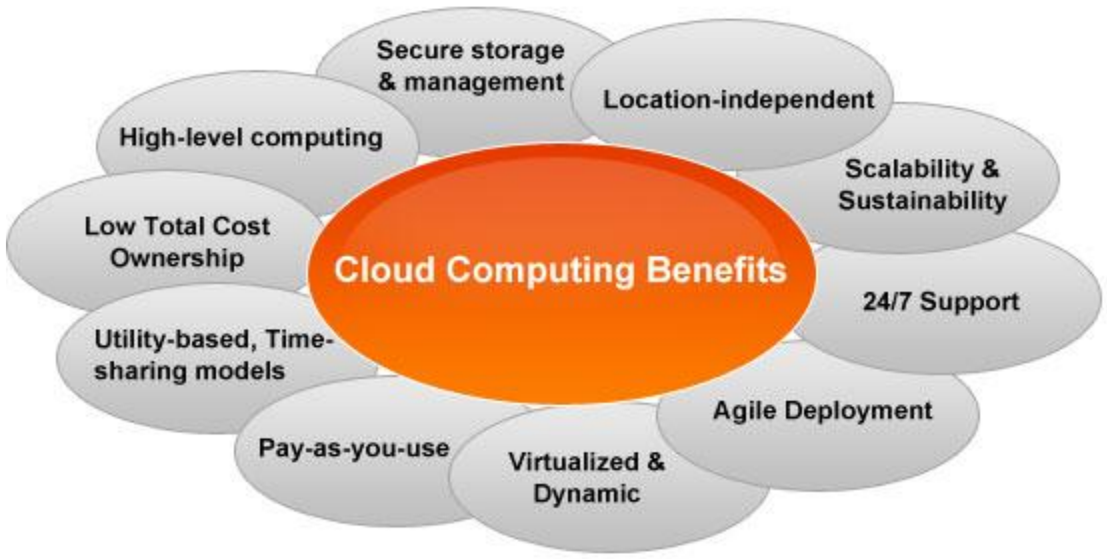


Fig 1.6 Benefits of cloud computing

1.6 SERVICE models

Cloud computing mainly offers three types of services and the models used for them are Platform-as-a-service (PAAS), Software-as-a-service (SAAS) and Infrastructure-as-a-service (IAAS).

1.6.1PLATFORM AS A SERVICE

It is a cloud computing model in which an application has been delivered over the Internet. Cloud provider hosts the hardware and software on their own infrastructure and thereby frees users from installing in-house hardware and software to create or run a new application. Paas components are:-

PRODUCERS:- They produced the platform by integrating hardware and software to create an development environment that is given as a service to customer.

Consumer:- They are those who uses application program interface concept(API) to use the services. They have been charged on a per-use basis.

1.6.2 SOFTWARE AS A SERVICE

In this, a complete software package or component of it is given as a service on demand. Only a single instance of an application is required to run on cloud to provide services to multiple users. Also a single user can access multiple varieties of software available on the cloud.

Characteristics of SaaS

- Commercially available software can be accessed or managed through web.
- Activities can be monitor and managed centrally.
- Software delivery is based on a one to many model rather than one to one model.
- No need for the end users to periodically updates the software.
- For user community benefits, faster releases of new features are provisioned.

1.6.3 INFRASTRUCTURE AS A SERVICE

In this model, virtualized computing resources have been provided over the Internet by a third party. A third party provider hosts hardware, software, servers, storage and othe infrastructure components on the behalf of its users.

Characteristics of IAAS

- Highly scalable services that can be offered on demand.
- Well suited for temporary, experimental or unexpectedly changed workloads.
- Administrative tasks can be automated.
- Dynamic scaling, desktop virtualization and policy based services.
- Common resources are shared among users.

1.6.4 DATABASE AS SERVICE

It offers the services of storage and management of structured data. There is no need of hiring third party to handle administration of databases as they are handled by the cloud providers. Also there is no need for the developers to understand the databases.

1.6.5 NETWORK AS A SERVICE

It is model to provision the virtual network services to the users on a pay-per-use or monthly subscription basis. NaaS can include flexible and extended VPN, custom routing, bandwidth on demand, security firewall, content monitoring and filtering, intrusion detection and prevention, antivirus and many more things.

1.6.6 COMMUNICATION AS A SERVICE

It provides SaaS for communication. CaaS can include Enterprise level VoIP, conferencing(audio and video), individual calls(audio and video), broadcasting messaging and so on. There is no need for investing in creating the infrastructure for communication. There is no need for client organization to hire trained personnel and installing expensive servers for running and managing the communication services.

Some other cloud computing services are listed below:-

- Storage as a service(STaaS).
- Security as a service(SECaas).
- Data as a service(DaaS).
- Desktop as a service(DeaaS).
- Testing as a service(TaaS).

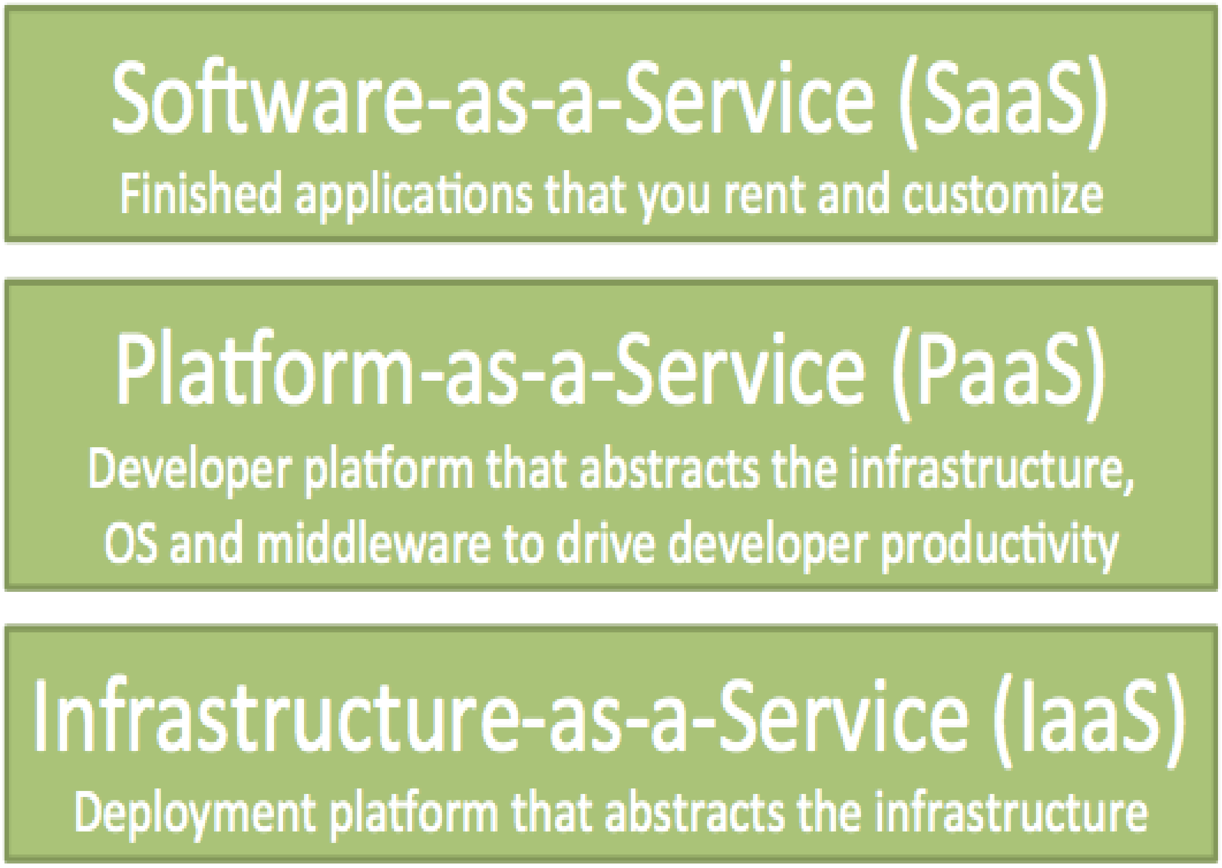

Software-as-a-Service (SaaS)
Finished applications that you rent and customize
Platform-as-a-Service (PaaS)
Developer platform that abstracts the infrastructure, OS and middleware to drive developer productivity
Infrastructure-as-a-Service (IaaS)
Deployment platform that abstracts the infrastructure

Fig 1.7 Three layers of cloud computing

1.7 Cloud Architecture

Cloud architecture is commonly made up of four things – cloud services, platform, infrastructure and storage. Cloud services can be accessed over Internet with the help of web browsers by the cloud clients. Loosely coupled storage architecture provides it consistent and convenient storage access facility. Data nodes can be scaled up into to millions and each separately delivers data to different users.

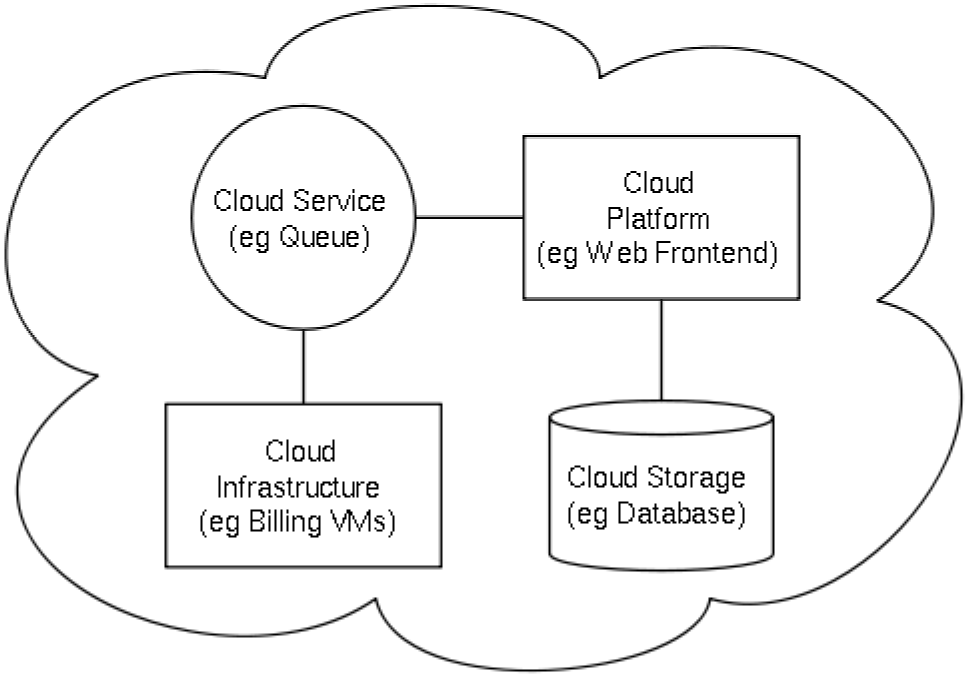

Cloud Service (eg Queue)
Cloud Platform (eg Web Frontend)
Cloud Infrastructure (eg Billing VMs)
Cloud Storage (eg Database)

Fig 1.8 Components of cloud computing

1.8 COMPONENTS OF CLOUD COMPUTING

Main components of Cloud Computing are given below:-

CLIENT:

They are called front end platforms and include servers, fat client, thin client, zero client, mobile devices. They interact with cloud storage via a middle layer that can be a application, through web browsers or through virtual sessions.

SERVICES:

Cloud services consists of real time provisioned products, services and solutions that are available online.

APPLICATION:

Cloud computing applications are accessed through web browsers and multiple users can access parallel from any part of the world. There is no need for downloading and installing of the applications by the users. Cloud providers do all these tasks for the users. For example – Google appEngine

PLATFORM:

Cloud computing provides platform as a service for the end users to create applications over the cloud. Cloud providers provide platform with the configuration of users choice to develop an application. Cloud platform provides great cost benefits with scalability and flexibility on demand.

STORAGE:

Data storage services are provided by the cloud which can be a private, public or a hybrid cloud storage service. Example of storage is Google Drive.

INFRASTRUCTURE:

Cloud service providers provide complete infrastructure for developing, deploying and maintaining applications, handling business operations which offers great benefits to the customers of cloud providers. Charges depends on pay-per-usage or monthly/yearly subscription basis.

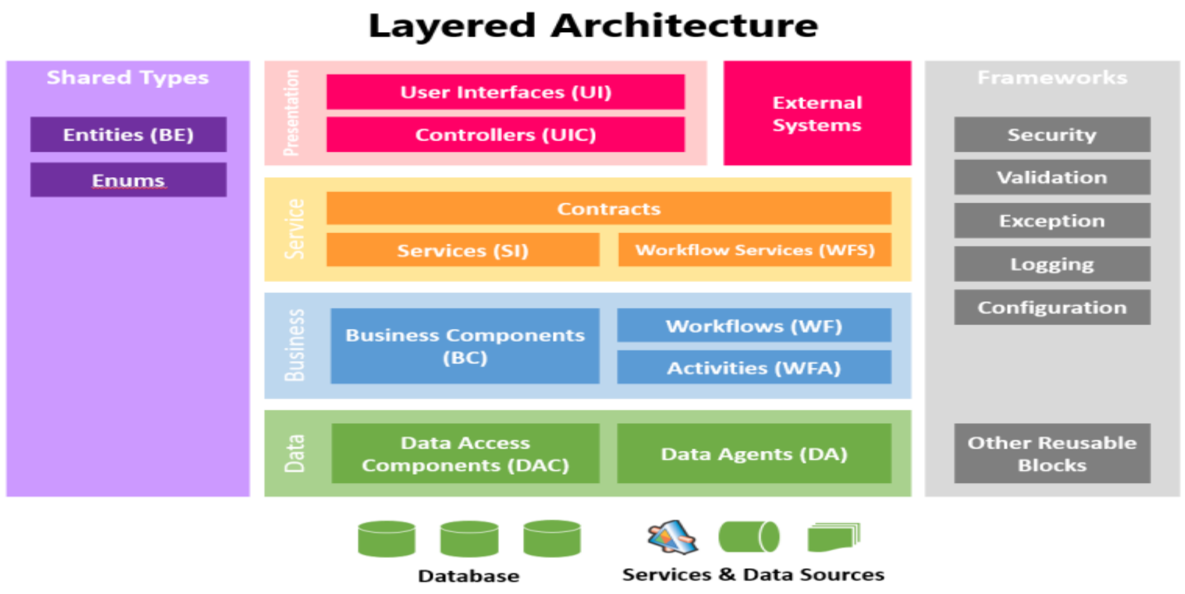


Fig 1.9 Layered architecture of cloud computing

1.9 DIFFERENT ROLES IN CLOUD COMPUTING

CLOUD PROVIDER

These are those entities that can verifiably and significantly produce the cloud computing services i.e. the can operates, manages and deliver various types of cloud services to the third parties.

