International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.2, No.4, August 2013, pp. 296~312
ISSN: 2089-3337

Service Support Aware Resource Allocation Policy for Enterprise Cloud-based Systems

Rabi Prasad Padhy*, Manas Ranjan Patra**
*Senior Software Engg, Oracle Corp., Bangalore, Karnataka, India
** PG Department of Computer Science, Berhampur University, Odisha, India

Article Info

Article history:

Received Jun 12th, 2013
Revised Jul 10th, 2013
Accepted Jul 30th, 2013

Keyword:

Resource Allocation
Resource Utilization
Resource Management
Resource Broker
Cloud Controller

ABSTRACT

Cloud Services can save business time, resources, money and it's the ultimate solution for solving challenges in the traditional software sales model. However, cloud computing must be advanced to focus on resource utilization and resource management. The use of cloud-based services allows consumers to allocate resources on-demand and to pay only for the resources they actually use. Considering a scenario in which cloud service providers stipulate on service support level i.e. Type of support (platinum, gold, silver and bronze) with end-users and lease cloud services in a way that guarantees SLA fulfillment, minimize operational costs and maximize the profit. So in this case allocating resources dynamically in the form of virtual machines to end users directly depends on the parameter 'type of support' a cloud consumer subscribes. In this research paper we proposed resource allocation algorithms for cloud providers. Our proposed policy engine designed in such a way that it ensure user request type which takes into account that the provider has to fulfill the Service Support Level criteria while minimizing the resources outsourced from the cloud infrastructure using resource allocation algorithms.

Corresponding Author:

Rabi Prasad Padhy,
Senior Software Engineer,
Oracle Corporation, Bangalore
Karnataka, India.
Email: rabi.padhy@gmail.com

1. INTRODUCTION

Cloud computing enables users to access compute resources on demand without the burden of owning, managing, and maintaining the resources. To support Infrastructure as a Service (IaaS), most cloud platforms use virtualized data centers. Typically, a cloud data center maintains a catalog that lists available virtual machine (VM) images. Those images may contain only the bare operating system such as Linux Red Hat or Windows, include popular applications such as database management systems, or even be created by users. Data centers typically provision diverse VMs to provide various services, applications and other compute resources. Computing resources provided by cloud vendors can be categorized as computing power, network resources (bandwidth, IP addresses etc.) and storage. The cloud services that were provided earlier can be classified into three categories: Software as a Service (SaaS) i.e. applications without the traditional costs of managing them in-house[1]. Some of the example includes operating systems, basic software like databases or web servers. Platform as a Service (PaaS) i.e. Access an environment for application development, management and integration in the cloud. Some of the example includes content managements

systems, social networking software and Infrastructure as a Service (IaaS) i.e. Provision hardware, storage and server equipment in the cloud. Some of the example includes Servers, storage, CPU, memory, bandwidth[2].

Cloud computing assembles large networks of virtualized services: hardware resources (CPU, storage, and network) and software resources (e.g., databases, message queuing systems, monitoring systems, load-balancers). Using cloud services, cloud users can deploy a wide variety of applications dynamically and on-demand. Cloud providers including Amazon Web Services (AWS), Microsoft Azure, Salesforce.com, Google App Engine, and others give users the option to deploy their application over a network of infinite resource pool with practically no capital investment and with modest operating cost proportional to the actual use. Cloud service allows large enterprise class and individual users to acquire computing resources from large scale data centers of service providers[3]. This cloud service is more involved in purchasing and consuming manners between providers and users than others. However, Cloud service providers charge users for these services. Specifically, to access data from their globally distributed storage edge servers, providers charge users depending on the user's location and the amount of data transferred. However, there are significant problems that exist with regard to efficient provisioning and delivery of applications using Cloud-based IT resources.

2. MOTIVATION

Cloud computing is the delivery of resources and services on an on-demand basis over a network. Three markets are associated to it. Infrastructure-as-a-Service (IaaS) designates the provision of IT and network resources such as processing, storage and bandwidth as well as management middleware. Platform-as-a-Service (PaaS) designates programming environments and tools supported by cloud providers that can be used by consumers to build and deploy applications onto the cloud infrastructure. Software-as-a-Service (SaaS) designates hosted vendor applications. IaaS, PaaS and SaaS all include self-service (APIs) and a pay-as-you-go billing model. Companies can use clouds either to run punctual batch jobs (e.g. video transcoding) or web applications. The cloud is appealing to them because of its ability to reduce capital expenditures and increase return on investment (ROI) since the traditional model where physical servers were bought and amortized in the long term is no more. Allocating resources to applications in clouds has been the subject of several works in recent years. In this paper, we concentrate on service support levels aware resource allocation by using policy engine and algorithms.