USERS

Cloud users are those for whom the services are being offered. It can be a single user or an organization. Security and privacy are the major concern for the data of users.

VENDOR

It is lying in between the cloud providers and users. They are responsible for delivery and adoption of cloud services.

1.9 ISSUES IN CLOUD COMPUTING

- **JOB SCHEDULING:** - For better utilization of resources we must schedule the jobs in a proper manner. In this way we can increase the efficiency of computing power of cloud environment.

- **LOAD BALANCING:** - It is the appropriate distribution of tasks on the virtual resources so that very few resources left idle and also no one comes under overload condition.
- **LICENSE MANAGEMENT**: - It is an agreement between the client and the service providers prior to the delivery of the services to the client.
- **SCALABILITY:** - It is the property of the system which decides how well the system performed even when it has been scaled up and down.
- **ELASTICITY:** - It is the property that decides when and how the system would have been scaled up and down.
- **AVAILABILITY OF SERVICES:** - Cloud services should be available on time and the resources should be allocated as per demand by the users.
- **PERFORMANCE UNCERTAINTY DUE TO VIRTUALIZATION:** - Virtual machines are not easily deployable beforehand as the requirement is not very clear early from the user side.
- **INTEROPERABILITY:** - Cloud infrastructure uses the shared concept to carry out the things. Same tools and application are used across multiple cloud platforms. Hence its required a need of migrating in or out and switching to clouds.
- **PORTABILITY:** - It ensures that the work on one cloud platform can be portable to another cloud platform.

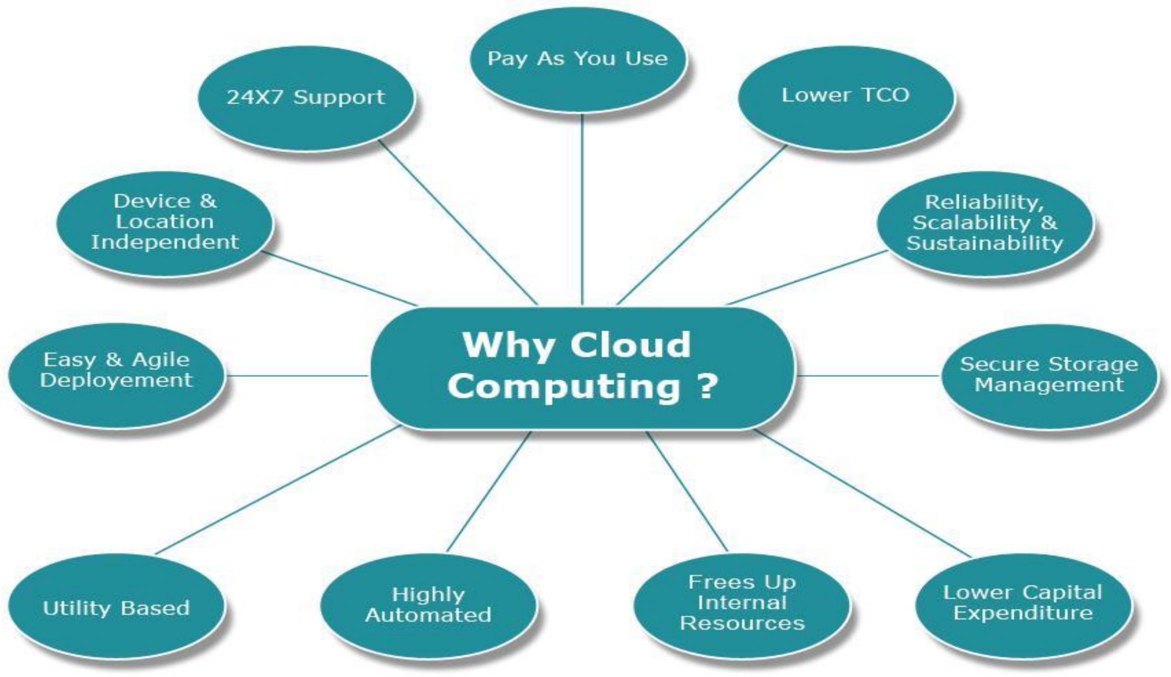

Pay As You Use
24X7 Support
Lower TCO
Device & Location Independent
Reliability, Scalability & Sustainability
Easy & Agile Deployement
Why Cloud Computing ?
Secure Storage Management
Utility Based
Highly Automated
Frees Up Internal Resources
Lower Capital Expenditure

Fig 1.10 Features of cloud computing

1.10 DISADVATAGES OF CLOUD COMPUTING

In spite of its many benefits, as mentioned above, Cloud computing also has its disadvantages. Businesses, especially smaller ones, need to be aware of these aspects before going in for this technology. The main risks involved in Cloud Computing are:

TECHNICAL ISSUES. Though it is true that information and data on the Cloud can be accessed any time and from anywhere, there are moments when the system can have some serious malfunction. Businesses should be aware of the fact that this technology is always prone to outages and other technical issues. Even the best Cloud service providers run into this kind of trouble, in spite of keeping up high standards of maintenance.

SECURITY IN THE CLOUD. The other major issue of Cloud is represented by security. Before adopting this technology, beneficiaries should know that they will be surrendering all their company's sensitive information to a third-party cloud service provider. This could potentially impose a great risk to the company. Hence, businesses need to make sure that they choose the most reliable service provider, who will keep their information totally secure. Switching to the cloud can actually improve security for a small business, as mentioned by Michael Redding, managing director of Accenture Technology Labs. "Because large cloud computing companies have more resources, he says, they are often able to offer levels of security an average small business may not be able to afford implementing on its own servers" (Outsource IT Headaches to the Cloud (The Globe and Mail)).

PRONE TO ATTACK. Storing information in the cloud could make the companies vulnerable to external hack attacks and threats; therefore there is always the lurking possibility of stealth of sensitive data.

POSSIBLE DOWNTIME. Cloud computing makes the small business dependent on the reliability of their Internet connection.

COST. At first glance, a cloud computing application may appear to be a lot cheaper than a particular software solution installed and run in-house. Still, the companies need to ensure that the cloud applications have all the features that the software does and if not, to identify which are the missing features important to them. A total cost comparison is also required. While many cloud computer vendors present themselves as utility-based providers, claiming that they only charge for what customers use, Gartner says that this isn't true; in most cases, a company must commit to a predetermined contract independent of actual use. Companies need to look closely at the pricing plans and details for each application.

INFLEXIBILITY. Choosing a Cloud computing vendor often means locking the business into using their proprietary applications or formats. For instance, it is not possible to insert a document created in another application into a Google Docs spreadsheet. Furthermore, a company needs to be able to add and/or subtract Cloud computing users as necessary as its business grows or contracts.

LACK OF SUPPORT. Anita Campbell (OPEN Forum) writes, "Customer service for Web apps leaves a lot to be desired - all too many cloud-based applications make it difficult to get customer service promptly – or at all. Sending an email and hoping for a response within 48 hours is not an acceptable way for most of us to run a business". The New York Times writes: "The bottom line: If you need handholding or if you are not comfortable trying to find advice on user forums, the cloud probably is not ideal " Thinking About Moving to the Cloud? There Are Trade-Offs.

1.11 CLOUDSIM

It is a simulation toolkit which is designed to simulate the cloud computing environment. This generalized framework enables seamless simulation, modeling and experimentation of new cloud computing application and infrastructure services. Cloudsim 3.03 is the latest release and it is built in JAVA. I use eclipse that is a JAVA IDE for experimentation. Cloudsim provides controllable and repeatable environment for the simulation of cloud. Cloudsim has various entities such datacenter, broker, hosts, virtual machines, cloudlet etc which is used to portrait real cloud entities.

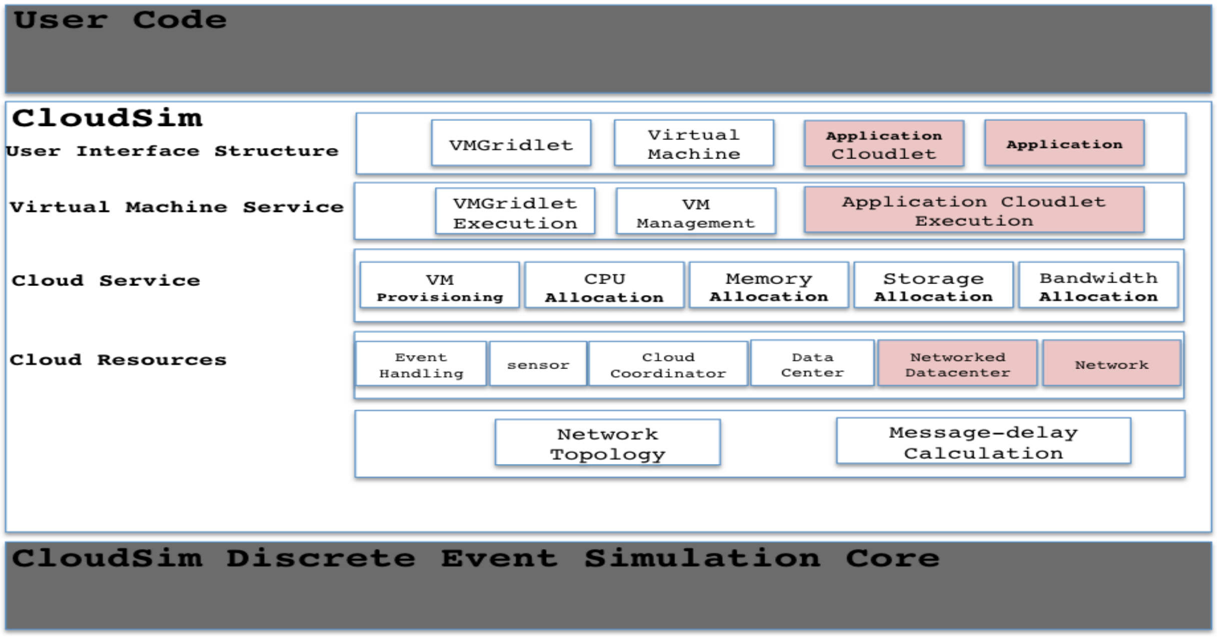


Fig 1.11 Cloudsim simulation core

1.12 CLOUDSIM ARCHITECTURE

Cloudsim has a layered architecture and that has 4 layers. These layers are USER level, User-level middle layer, Core layer and System level layer. USER level layer contains applications that are directly available to the end user. User-level layer provide programming environment and composition tools. It includes software framework to create rich and cost effective interfaces. Core level layer provide hosting and management environment for user level application services. Computing power in cloud environment has been provided by the System level layer. It includes the datacenters which in turn include thousand of hosts with each host has some physical resources like storage servers and application servers. Conservation of energy is an important issue in the datacenters. [22] Virtual machines workloads must be consolidated into lesser number of physical machines. Therefore computation workloads must be divided optimally for the purpose of energy reduction. Distribution policy of workloads is divided into two categories – Time shared and Space shared. In time shared policy all the tasks are assigned simultaneously to the virtual machines while in space shared policy one task at a time is given to a virtual machine.

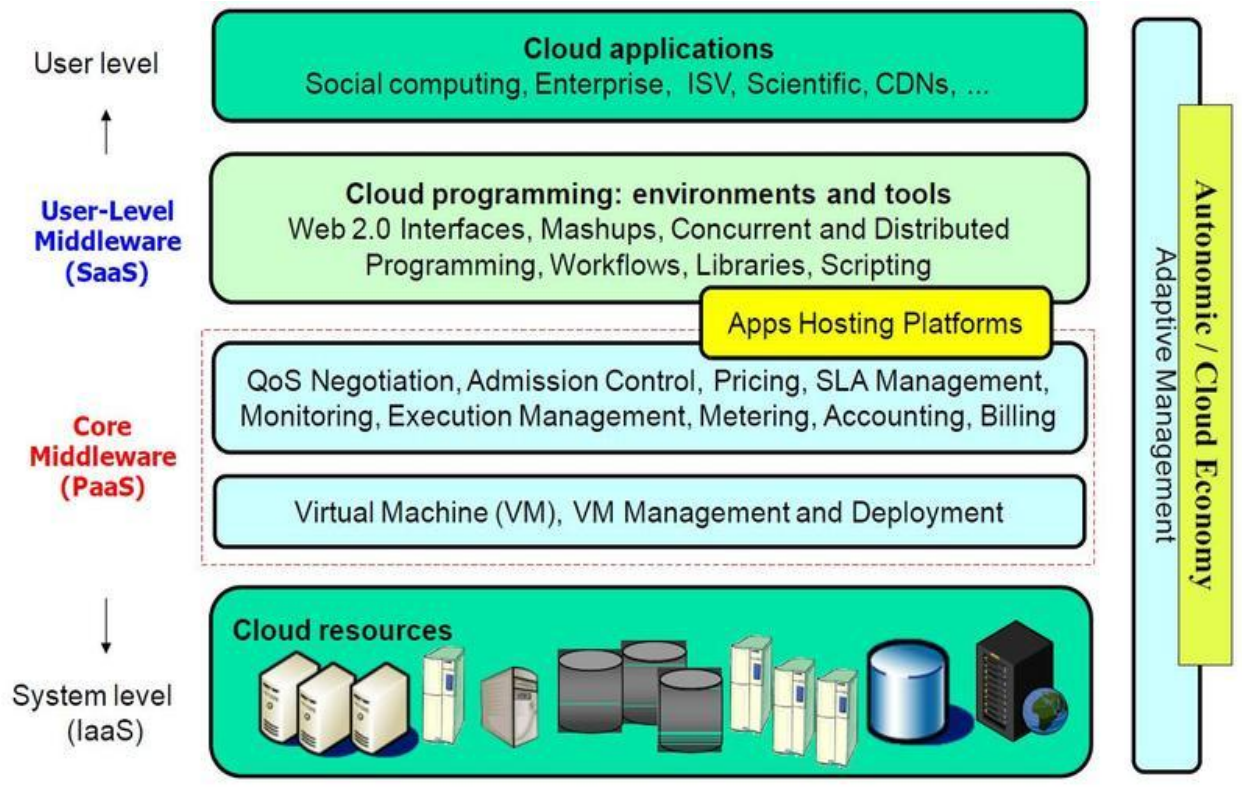

User level
Cloud applications
Social computing, Enterprise, ISV, Scientific, CDNs, ...
User-Level Middleware (SaaS)
Cloud programming: environments and tools
Web 2.0 Interfaces, Mashups, Concurrent and Distributed Programming, Workflows, Libraries, Scripting
Apps Hosting Platforms
Core Middleware (PaaS)
QoS Negotiation, Admission Control, Pricing, SLA Management, Monitoring, Execution Management, Metering, Accounting, Billing
Virtual Machine (VM), VM Management and Deployment
Cloud resources
System level (IaaS)
Autonomic / Cloud Economy
Adaptive Management

Fig 1.12 Layered architecture of Cloudsim

1.13 CLASS IN CLOUDSIM

The overall Class design diagram for CloudSim is shown in Figure 1.13.