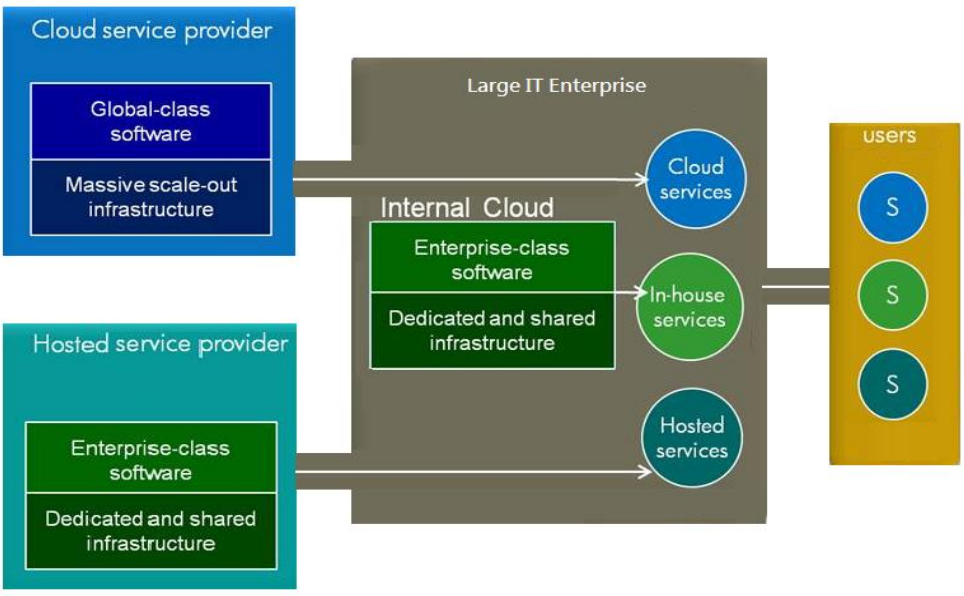

Figure 1. High Level View of Different Service Utilization of a Large IT Enterprise

The computing resources, either software or hardware, are virtualized and allocated as services from providers to users[4]. The computing resources can be allocated dynamically upon the requirements and preferences of consumers. Traditional system-centric resource management architecture cannot process the resource assignment task and dynamically allocate the available resources in a cloud computing environment. Since the consumers may access applications and data of the "Cloud" from anywhere at any time, it is difficult for the cloud service providers to allocate the cloud resources dynamically and efficiently. In cloud

computing, the underlying large-scale computing infrastructure is often heterogeneous, not only because it's not economic and reliable to procure all the servers, network devices and power supply devices in one size and one time, but because different application requires different computer hardware, e.g. workflow extensive computing might need standard and cheep hardware; scientific computing might need specific hardware other than CPU like GPU or ASIC. There are kinds of resources in the large-scale computing infrastructure need to be managed, CPU load, network bandwidth, disk quota, and even type of operating systems. To provide better quality of service, resources are provisioned to the users or applications, via load balancing mechanism, high availability mechanism and security and authority mechanism. To maximize cloud utilization, the capacity of application requirements shall be calculated so that minimal cloud computing infrastructure devices shall be procured and maintained. Given access to the cloud computing infrastructure, applications shall allocate proper resources to perform the computation with time cost and infrastructure cost minimized. Proper resources shall be selected for specific applications[5].