BwProvisioner: This is an abstract class that models the policy for provisioning of bandwidth to VMs. The main role of this component is to undertake the allocation of network bandwidths to a set of competing VMs that are deployed across the data center. Cloud system developers and researchers can extend this class with their own policies (priority, QoS) to reflect the needs of their applications. The *BwProvisioningSimple* allows a VM to reserve as much bandwidth as required; however, this is constrained by the total available bandwidth of the host.

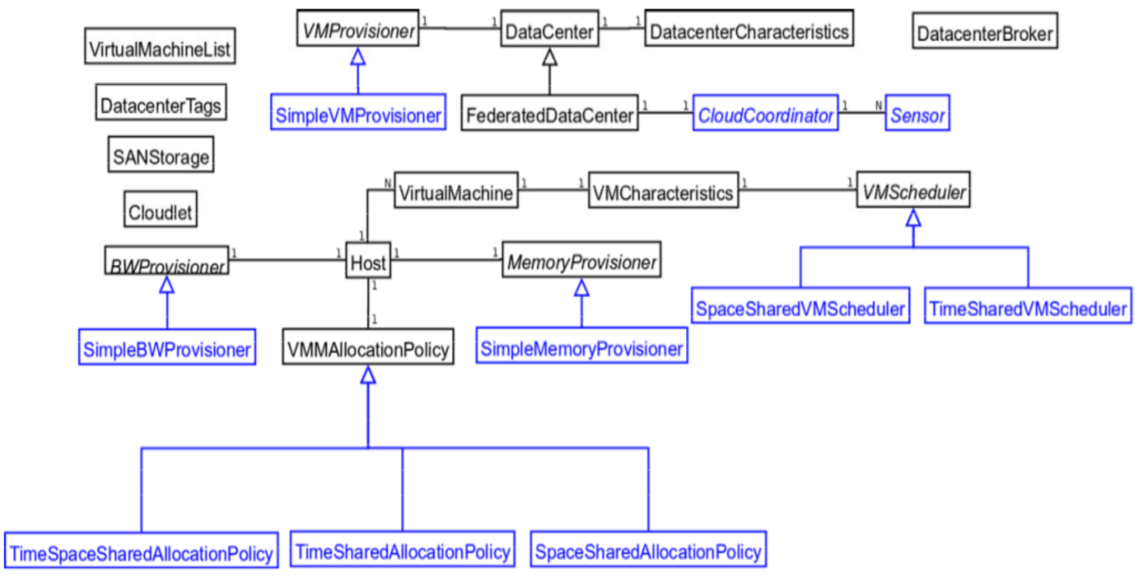


Fig 1.13 Class diagram of Cloudsim

CloudCoordinator: This abstract class extends a Cloud-based data center to the federation. It is responsible for periodically monitoring the internal state of data center resources and based on that it undertakes dynamic load-shredding decisions. Concrete implementation of this component includes the specific sensors and the policy that should be followed during load-shredding. Monitoring of data center resources is performed by the *updateDatacenter()* method by sending queries Sensors. Service/Resource Discovery is realized in the *setDatacenter()*abstract method that can be extended for implementing custom protocols and mechanisms (multicast, broadcast, peer-to-peer). Further, this component can also be extended for simulating Cloud-based services such as the Amazon EC2 Load-Balancer. Developers aiming to deploy their application services across multiple clouds can extend this class for implementing their custom inter-cloud provisioning policies.

Cloudlet: This class models the Cloud-based application services (such as content delivery, social networking, and business workflow). CloudSim orchestrates the complexity of an application in terms of its computational requirements. Every application service has a pre-assigned instruction length and data transfer (both pre and post fetches) overhead that it needs to undertake during its life cycle. This class can also be extended to support modeling of other performance and composition metrics for applications such as transactions in database-oriented applications.

CloudletScheduler: This abstract class is extended by the implementation of different policies that determine the share of processing power among Cloudlets in a VM. As described

previously, two types of provisioning policies are offered: space-shared (*CloudetSchedulerSpaceShared*) and time-shared (*CloudletSchedulerTimeShared*).

Datacenter: This class models the core infrastructure-level services (hardware) that are offered by Cloud providers (Amazon, Azure, App Engine). It encapsulates a set of compute hosts that can either be homogeneous or heterogeneous with respect to their hardware configurations (memory, cores, capacity, and storage). Furthermore, every Datacenter component instantiates a generalized application provisioning component that implements a set of policies for allocating bandwidth, memory, and storage devices to hosts and VMs.

DatacenterBroker or Cloud Broker: This class models a broker, which is responsible for mediating negotiations between SaaS and Cloud providers; and such negotiations are driven by QoS requirements. The broker acts on behalf of SaaS providers. It discovers suitable Cloud service providers by querying the CIS and undertakes online negotiations for allocation of resources/services that can meet the application's QoS needs. Researchers and system developers must extend this class for evaluating and testing custom brokering policies. The difference between the broker and the CloudCoordinator is that the former represents the customer (i.e. decisions of these components are made in order to increase user-related performance metrics), whereas the latter acts on behalf of the data center, i.e. it tries to maximize the overall performance of the data center, without considering the needs of specific customers.

DatacenterCharacteristics: This class contains configuration information of data center resources.

Host: This class models a physical resource such as a compute or storage server. It encapsulates important information such as the amount of memory and storage, a list and type of processing cores (to represent a multi-core machine), an allocation of policy for sharing the processing power among VMs, and policies for provisioning memory and bandwidth to the VMs.

NetworkTopology: This class contains the information for inducing network behavior (latencies) in the simulation. It stores the topology information, which is generated using the BRITE topology generator.

RamProvisioner: This is an abstract class that represents the provisioning policy for allocating primary memory (RAM) to the VMs. The execution and deployment of VM on a host is feasible only if the RamProvisioner component approves that the host has the required amount of free memory. The *RamProvisionerSimple* does not enforce any limitation on the amount of memory that a VM may request. However, if the request is beyond the available memory capacity, then it is simply rejected.

SanStorage: This class models a storage area network that is commonly ambient in Cloud-based data centers for storing large chunks of data (such as Amazon S3, Azure blob storage).

SanStorage implements a simple interface that can be used to simulate storage and retrieval of any amount of data, subject to the availability of network bandwidth. Accessing files in a SAN at run-time incurs additional delays for task unit execution; this is due to the additional latencies that are incurred in transferring the data files through the data center internal network.
Sensor: This interface must be implemented to instantiate a sensor component that can be used by a CloudCoordinator for monitoring specific performance parameters (energy-consumption, resource utilization). Recall that, CloudCoordinator utilizes the dynamic performance information for undertaking load-balancing decisions. The methods defined by this interface are: (i) set the minimum and maximum thresholds for performance parameter and (ii) periodically update the measurement. This class can be used to model the real-world services offered by leading Cloud providers such as Amazon's CloudWatch and Microsoft Azure's FabricController. One data center may instantiate one or more Sensors, each one responsible for monitoring a specific data center performance parameter.

Vm: This class models a VM, which is managed and hosted by a Cloud host component. Every VM component has access to a component that stores the following characteristics related to a VM: accessible memory, processor, storage size, and the VM's internal provisioning policy that is extended from an abstract component called the *CloudletScheduler*.

VmmAllocationPolicy: This abstract class represents a provisioning policy that a VM Monitor utilizes for allocating VMs to hosts. The chief functionality of the VmmAllocationPolicy is to select the available host in a data center that meets the memory, storage, and availability requirement for a VM deployment.

VmScheduler: This is an abstract class implemented by a Host component that models the policies (space-shared, time-shared) required for allocating processor cores to VMs. The functionalities of this class can easily be overridden to accommodate application-specific processor sharing policies.

1.14. FEDERATION (INTER-NETWORKING) OF CLOUDS

Current Cloud computing providers have several data centers at different geographical locations over the Internet in order to optimally serve customer needs around the world. However, the existing systems do not support mechanisms and policies for dynamically coordinating load shredding among different data centers in order to determine the optimal location for hosting application services to achieve reasonable QoS levels. Further, the Cloud service providers are unable to predict the geographic distribution of end-users consuming their services; hence, the load coordination must happen automatically, and distribution of services must change in response to changes in the load behavior. Figure 1.14 depicts such a Cloud computing architecture that consists of service consumers' (SaaS providers') brokering and providers' coordinator services that support utility-driven internetworking of clouds:

Application provisioning and workload migration. Federated inter-networking of administratively distributed clouds offers significant performance and financial benefits such as: (i) improving the ability of SaaS providers in meeting QoS levels for clients and offer improved service by optimizing the service placement and scale; (ii) enhancing the peak-load handling and dynamic system expansion capacity of every member cloud by allowing them to dynamically acquire additional resources from federation. This frees the Cloud providers from the need of setting up a new data center in every location; and (iii) adapting to failures, such as natural disasters and regular system maintenance, is more graceful as providers can transparently migrate their services to other domains in the federation, thus avoiding SLA violations and the resulting penalties. Hence, federation of clouds not only ensures business continuity but also augments the reliability of the participating Cloud providers.

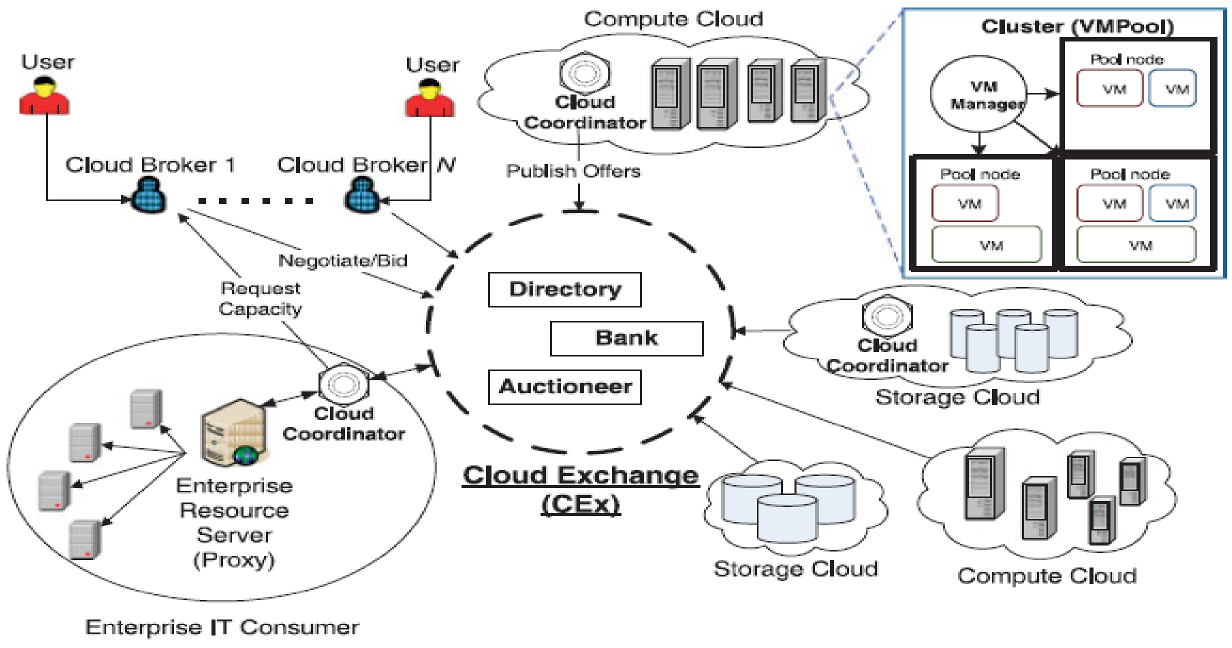


Fig 1.14 Clouds and their federated network

One of the key components of the architecture presented in Figure 1.14 is the Cloud Coordinator. This component is instantiated by each cloud in the system whose responsibility is

to undertake the following important activities: (i) exporting Cloud services, both infrastructure and platform-level, to the federation; (ii) keeping track of load on the Cloud resources (VMs, computing services) and undertaking negotiation with other Cloud providers in the federation for handling the sudden peak in resource demand at local cloud; and (iii) monitoring the application execution over its life cycle and overseeing that the agreed SLAs are delivered. The Cloud brokers acting on behalf of SaaS providers identify suitable Cloud service providers through the Cloud Exchange (CEx).

Further, Cloud brokers can also negotiate with the respective Cloud Coordinators for allocation of resources that meets the QoS needs of hosted or to be hosted SaaS applications. The CEx acts as a market maker by bringing together Cloud service (IaaS) and SaaS providers. CEx aggregates the infrastructure demands from the Cloud brokers and evaluates them against the available supply currently published by the Cloud Coordinators.

The applications that may benefit from the aforementioned federated Cloud computing infrastructure include social networks such as Facebook and MySpace, and Content-Delivery Networks (CDNs). Social networking sites serve dynamic contents to millions of users, whose access and interaction patterns are difficult to predict. In general, social networking web sites are built using multi-tiered web applications such as WebSphere and persistency layers like the MySQL relational database. Usually, each component will run on a different VM, which can be hosted in data centers owned by different Cloud computing providers. Additionally, each plug-in developer has the freedom to choose which Cloud computing provider offers the services that are more suitable to run his/her plug-in. As a consequence, a typical social networking web application is formed by hundreds of different services, which may be hosted by dozens of Cloud-oriented data centers around the world. Whenever there is a variation in the temporal and spatial locality of workload (usage pattern), each application component must dynamically scale to offer good quality of experience to users.

Domain experts and scientists can also take advantage of such mechanisms by using the cloud to leverage resources for their high-throughput e-Science applications, such as Monte–Carlo simulation and Medical Image Registration. In this scenario, the clouds can be augmented to the existing cluster and grid-based resource pool to meet research deadlines and milestones.

1.15 OTHER IMPORTANT ASPECTS IN CLOUD COMPUTING

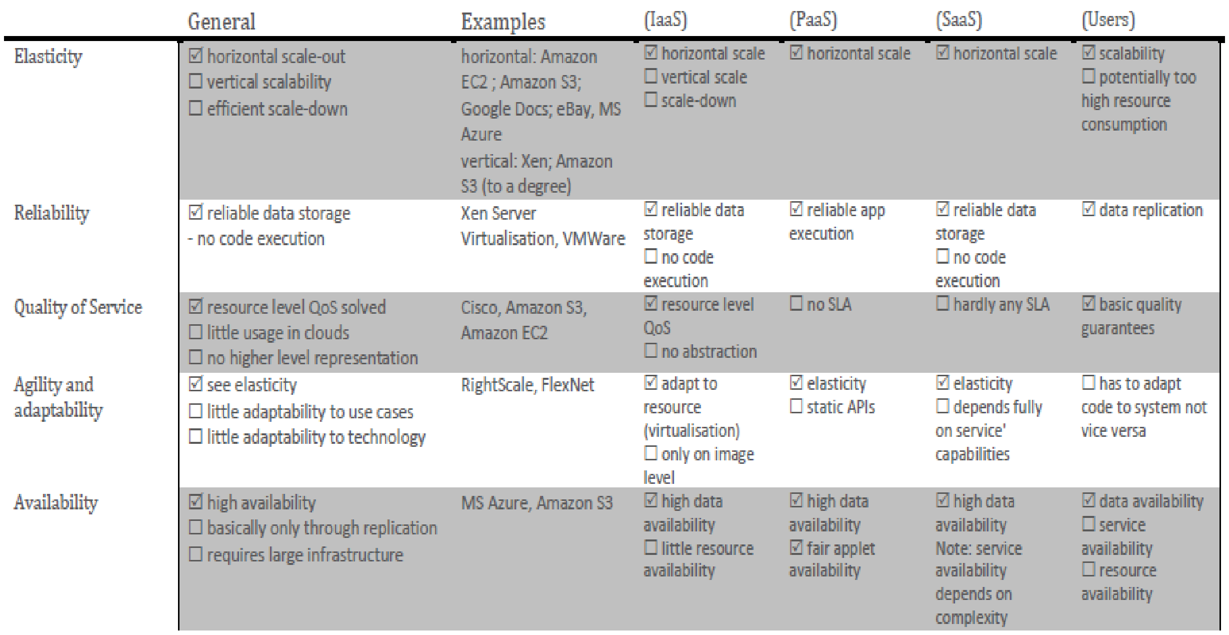
Why organizations are interested in cloud computing?

Cloud computing can significantly reduce the cost and complexity of owning and operating computers and networks. If an organization uses a cloud provider, it does not need to spend money on information technology infrastructure, or buy hardware or software licenses. Cloud services can often be customized and flexible to use, and providers can offer advanced services that an individual company might not have the money or expertise to develop.

CLOUDS IN THE FUTURE INTERNET

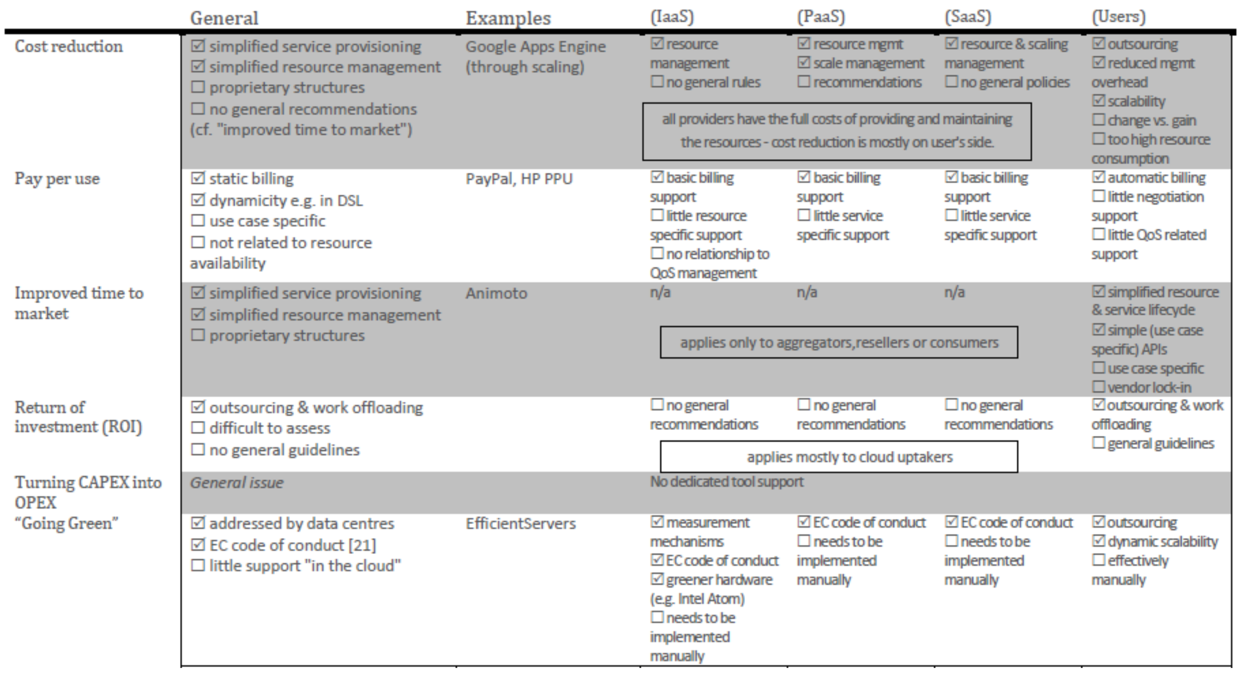
The Future Internet covers all research and development activities dedicated to realizing tomorrow's internet, i.e. enhancing a networking infrastructure which integrates all kind of resources, usage domains etc. As such, research related to cloud technologies form a vital part of the Future Internet research agenda. Confusions regarding the aspects covered by cloud computing with respect to the Future Internet mostly arise from the broad scope of characteristics assigned to "clouds", as is the logical consequence of the re-branding boom some years ago. So far, most cloud systems have focused on hosting applications and data on remote computers, employing in particular replication strategies to ensure availability and thus achieving a load balancing scalability. However, the conceptual model of clouds exceeds such a simple technical approach and leads to challenges not unlike the ones of the future internet, yet with slightly different focus due to the combination of concepts and goals implicit to cloud systems. In other words, as a technological realization driven by an economic proposition, cloud infrastructures would offer capabilities that enable relevant aspects of the future internet, in particular related to scalability, reliability and adaptability. At the same time, the cloud concept addresses multiple facets of these functionalities.

Non-technical aspects of cloud computing



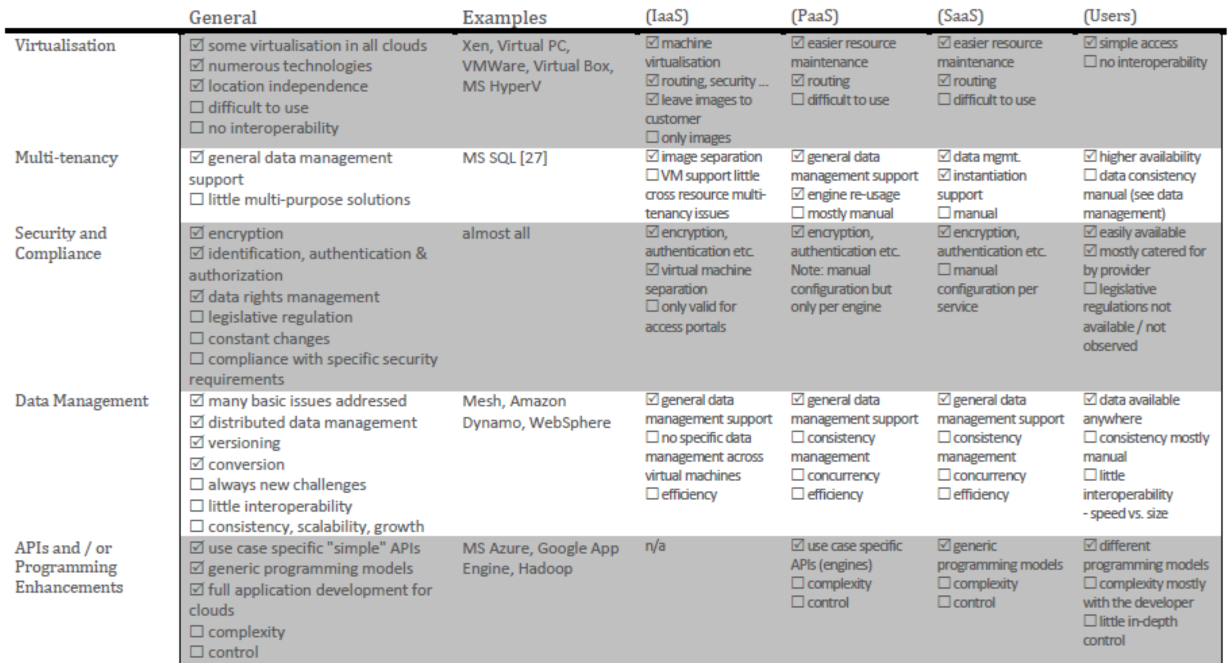
	General	Examples	(IaaS)	(PaaS)	(SaaS)	(Users)
Elasticity	☑ horizontal scale-out ☐ vertical scalability ☐ efficient scale-down	horizontal: Amazon EC2 ; Amazon S3; Google Docs; eBay, MS Azure vertical: Xen; Amazon S3 (to a degree)	☑ horizontal scale ☐ vertical scale ☐ scale-down	☑ horizontal scale	☑ horizontal scale	☑ scalability ☐ potentially too high resource consumption
Reliability	☑ reliable data storage - no code execution	Xen Server Virtualisation, VMWare	☑ reliable data storage ☐ no code execution	☑ reliable app execution	☑ reliable data storage ☐ no code execution	☑ data replication
Quality of Service	☑ resource level QoS solved ☐ little usage in clouds ☐ no higher level representation	Cisco, Amazon S3, Amazon EC2	☑ resource level QoS ☐ no abstraction	☐ no SLA	☐ hardly any SLA	☑ basic quality guarantees
Agility and adaptability	☑ see elasticity ☐ little adaptability to use cases ☐ little adaptability to technology	RightScale, FlexNet	☑ adapt to resource (virtualisation) ☐ only on image level	☑ elasticity ☐ static APIs	☑ elasticity ☐ depends fully on service' capabilities	☐ has to adapt code to system not vice versa
Availability	☑ high availability ☐ basically only through replication ☐ requires large infrastructure	MS Azure, Amazon S3	☑ high data availability ☐ little resource availability	☑ high data availability ☑ fair applet availability	☑ high data availability Note: service availability depends on complexity	☑ data availability ☐ service availability ☐ resource availability

Economic aspects of cloud computing



	General	Examples	(IaaS)	(PaaS)	(SaaS)	(Users)
Cost reduction	☑ simplified service provisioning ☑ simplified resource management ☐ proprietary structures ☐ no general recommendations (cf. "improved time to market")	Google Apps Engine (through scaling)	☑ resource management ☐ no general rules all providers have the full costs of providing and maintaining the resources - cost reduction is mostly on user's side.	☑ resource mgmt ☑ scale management ☐ recommendations	☑ resource & scaling management ☐ no general policies	☑ outsourcing ☑ reduced mgmt overhead ☑ scalability ☐ change vs. gain ☐ too high resource consumption
Pay per use	☑ static billing ☑ dynamicity e.g. in DSL ☐ use case specific ☐ not related to resource availability	PayPal, HP PPU	☑ basic billing support ☐ little resource specific support ☐ no relationship to QoS management	☑ basic billing support ☐ little service specific support	☑ basic billing support ☐ little service specific support	☑ automatic billing ☐ little negotiation support ☐ little QoS related support
Improved time to market	☑ simplified service provisioning ☑ simplified resource management ☐ proprietary structures	Animoto	n/a applies only to aggregators,resellers or consumers	n/a	n/a	☑ simplified resource & service lifecycle ☑ simple (use case specific) APIs ☐ use case specific ☐ vendor lock-in
Return of investment (ROI)	☑ outsourcing & work offloading ☐ difficult to assess ☐ no general guidelines		☐ no general recommendations applies mostly to cloud uptakers	☐ no general recommendations	☐ no general recommendations	☑ outsourcing & work offloading ☐ general guidelines
Turning CAPEX into OPEX	*General issue*		No dedicated tool support			
"Going Green"	☑ addressed by data centres ☑ EC code of conduct [21] ☐ little support "in the cloud"	EfficientServers	☑ measurement mechanisms ☑ EC code of conduct ☑ greener hardware (e.g. Intel Atom) ☐ needs to be implemented manually	☑ EC code of conduct ☐ needs to be implemented manually	☑ EC code of conduct ☐ needs to be implemented manually	☑ outsourcing ☑ dynamic scalability ☐ effectively manually

Technological aspects of cloud computing



	General	Examples	(IaaS)	(PaaS)	(SaaS)	(Users)
Virtualisation	☑ some virtualisation in all clouds ☑ numerous technologies ☑ location independence ☐ difficult to use ☐ no interoperability	Xen, Virtual PC, VMWare, Virtual Box, MS HyperV	☑ machine virtualisation ☑ routing, security ... ☑ leave images to customer ☐ only images	☑ easier resource maintenance ☑ routing ☐ difficult to use	☑ easier resource maintenance ☑ routing ☐ difficult to use	☑ simple access ☐ no interoperability
Multi-tenancy	☑ general data management support ☐ little multi-purpose solutions	MS SQL [27]	☑ image separation ☐ VM support little cross resource multi-tenancy issues	☑ general data management support ☑ engine re-usage ☐ mostly manual	☑ data mgmt. ☑ instantiation support ☐ manual	☑ higher availability ☐ data consistency manual (see data management)
Security and Compliance	☑ encryption ☑ identification, authentication & authorization ☑ data rights management ☐ legislative regulation ☐ constant changes ☐ compliance with specific security requirements	almost all	☑ encryption, authentication etc. ☑ virtual machine separation ☐ only valid for access portals	☑ encryption, authentication etc. Note: manual configuration but only per engine	☑ encryption, authentication etc. ☐ manual configuration per service	☑ easily available ☑ mostly catered for by provider ☐ legislative regulations not available / not observed
Data Management	☑ many basic issues addressed ☑ distributed data management ☑ versioning ☑ conversion ☐ always new challenges ☐ little interoperability ☐ consistency, scalability, growth	Mesh, Amazon Dynamo, WebSphere	☑ general data management support ☐ no specific data management across virtual machines ☐ efficiency	☑ general data management support ☐ consistency management ☐ concurrency ☐ efficiency	☑ general data management support ☐ consistency management ☐ concurrency ☐ efficiency	☑ data available anywhere ☐ consistency mostly manual ☐ little interoperability - speed vs. size
APIs and / or Programming Enhancements	☑ use case specific "simple" APIs ☑ generic programming models ☑ full application development for clouds ☐ complexity ☐ control	MS Azure, Google App Engine, Hadoop	n/a	☑ use case specific APIs (engines) ☐ complexity ☐ control	☑ generic programming models ☐ complexity ☐ control	☑ different programming models ☐ complexity mostly with the developer ☐ little in-depth control

APIs, Programming Models & Resource Control

Cloud virtual machines tend to be built for fixed resource environments, thus allowing horizontal scalability (instance replication) better than vertical scalability (changes in the resource structure) – however, future systems will have to show more flexibility with this respect to adapt better to requirements, capabilities and of course green issues. In addition, more fine grained control over e.g. distribution of data etc. must be granted to the developer in order to address legislation issues, but also to exploit specific code requirements. Cloud systems will thus face similar issues that HPC has faced before with respect to description of connectivity requirements etc., but also to ensure reliability of execution, which is still a major obstacle in distributed systems. At the same time, the model must be simple enough to be employed by average developers and / or business users. Cloud systems provide enhanced capabilities and features, ranging from dynamically scalable applications and data, over controlled distribution to integration of all types of resources (including humans). In order to exploit these features during development of enhanced applications and services, the according interfaces and features need to be provided in an easy and intuitive fashion for common users, but should also allow for extended control for more advanced users. In order to facilitate such enhanced control features, the cloud system needs to provide new means to manage resources and infrastructure, potentially taking quality of service, the green agenda and other customer specifications into consideration. This, however, implies that future cloud systems have to discard the classical layered model (see also [29]). Development support for new "cloudified" applications has to ensure movability of application (segments) across the network, enabling a more distributed execution and communication model within and between applications. Since cloud applications are likely to be used by much more tenants and users than non-cloud applications ("long tail"), customizability must be considered from the outset. The issue applies equally to distributed code, as to distributed data. Data is expected to become exceedingly large (see "Data Management" above) - hence an interesting approach in cloud system's code management consists in moving the software to the data, rather than the other way round, since most code occupies less space than the data they process. However this is intrinsically against the current trend for clouds to be provided in remote data centers with code and data co-existing. Main issues: connectivity; intelligent distribution (code & data); multi-tenancy; enhanced manageability; reliability; ease of use; development and deployment support.

Economic Concerns

In order to provide a cloud infrastructure, a comparatively high amount of resources needs to be available, which implies a considerable high investment for start-up. As it is almost impossible to estimate the uptake and hence the profit of services offered to the customers, it remains difficult to assess the return of investment and hence the sensible amount of investment to maximize the profit. With the cloud outsourcing principle being comparatively new on the market, new knowledge about business models, market situation, how to extract value and under what conditions etc. are required – in other words, new expert systems and best use recommendations are required. This also includes issues related to the "green agenda", namely policies basing on dedicated benchmarks under what circumstances to reduce resource usage and / or switch between different power settings etc. This implies new scheduling mechanisms that weigh green vs. business (profit & quality) issues. In a cloud environment it would be possible to improve 'green' credentials by utilizing more efficient processors and memory. A few large data centers with clouds are likely to be more 'green' than millions of smaller but already large data centers. Fan et al. argued that up to 50% savings in energy consumption are possible for data warehouses. Notably, from a global perspective, sharing resources may be greener than down-powering idle resources, if this reduces their production (and hence the according carbon footprint) in the first instance. In general, business control is principally possible, yet linkage between the technical and economical perspective is still weak and hence maintenance of e.g. service quality respecting the economical descriptions still requires improvement. An indirect economical issue that will have to be solved through e.g., means for improved interoperability (see below), consists in the current tendency towards vendor-lock in. Most vendors want to maintain this status in order to secure their customer base, yet with scope and competition growing in the near future, it is to be expected that even larger vendors will adopt more interoperable approaches. As a side note it should be mentioned that already some major key player are basing their system on more standard based approaches, such as MS Azure.

CLOUD COMPUTING OPERATIONS

Cloud Computing operation refers to delivering superior cloud service. Today, cloud computing operations have become very popular and widely employed by many of the organizations just because it allows to perform all business operations over the Internet. These operations can be performed using a web application or mobile based applications. There are a number of operations that are performed in cloud, some of them are :

- Email marketing
- Content management
- System wide reporting
- Online marketplace
- Accounting services
- Human resource management and recruitment services

MANAGING CLOUD OPERATIONS

There are several ways to manage day-to-day cloud operations, as shown in the following diagram:

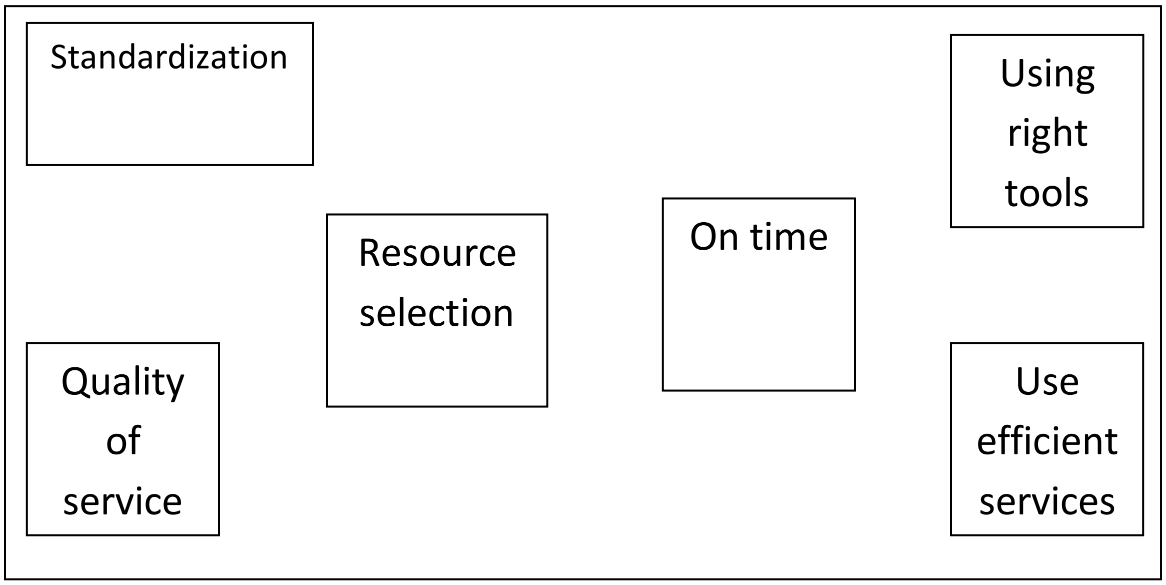


Fig. 1. Managing operations

- Always employ right tools and resources to perform any function in the cloud.
- Things should be done at right time and at right cost.
- Selecting an appropriate resource is mandatory for operation management.
- The process should be standardized and automated to avoid repetitive tasks.
- Using efficient process will eliminate the waste and redundancy.
- One should maintain the quality of service to avoid re-work later.

1.16 SUMMARY

In this chapter, we first discussed various computing models then we explain how cloud computing comes into picture and how it is better than other technologies. We also tried to explain the relationship between virtualization and cloud computing and from that we inferred that virtualization is the key component of cloud computing. After that we explain various types of clouds and the layered architecture of cloud. In last we discuss various issues of cloud computing which are necessary to be resolved and gave an overview about the cloudim framework.

Chapter 2

TASK SCHEDULING

2.1 INTRODUCTION

Task scheduling is one of the critical issues in cloud computing environment where many tasks are rivalry together for getting execution. For this purpose various tasks scheduling algorithms are designed so that each task gets its fair chance for execution. Cloud computing environment is a distributed environment in which resources are distributed among many host. Therefore, for proper utilization of resources, increased throughput, and reduced power are the major factors that must be kept in mind while designing a task scheduling algorithm. [19]Job scheduling problem is a combinatorial in nature in the computer science field where ideal jobs are allocated to available resource at a particular instance of time. Total execution time and make-span time is the time required for the complete execution of a set of jobs.

Task scheduling algorithm based on their nature can be classified into two groups.

1. STATIC TASK SCHEDULING ALGORITHMS: - In this category algorithms are designed keeping in mind the fact that the information regarding task execution time, their data dependencies and synchronization requirements are known prior their execution. Pipelining concept can be used in this case to increase the efficiency.

2. DYNAMIC TASK SCHEDULING ALGORITHMS: - In this category, prior knowledge of task execution time is not known therefore we have to make few assumptions before execution and thus on-the-fly decisions has to be made. Therefore these scheduling algorithms has two goals – first is reduction of make-span time and secondly, reduction of time caused by running scheduler in between program execution.

Dynamic scheduling algorithms outrun static scheduling algorithms in case of performance. [4] Varieties of scheduling algorithms are used in distributed systems. Throughput and performance can be the good factor in choosing a task scheduling algorithm. High values of these parameters means good scheduling.

In cloud computing environment, a task scheduling algorithm can be applied on different levels. It can be applied at broker level where a broker decides which task is given to which virtual machine or it can be applied at virtual machine in which a virtual machinedecides on which host a task will run or it can applied at host which schedule tasks on physical resources.

In our scheduling algorithm, policy is applied at broker level i.e. between tasks and virtual machines.

2.2 TYPES OF SCHEDULING

Tasks scheduling algorithms can be divides into two groups based on mode:

2.2.1 BATCH MODE HEURISTIC SCHEDULING ALGORITHMS (BHMA)

In this mode, first tasks are collected over a predetermined amount of time into a buffer and then they are allocated to machines based on scheduling policy. It can be applied on both homogenous as well as heterogeneous computer systems. MIN-MIN, MIN-MEAN, MAX-MIN, FIRST COME FIRST SERVED ROUND ROBIN, PRIORITY BASED algorithms come under this category.

2.2.2 ONLINE MODE HEURISTIC SCHEDULING ALGORITHMS (OHMA)

In this mode, a task is mapped to a machine as it arrives at the scheduler. No buffer is maintained in this case. Typical cloud computing environment has this kind of scenario in which a task can come at any time which require urgent handling. Therefore online mode is well suited for cloud computing. Most FIT SCHEDULING algorithm is the typical example of this type of category.

2.3 PROCESS OF SCHEDULING

Scheduling process is categorized into following phases:

DISCOVERY AND FILTERING OF RESOURCES: - Each datacenter is registered with a datacenter broker and transfer all the resource information to the broker. On arrival of tasks to the broker site, broker first gets the status of the requested resource from the datacenter and then filters out the resource that is free and in active state.

RESOURCE SELECTION: - Based on the resource availability and scheduling policy resources are selected from the resource pool and assigned to the tasks.

SUBMISSION OF TASKS: - After mapping of tasks, tasks are sent to their respective allocated resource.

2.4 VARIOUS TYPES OF TASKS SCHEDULING ALGORITHMS

FIRST COME FIRST SERVE:

As the name implies, the task which arrives first gets the resource first. Initially, all the tasks are collected in a buffer. Their arrival times are also stored and as the resource becomes available they are submitted to it based on who arrives first. Advantage of this algorithm is that it is simple to implement and fast result. Main drawback of this algorithm is urgent tasks has to wait for a longer period of time and also less utilization of resources.

ROUND ROBIN SCHEDULING ALGORITHM:

In this algorithm, time is sliced out into fixed intervals known as time-slice or quantum. All the jobs are stored in a buffer and are served in a FIFO order but for a specific amount of time. If the time quantum expires then the current job is preempted and store in a waiting buffer and the resource is allocated to next task. Advantage of round robin is that response time of the system decreases.

PRIORITY BASED SCHEDULING

In it priority is first assigned to the tasks. Priority determination is based on some pre determined parameters. Arrived tasks are stored in a priority queue along with their priorities. Task with highest priority has been taken out from the queue and submitted to the resource. Starvation is the major disadvantage of priority based scheduling in which a lower priority task may never get a chance for the resource.

MIN-MIN SCHEDULING ALGORITHM:

In this algorithm, a task with minimum amount of time required of a resource, mainly processor is scheduled to a processor with minimum processing power. In this way smaller jobs are scheduled first, therefore there is a possibility for starvation for longer jobs.

MAX-MIN SCHEDULING ALGORITHM

In this algorithm, a task with max amount of time required of a resource (CPU) is scheduled to a processor with minimum amount of processing power. In this way longer jobs are scheduled first, therefore there is a possibility for starvation for smaller jobs.

MOST FIT TASK SCHEDULING ALGORITHM:

In this algorithm, a task with high fitness value is selected from the task pool and submitted to the required resource. Experimentally it has been shown that make-span is high in this case.

2.4 SUMMARY

In this chapter, first we introduce the concept of scheduling and then define different categories of scheduling algorithm based on mode and nature of them. Then we describe the process of scheduling and how it applies on cloud computing environment. After that there are some different types of scheduling algorithms are given.

Chapter 3

RELATED WORK

Cloud computing is known as heterogeneous systems as it holds different types and large amount of data. Recently it is being a emerging technology requires to improve the resource utilization, reducing processing cost, better performance of the server, minimizing response as well as response time and for that it is very essential to schedule the tasks in the cloud and for that various scheduling algorithms are being designed. Various authors have proposed the solution to this scheduling problem and some of them have been already applied in the cloud environment.

Scheduling of tasks is a NP-complete problem so many static as well as dynamic algorithms require a lot of computation to produce a schedule which may be may not be optimal. Initially first come first served algorithm had been used because of its simplicity but due the diverse nature of cloud and viewing the on demand requirements of the users many different scheduling algorithms are required to be produced. [9]RASA is one of the scheduling algorithms which is a combination of two algorithms that are MIN-MIN and MIN-MAX algorithm. It takes their benefits and get rid of their shortcomings. The main drawback of this algorithm is that it does not consider many important parameters like priority of the jobs, their arrival rate and the cost of task execution on a particular resource. For distributed system, it has been seen that it provide better result than any other scheduling algorithm.

[8] In RSDC, a task is divided into sub tasks. To balance the tasks over resources, their request and acknowledge times have been calculated. Based on these times, shared resources are allocated due to which efficiency of the system increases. [10] Priority based scheduling algorithm is the algorithm designed to schedule tasks based on their quality of service requirements. Tolerance delay and the service cost are the parameters which are used to calculate the priority of the user tasks. User tasks are buffered in a queue on arrival and are sorted according to their priority. Now they are scheduled to their allotted resource when their fair chance comes. This algorithms guarantees high quality of service and achieves high completion rate.

[14]Improved cost based scheduling algorithm creates a schedule which is very efficient. In this three separate lists are prepared using priority of tasks; cost of computation of these tasks over resources and communication overhead required for their execution. After that, tasks are mapped to the available resource. The round robin algorithm is static algorithm that makes all scheduling algorithm before start of it. The major advantage of round robin is that it decreases the response time of tasks efficiently. But it has an extra overhead of context switching which increases the execution time of longer jobs to extend and thus waste necessary time of the resource.

The FPLTF scheduling algorithms choose the longest tasks from the task pool and assign it to the fastest processor. Longest, here means is the task that requires longest time with the resource. Resource is mainly the processor. Work queue algorithm works on the fact that more number of tasks will be assigned to the processor which has the maximum power. In this way completion time or make-span time can be reduced drastically.

As scheduling algorithms is a combinatorial NP complete problem, many optimization algorithms has been introduced. GA(Genetic algorithm) is one of the optimization techniques, which is based on the theory of natural selection. In GA, initially a pool of chromosome which is a pool possible solution to the scheduling algorithm has been found out. In the next step, two parents are taken out from the pool and called them parents. These two parents are having the high fitness value. These parents are crossed over with each other to produce siblings. In the next step, these siblings are mutated according to mutation probability theory, and after that a new solution has been produced. GA uses selection, cross over and mutation to remove inferior individual to choose the optimal schedule. It is an adaptive technique used globally and widely accepted. The main problem with GA is the convergence of the solution. Initial task and resource mapping may result in longer convergence and more number of iterations.

The author of paper [] describe the nature of ants and use them to solve the scheduling problem. He manifested the collective nature of ant and uses their intelligence to solve this problem.

Dynamic task scheduling algorithms uses three things to schedule the tasks and they are – task size, resource processing power and load on resource.

Task size is the duration of the task for which it requires resource time, resource processing power decides how many tasks it can handle. More power means more processing speed and more tasks it can handle in relative amount of time. Resource load determine the time of unavailability of resource. More load means less free time. Scheduler assign task to the resource on the criteria that which task gives the least completion time on which machine and it is known as best completion time.

Earliest Deadline First - The basic principle of this algorithm is very intuitive and simple to understand. In this scheduling algorithm, at every scheduling point the task having the shortest deadline is taken up for scheduling. Earliest Deadline First (EDF) or Least Time to go is a dynamic scheduling algorithm used in real-time operating systems that places processes in a priority queue. Whenever a scheduling event occurs (end of task, release of new task.) then the queue will be searched for the process that is closest to its deadline, the found process will be the next that is going to be scheduled for execution [23].

The deadline distribution algorithm [24] meets the deadline for delivering results to minimize the cost. This algorithm partitions a workflow and distributes the overall deadline into each task based on their workload and dependencies. Deadline distribution algorithm uses Synchronization Task Scheduling (STS) for synchronization tasks and Branch Task Scheduling (BTS) for branch partition respectively. Each task has its own sub deadline and a local optimal schedule can be generated for every task. If each local schedule guarantees completion of their task execution within their sub-deadlines, the whole workflow execution will be completed within the overall deadline. Similarly, the result of the cost minimization solution for each task leads to an optimized cost solution for the entire workflow [25]. Therefore, an optimized workflow schedule can be constructed from all local optimal schedules. The schedule allocates every workflow task to a selected service such that it can meet its assigned sub-deadline at a low execution cost.

Backtracking algorithm finds solutions to computational problem that incrementally builds candidates to the solutions, and abandons each partial candidate c as soon as it determines that c cannot possibly be completed to a valid solution [26]. The back-tracking algorithm [27] assigns available tasks to the least expensive computing resources. The unscheduled task's parent tasks are scheduled. If more than one task is available, the algorithm assigns the task with the largest computational demand to the fastest resources in its available resource list. This process is repeated until all tasks have been scheduled. After each step, the execution time of the current assignment is computed. If the execution time exceeds the time constraint, the back tracks the previous step, removes the resource with minimum expense from its resource list and reassigns tasks with the reduced resource set. Backtracking keeps the previous steps where the resource list is empty and reduces the relative resource list and reassigns the tasks.

Modified ant colony optimization is an approach for diversified service allocation and scheduling mechanism in cloud paradigm. The main aim of this optimization method is to minimize the scheduling throughput to service all the diversified requests according to the different resource allocator available under cloud computing environment [28].

Improved activity based cost based algorithm- This algorithm is used for efficient scheduling of tasks to available resources in cloud. In this algorithm the cost and computation performance are scheduled. It also improves the computation or communication ratio by grouping the user tasks according to a particular cloud resource's processing capability and sends the grouped jobs to the resource [29].

Particle Swarm Optimization algorithm Particle swarm optimization is used to minimize the total cost of execution of application workflows on Cloud computing environments. It calculates the total cost of execution by varying the communication cost between resources and the execution cost of computing resources [30].

Market Oriented Hierarchical Scheduling Market-oriented hierarchical scheduling strategy consists of service-level scheduling and task-level scheduling. Service level scheduling deals the Task-to-Service assignment and the task-level scheduling deals with the optimization of the Task to-VM assignment in local cloud data centers. It can be used to optimize the time and cost simultaneously [31].

Profit-driven Service Request Scheduling Two sets of profit-driven service request scheduling algorithms are presented. These algorithms are devised incorporating a pricing model using process sharing (PS) and two allowable delay metrics based on service and application. Authors have demonstrated the efficiency of consumer applications with interdependent services. The evaluation of the algorithm was done on the basis of maximum profit and utilization [32]. A comparative

Chapter 4

PROPOSED WORK

4.1 MOTIVATION

Cloud computing is the today's emerging technology which offers various services on demand over Internet. Success of cloud computing depends upon these offerings and also on how good these services are being served by the cloud providers. To solve this purpose it is required that internal working of the cloud should be properly designed and managed. Better utilization of the resources and using energy efficient methodology are the major concerns for the working of cloud computing environment. A good task scheduling algorithm is required for the proper and efficient utilization of resources. Many scheduling algorithms are available which fit in different conditions. Like if provider offers high response time for his services then it can use round robin algorithm or high quality of services required then a good priority based scheduling algorithm is required. Task scheduling is a NP complete problem therefore an optimization searching technique is also applicable here which can search an optimally accepted solution within a short duration from a large search space. [6] Task scheduling algorithm has to accomplish aim like expected results, efficient utilization of resources, low response time, low make-span time, higher throughput, low execution time and many more.

4.2 PARTICLE SWARM OPTIMIZATION

It is an optimization technique that uses swarm intelligence for the purposes of solving complex problems. PSO is proposed by Kennedy, Eberhart and Shi. It was used to simulate social behavior. Swarm is the group of particles or candidate solutions. PSO found solutions by moving the particles into the search space using some mathematical formula over particle position or velocity. Each particle's position is moved to other position based on its local best position and its local best position is updated under the influence of global best known position in the search space which is under the influence of current particle as well as other particles of the swarm.

4.3 PROBLEM DEFINITION

A task scheduling problem in cloud environment is define as the process of mapping n tasks to the m virtual machines where one task can be mapped to only one virtual machine but one virtual machine can be mapped to one or more than one tasks. Consider a set of virtual machines, V = {vm_1, vm_2, vm_3,……..,vm_n} and a set of tasks, T = { t_1, t_2, t_3,……t_n}. Now a candidate solution to the scheduling problem should look like this.

Solution: -

Task1	Task2	Task3	…………………………..	Taskn
Vm2	Vm3	Vm6	………………………….	Vmm

4.4 PROPOSED ALGORITHM

The proposed algorithm is based on the assumption that virtual machine has a single resource which is processing element and task has a single requirement which is processing elements. Required processing elements and the available processing elements both can be one or more than one separately.

1. Steps involved in algorithm

2. Set parameters like iteration number, task count, particle count, virtual machine count.

3. Initialize position and velocity vector.

4. For i =1 till iteration number do the following

5. For each particle do the following

6. Calculate fitness of each particle and called it as local fitness

7. If local fitness < best fitness of the particle then

8. Best fitness of the particle = local fitness and

9. Best position of the particle = current position of particle

10. End if.

11. If local fitness < global fitness of the particle then

12. GlobalBest fitness of the swarm = local fitness and

13. GlobalBest position of the particle = current position of particle

14. End if.

15. Update velocity.

16. Update position.

17. End for

Fitness function F to calculate local fitness is as follows: -

F is defines as the summation of modulus of the difference between the available processing elements available with a virtual machine and the required processing elements by the tasks that are submitted to it.

$$F(\text{particle}) = \sum \text{mod}(\text{PE in virtual machine} - \text{PE required by task})$$

Objective function of the optimization problem is to minimize the Fitness function F(particle).

$$\text{Objective function} = F_{min}(\text{particle}).$$

Velocity and position of the particle in Step 14 and 15 of the algorithm can be updated using the formula given below.

Velocity at (K+1) th iteration = w*velocity at K th iteration + C_1 * r_1 * (particle best position at K th iteration – particle position at K th iteration) + C_2 * r_2 * (global best position known till K th iteration - particle position at K th iteration) and

Position at (K+1)th iteration = (position at K th iteration + velocity at (K+1)th iteration)%VM count) where,

W = weightage given to previous,

C_1 , C_2 = Learning factors,

R_1 , r_2 , r_3 = random numbers between 0 and 1.

VM count = total number of virtual machines available in the data center.

While updating velocity of the particles, it is taken care that they should remain in permissible limits i.e. particle should not move abruptly in the search space. Therefore velocity should be kept between vmax and vmin.

Velocity updation process is the key phase in moving the particle to the correct destination. Proposed method has a constraint that at a particular instance of time a task can be assigned to only virtual machine. Hence a modification is required in velocity updation step of traditional PSO technique to fit in the given scheduling problem.

4.5 MODIFICATION IN VELOCITY UPDATION STEP

Steps involved in modification are as follow:

1. Create a temporary matrix,and called it velocity-grid, of size m*n where,
   m is the task count and
   n is the virtual machine count
2. Randomly assign value in it from the range of (0, 1).
3. Create another temporary matrix, and called it as sigmoid grid, of size m*n
4. Calculate the valueof each element of sigmoid grid as follows:
   sigmoid_grid[I,j] = 1/(1+exp$^{\text{-velocity\_grid[I,j]}}$)
   where $0 \leq i \leq task\_count$ and $0 \leq j \leq virtual\ machine\_count$
5. Create another matrix and called it as normalized matrix of size m*n
   normalized_grid[I,j] = sigmoid_grid[i,j]/∑sigmoid_grid[i,k]
   where $0 \leq i \leq task\_count, 0 \leq j \leq virtual\ machine\_count$ and $0 \leq j \leq virtual\ machine\_count.$
6. Based on the normalized grid update the position of particle as follows:
   Assign the task to that virtual machine whose value of the normailized_task grid is bigger than the values of the normalized_task grid of other virtual machines for that current task.

To illustrate the proposed algorithm, consider the scenario below where a datacenter has 3 virtual machines and task size is of length 4 with their respective available or required processing elements are given below in the table. Consider particle numbers to be 3 in number.

VIRTUAL MACHINE TABLE

Virtual machine ID	Available PES
Vm1	2
Vm2	2
Vm3	3

TASKS TABLE

Tasks ID	Required PES
T1	1
T2	2
T3	2
T4	2

Before entering the loop, velocity and position of are initialized randomly as follows: -

PARTICLES POSITION

	T1p	T2p	T3p	T4p	Fitness value
Ptc1.	Vm1	Vm2	Vm2	Vm3	
Ptc2.	Vm3	Vm1	Vm3	Vm1	
Ptc3.	Vm1	Vm3	Vm2	Vm2	

PARTICLES VELOCITY

	T1v	T2v	T3v	T4v
Ptc1	1	1	-2	4
Pt2	-3	1	-1	2
Pt3	-4	4	2	2

Now, the algorithm undergoes in the generation of best position of the particles and after that generates the global best position known till the current situation.

After that the position and velocity of the particle undergo in step number 15 and 16 of the algorithm. After 1st iteration, second position of the particle becomes: -

PARTICLE POSITION AFTER 1ST ITERATION

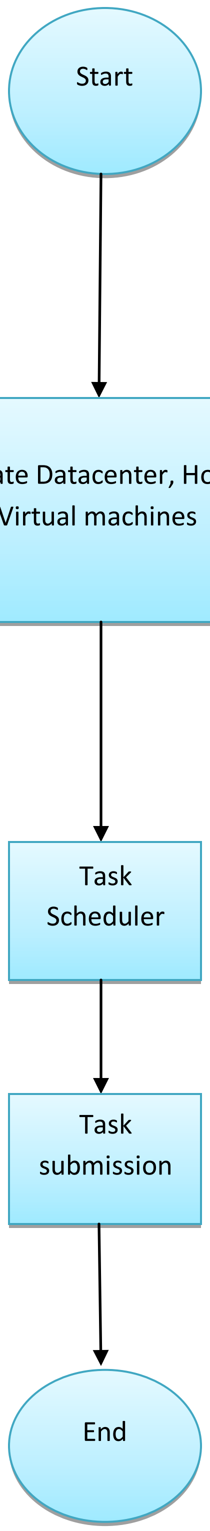
	T1p	T2p	T3p	T4p	Fitness value
Ptc1.	Vm1	Vm2	Vm2	Vm3	
Ptc2.	Vm3	Vm1	Vm3	Vm1	
Ptc3.	Vm1	Vm3	Vm2	Vm2	

PARTICLES VELOCITY AFTER 1ST ITERATION

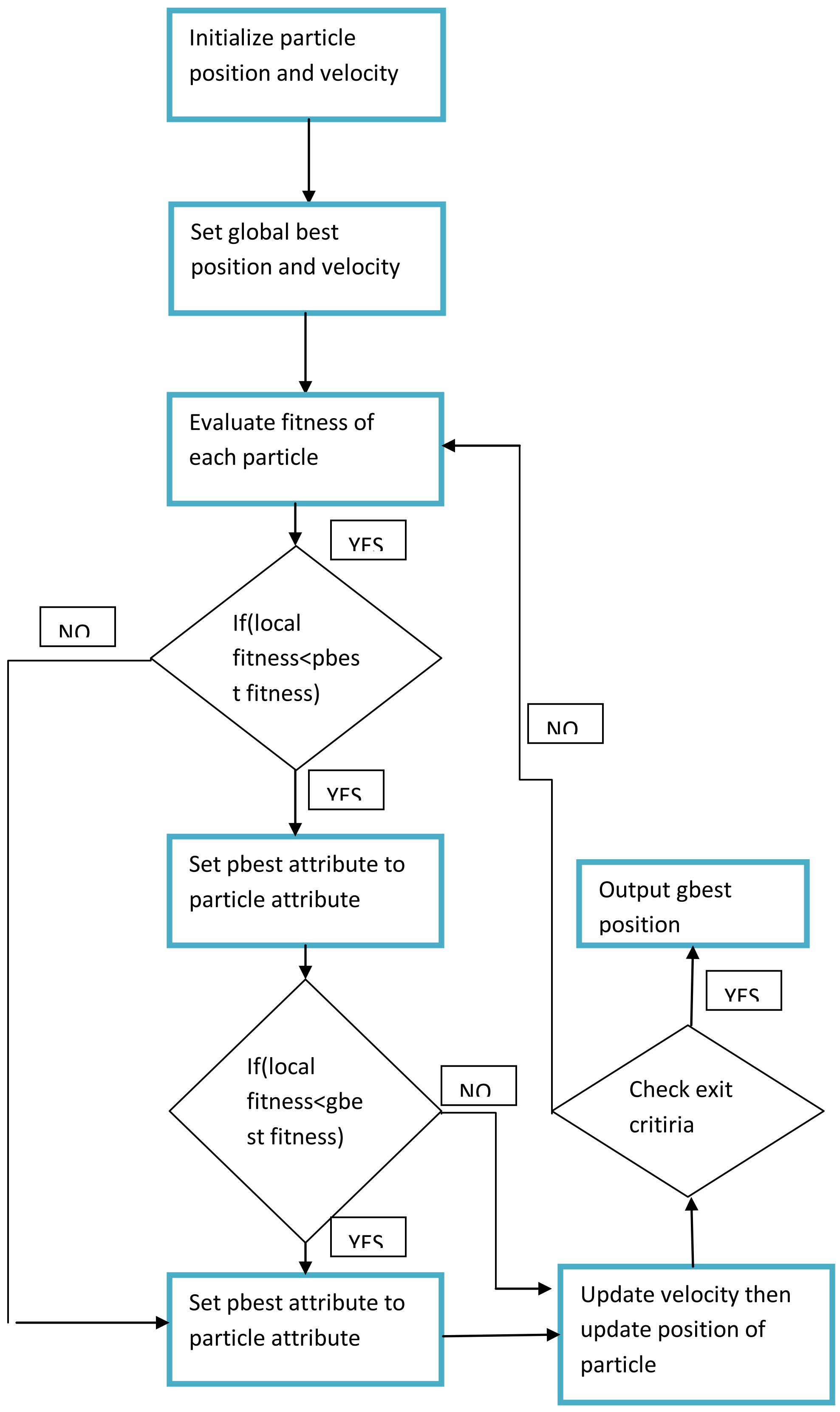
	T1v	T2v	T3v	T4v
Ptc1	-1	1	3	4
Pt2	-2	2	-2	-3
Pt3	-4	-2	-2	2

Now the exit criterion is checked by the algorithm which is if any of the particle reaches the fitness value = 0 or maximum iteration becomes 100 then algorithm quits and produces a schedule which then can be used for sending tasks to their respective allocated virtual machine. Otherwise algorithm goes for another round of iteration.

4.6 FLOW CHART OF PROPOSED ALGORITHM



DETAIL FLOW CHART OF TASK SCHEDULER


Initialize particle position and velocity
Set global best position and velocity
Evaluate fitness of each particle
YES
NO
If(local fitness<pbest fitness)
YES
NO
Set pbest attribute to particle attribute
If(local fitness<gbest fitness)
NO
YES
Output gbest position
YES
Check exit critiria
Set pbest attribute to particle attribute
Update velocity then update position of particle

Chapter 5

EXPERIMENT RESULTS

PRACTICAL RUN OF PROPOSED ALGORITHM

INPUT DATA

Particle count	20
Virtual machine count	5
Task count	10
W factor	0.9
Confidence factor, C1	2
Confidence factor, C2	2
Vmax	8
Vmin	-8

Here we discuss different cases in terms of required processing elements by user tasks and available processing elements in virtual machines.

CASE I.

INPUT DATA

TASK ID	**REQUIRED PES**
1	2
2	1
3	1
4	1
5	3
6	3
7	3
8	3
9	1
10	1

VM ID	**AVAILABLE PES**
1	3
2	3
3	5
4	5
5	3

--------------**OUTPUT**---------------------

TASK ID	VM ID
1	1
2	1
3	4
4	3
5	2
6	5
7	3
8	4
9	4
10	3

Task left= 0.

CASE II

TASK ID	REQUIRED PES
1	2
2	3
3	2
4	3
5	2
6	2
7	1
8	3
9	2
10	3

VM ID	AVAILABLE PES
1	3
2	5
3	4
4	4
5	3

--------------**OUTPUT**---------------------

TASK ID **VM ID**

1	4
2	1
3	2
4	2
5	3
6	NA
7	4
8	5
9	NA
10	NA

Task left = 3.

CASE III

TASK ID	REQUIRED PES
1	3
2	1
3	2
4	3
5	3
6	1
7	2
8	1
9	1
10	2

VM ID	AVAILABLE PES
1	5
2	4
3	3
4	5
5	5

--------------**OUTPUT**---------------------

TASK ID **VM ID**

1	5
2	1
3	1
4	5
5	3
6	4
7	2
8	4
9	1
10	4

Task left = 0.

CASE IV

TASK ID	**REQUIRED PES**
1	2
2	2
3	3
4	2
5	2
6	1
7	2
8	1
9	3
10	2

VM ID	**AVAILABLE PES**
1	3
2	3
3	5
4	5
5	5

--------------**OUTPUT**---------------------

TASK ID	**VM ID**
1	3

2	2
3	4
4	4
5	5
6	5
7	5
8	3
9	1
10	4

Task left = 0.

COMPARISON RESULTS

The results of proposed algorithms are compared with other static scheduling algorithms like FCFS and bankers algorithm applied in cloud environment in the following graphs. Banker's algorithm is the one that is designed by Edsger Dijkstrafor maximum allocation of resources and to avoid deadlocks. Different scenario are considered for this purpose.

SCENARIO I

In this scenario, proposed algorithm is run under following condition:

INPUT DATA

Particle count	20
Virtual machine count	5
Task count	Vary from 10 to 100
W factor	0.9
Confidence factor, C1	2
Confidence factor, C2	2
Vmax	4
Vmin	-4

OUTPUT GRAPH (VIRTUAL MACHINE COUNT = 5)

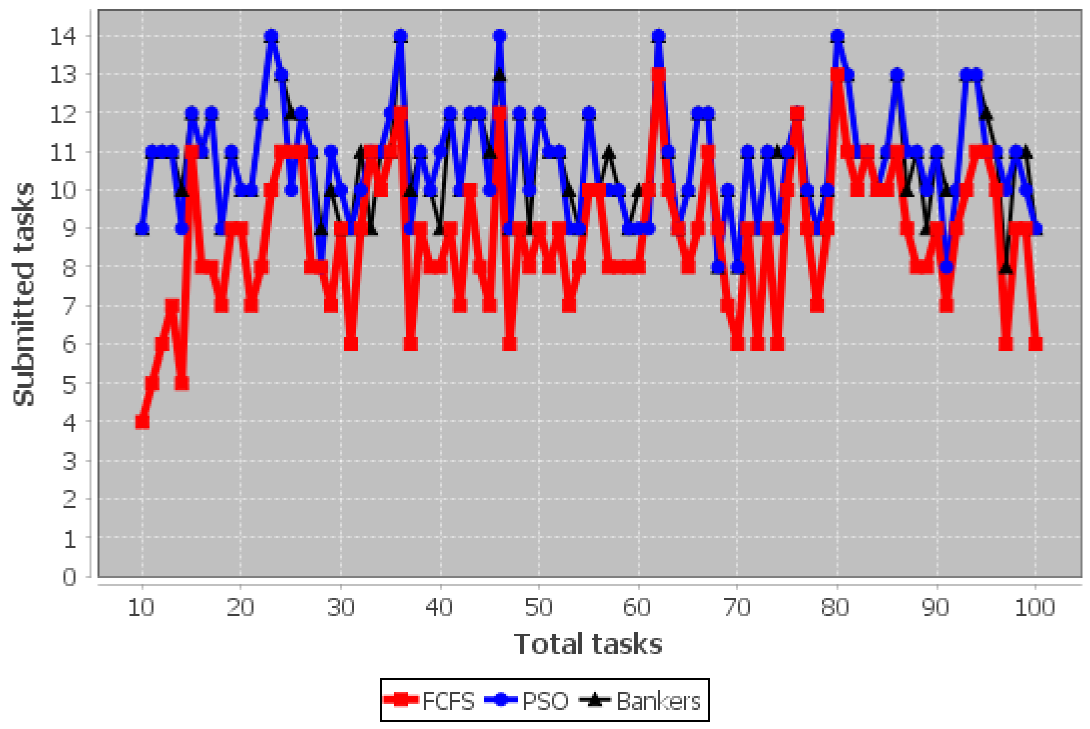


Fig. 5.1

SCENARIO II

In this scenario, proposed algorithm is run under following condition:

INPUT DATA

Particle count	20
Virtual machine count	10
Task count	Vary from 10 to 100
W factor	0.9
Confidence factor, C1	2
Confidence factor, C2	2
Vmax	4
Vmin	-4

OUTPUT GRAPH (VIRTUAL MACHINE COUNT = 10)

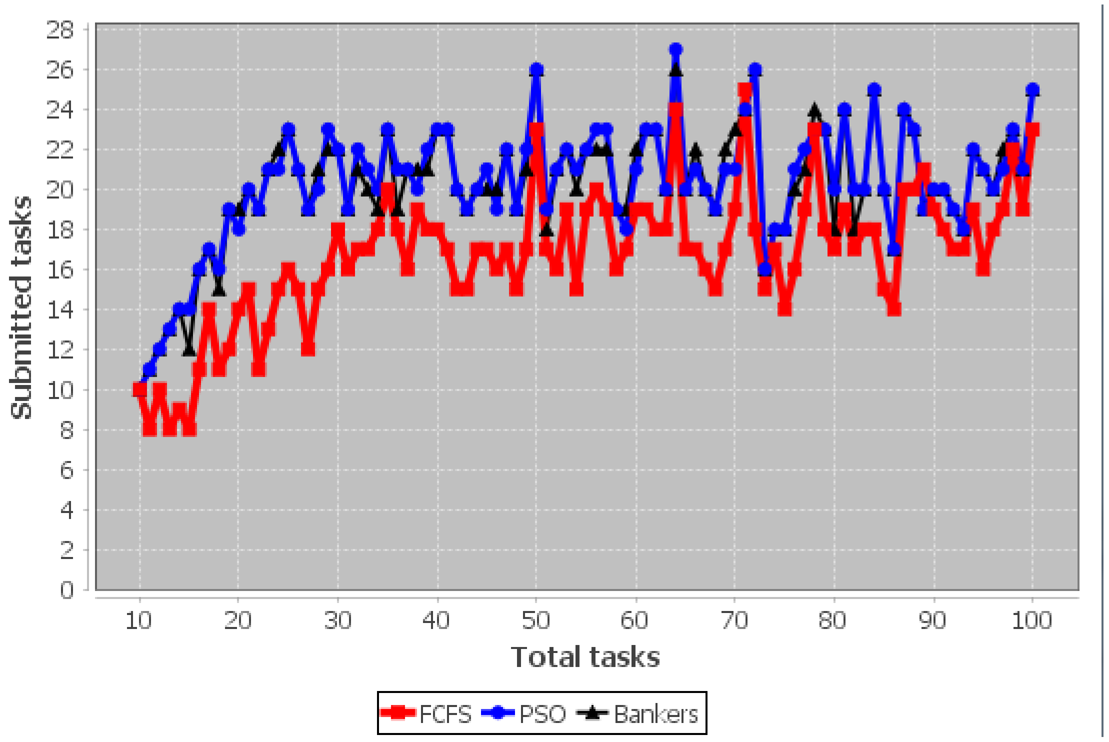


Fig. 5.2

SCENARIO III

In this scenario, proposed algorithm is run under following condition:

INPUT DATA

Particle count	20
Virtual machine count	15
Task count	Vary from 10 to 100
W factor	0.9
Confidence factor, C1	2
Confidence factor, C2	2
Vmax	4
Vmin	-4

OUTPUT GRAPH (VIRTUAL MACHINE COUNT = 15)

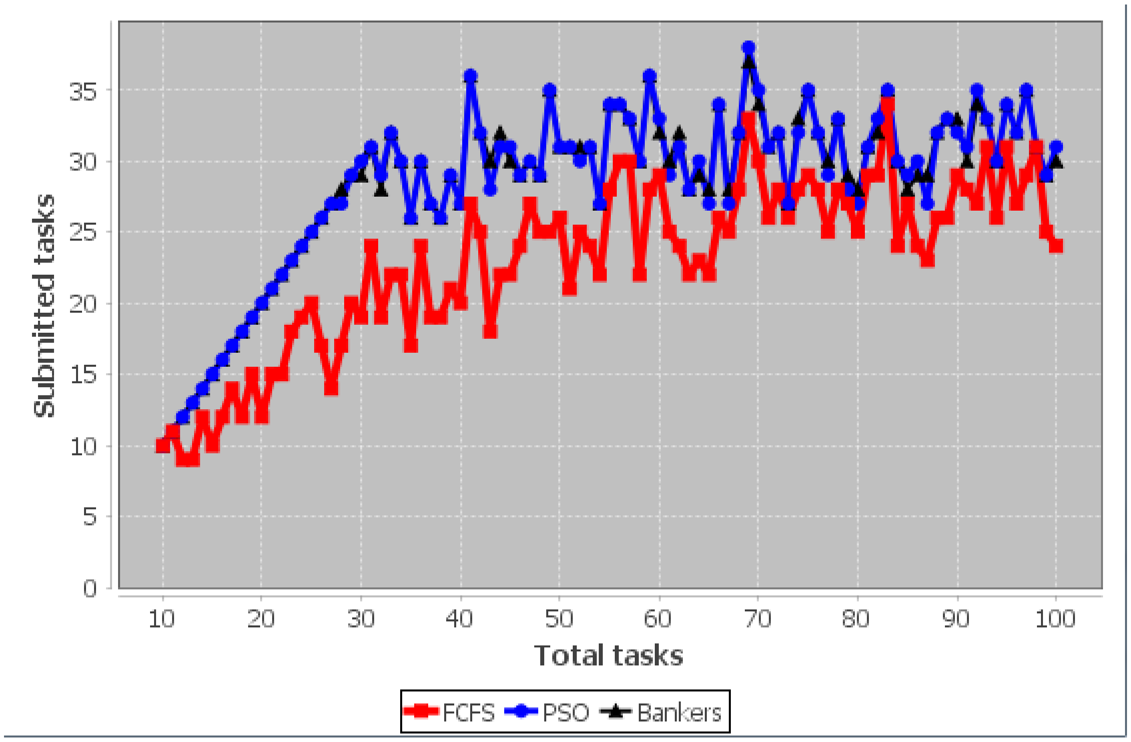


Fig. 5.3

SCENARIO IV

In this scenario, proposed algorithm is run under following condition:

INPUT DATA

Particle count	20
Virtual machine count	20
Task count	Vary from 10 to 100
W factor	0.9
Confidence factor, C1	2
Confidence factor, C2	2
Vmax	4
Vmin	-4

OUTPUT GRAPH (VIRTUAL MACHINE COUNT = 20)

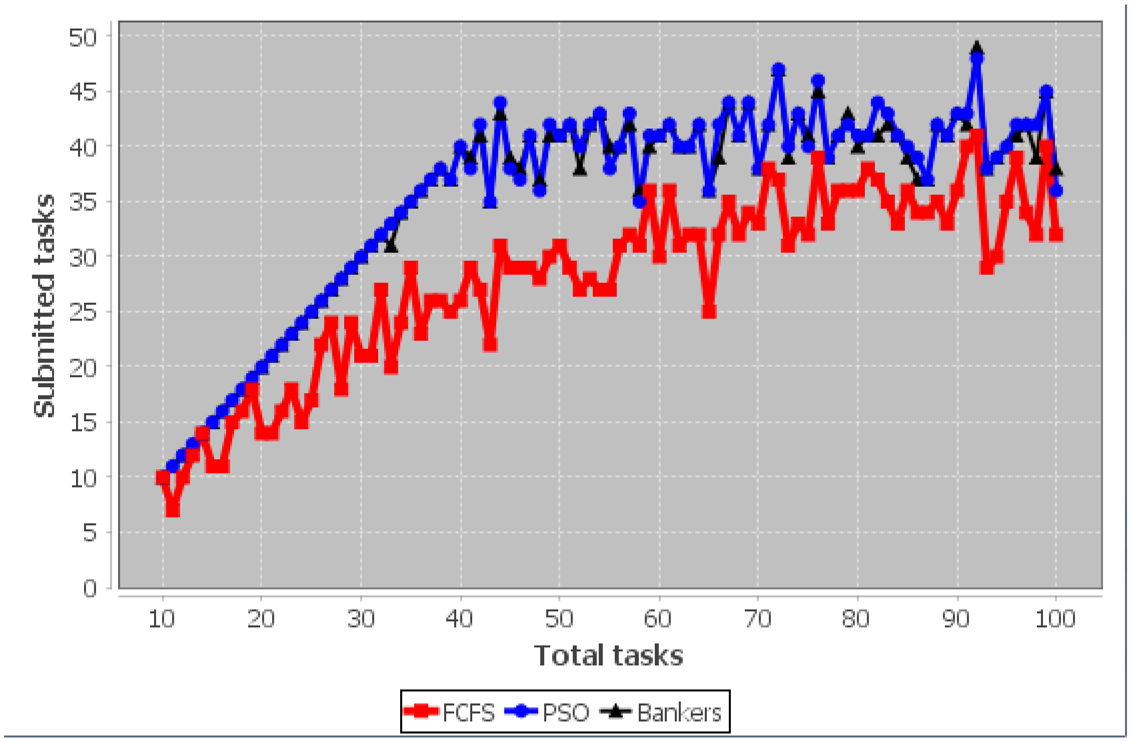


Fig. 5.4

It is clear from the above four results captured at different input data that proposed task scheduling algorithm is giving better results than other scheduling algorithm (FCFS and bankers algorithm) in allocation resources to user tasks.

Chapter 6

COCLUSION AND FUTURE WORK

CONCLUSION

Task scheduling is one of the crucial tasks in cloud computing environment. It is the process of mapping user’s tasks onto the virtual machines. Many algorithms are designed for this problem like static algorithms e.g. FIFO, round robin, priority based scheduling algorithm etc. Some dynamic algorithms like independent task scheduling algorithms, genetic algorithms etc. are also designed to solve this problem. All designed algorithms have the common aim which is to get increased throughput, high scalability, high reliability, and high performance of the cloud system. But all are success in achieving few of these targets. In our approach we use an optimization technique named PSO (Particle swarm optimization technique) which is a meta-heuristic technique to solve the scheduling problem. It uses probability in assigning tasks to the virtual machines. In our experiment we proved that PSO is effective in efficient mapping of resource to the requested tasks. Task scheduling is a combinatorial searching problem and the available search is of non-polynomial order. Therefore we used the swarm intelligence to narrow down the search space. We have used different combination of input data in our experiment. We vary the number of virtual machine as well as the number of tasks in a single host that reside on a single datacenter. Through our experiment we proved that a task scheduler that used PSO technique is effective and efficient in solving scheduling problem. In our results, we have shown that the proposed technique is giving better output than FIFO scheduling and bankers scheduling algorithms.

FUTURE WORK

Although the proposed technique is proves to be a better technique of finding solution of given problem, yet there is a lot of space for improvement. Other optimization techniques like DNA, honey bee, firefly searching methods can be applied at different phases of propose method to take advantage of these algorithms. Also, given algorithm considers only criteria and that is checking processing elements availability in virtual machines. Other factors can also be included in evaluating fitness function. These factors are make-span time, execution time, performance, idle time of virtual machines, energy and power of the cloud environment etc.

REFERENCES:

[1] Pinal Salot M.E, Computer Engineering, Alpha College of Engineering, Gujarat, India (2013)" A survey of various scheduling algorithm in cloud computing environment".

[2] Poonam Devi Department of CSE BITS College, BHIWANI, MDU, India (2013) BITS College."Implementation of Cloud Computing By Using Short Job Scheduling".

[3] Sujit Tilak, 2Department of COMPUTER, PIIT, New Panvel, Maharashtra, India (2012). "A Survey of Various Scheduling Algorithms in Cloud Environment".

[4] Yogita Chawla 1,2Pune University, G.H Raisoni College of Engg & Mgmt, Gate No.: 1200 Wagholi, Pune – 412207(2012) "A Study on Scheduling Methods in Cloud Computing".

[5] Jairam Naik K. Research Scholar, Faculty of Computer Science & Engineering, JNT University, Hyderabad, India (2012) "Scheduling Tasks on Most Suitable Fault tolerant Resource for Execution in Computational Grid".

[6] Swachil Patel (2013) "Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review".

[7] Sonam rathore "efficient allocation of virtual machine in cloud computing environment".

[8] Arash Ghorbannia Delavar, Mahdi Javanmard , Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi "RSDC (Reliable Scheduling Distributed in Cloud Computing)" in International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3, June 2012

[9] Saeed Parsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009.Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.

[10] Dr. M. Dakshayini, Dr. H. S. Guruprasad "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment" International Journal of Computer Applications (0975 – 8887) Volume 32– No.9, October 2011

[11] Shamsollah Ghanbari, Mohamed Othman "A Priority based Job Scheduling Algorithm in Cloud Computing" International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012).

[12] El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy "Extended Max-Min Scheduling Using Petri Net and Load Balancing" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012

[13] Shalmali Ambike, Dipti Bhansali, Jaee Kshirsagar, Juhi Bansiwal " An Optimistic Differentiated Job Scheduling System for Cloud Computing" International Journal of Engineering Research and

Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2,Mar-Apr 2012, pp.1212-1214

[14] Mrs.S.Selvarani1; Dr.G.Sudha Sadhasivam, "improved cost-based algorithm for task scheduling in Cloud computing", IEEE 2010.

[15] Swachil Patel, Upendra Bhoi "Priority Based Job Scheduling Techniques In Cloud Computing: A Systematic Review" International Journal of scientific & Technology research volume 2, issue 11, November 2013 ISSN 2277-8616.

[16] Jairam Naik K" Scheduling Tasks on Most Suitable Fault tolerant Resource for Execution in Computational Grid International Journal of Grid and Distributed Computing" Vol. 5, No. 3, September, 2012.

[17] Salim Bitam Computer science department Mohamed Khider University of Biskra "Bees Life Algorithm for Job Scheduling in Cloud Computing".

[18] Tejinder Sharma, Vijay Kumar Banga **"**Efficient and Enhanced Algorithm in Cloud Computing".

[19] R.Raju "Minimizing the Makespan using Hybrid Algorithm for Cloud Computing".

[20] Rodrigo N. Calheiros1, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms."

[21] AV.Karthick, "An Efficient Multi Queue Job Scheduling for Cloud Computing" 2014 World Congress on Computing and Communication

[22] Yung-Ching Hsu , "Job Sequence Scheduling for Cloud Computing" 2011 International Conference on Cloud and Service Computing.

[23] J. Singh, B. Patra, S. P. Singh ,"An Algorithm to Reduce the Time Complexity of Earliest Deadline First Scheduling Algorithm in RealTime System", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.2, February 2011.

[24] J. Yu, R. Buyya and C. K. Tham, "A Cost-based Scheduling of Scientific Workflow Applications on Utility Grids", Proc. of the 1st IEEE International Conference on e-Science and Grid Computing, Melbourne, Australia, December 2005 , pp140-147.

[25] Kaur, P.D., Chana, I. ʊUnfolding the distributed computing paradigm ,In: International Conference on Advances in Computer Engineering, pp. 339-342 (2010).

[26] Gilles Brassard, Paul Bratley(1995). Fundamentals of Algorithmics.Prentice-Hall. [24] D. A. Menasc and E.Casalicchio, "A Framework for Resource Allocation in Grid Computing", Proc. of the 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, Volendam, The Netherlands, October, 2004,pp 259-267.

[27] D. A. Menasc and E.Casalicchio, “A Framework for Resource Allocation in Grid Computing”, Proc. of the 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, Volendam, The Netherlands, October, 2004,pp 259-267.

[28] S Banerjee, I Mukherjee, and P.K. Mahanti “Cloud Computing Initiative using Modified Ant Colony Framework” , World Academy of Science, Engineering and Technology, , 56 2009, pp 221-224.

[29] Selvarani, S.; Sadhasivam ,G.S.,”Improvedcostbased algorithm for task scheduling in cloud computing”, Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International conference on 2010 , pp 1–5.

[30] K. Liu, Y. Yang, J. Chen, X. Liu, D. Yuan , H. Jin, “A CompromisedTime-Cost Scheduling Algorithm in SwinDeW-C for Instance-intensive Cost-Constrained Workflows on Cloud Computing Platform”, Int. Journal of High Performance Computing Applications, Volume 24 Issue 4, 2010.

[31] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, “A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing”, in 2009 IEEE International Symposium on Parallel and Distributed Processing.

[33] Young Choon Lee, Chen Wang, Albert Y. Zomaya, Bing BingZhou,”Profit-driven Service Request Scheduling in Clouds”, 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing,2010