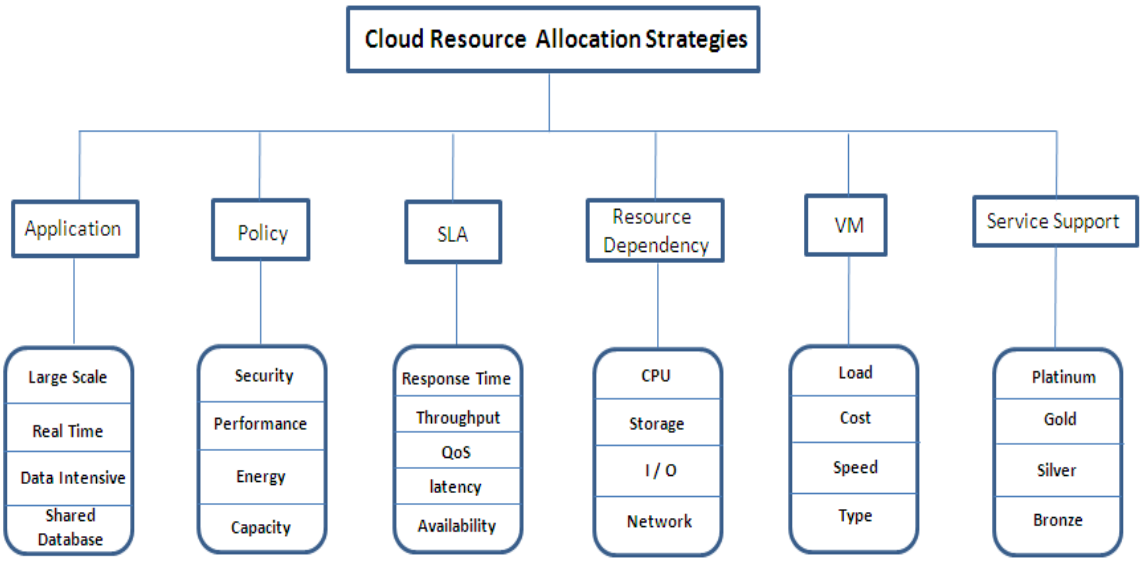

Figure 2. Resource allocation strategies in Cloud Computing Environment

3. COMPONENTS OF OUR PROPOSED MODEL

In this section, we discuss the key concepts and overall design of our proposed model. Figure 3 shows the overall system architecture with multiple VMs accessing a shared pool of physical cloud data center infrastructure. A cloud environment spans several datacenters interconnected by an internet. Each of these datacenters contains a large number of machines that are connected by a high-speed network. Users access sites hosted by the cloud environment through the public Internet. The physical cloud data center infrastructure has four main components like server pool, storage pool and Network pool. The resource pool manager is responsible for determining how much capacity should be provided to each VM according to the resource allocation policy. Cloud broker is responsible for scheduling the VMs in accordance with the request priority. The VMs considered in this research paper are system-level VMs, which virtualize an entire physical host's resources, including CPU, memory, and I/O devices and present virtual resources to the guest OS's and applications. System virtualization is implemented by the layer of software called virtual machine monitor (hypervisor), which is responsible of multiplexing physical resources among the VMs.

User: End users interact with the cloud services and send resource requests. Users generally fall into three categories namely Mobile (smart phones or Tablets), Thin (no computation work, no internal memory but only display the information and servers do all the work for them) and Thick (web browsers like internet explorer or Mozilla Firefox or Google chrome) to connect to the different cloud[6].

Cloud Data Center: A Cloud Data Center is a data center where some or all of the hardware (e.g., servers, routers, switches, and links) are virtualized. Typically, a physical hardware is virtualized using software called hypervisor that divides the equipment into multiple isolated and independent virtual instances[7]. For example, a physical machine (server/host/node) is virtualized via a hypervisor that creates

virtual machines (VMs) having different capacities (CPU, memory, disk space) and running different operating systems and applications. The Cloud is a collection of virtual resources (VMs, virtual switches, and virtual routers) connected via virtual links. While a Virtualized Data Center is a physical data center with deployed resource virtualization techniques, a Virtual Data Center is a logical instance of a Virtualized Data Center consisting of a subset of the physical data center resources. A Virtual Network (VN) is a set of virtual networking resources: virtual nodes (end-hosts, switches, routers) and virtual links; thus, a VN is a part of a VDC.

Resource Pool: A resource pool is comprised of server, network and storage scale units that share a common hardware and configuration baseline which hosts virtualized workloads. Subsets of these resources are allocated to the customers as needed and conversely, returned to the pool when they are not needed. Ideally, the Resource Pool should be as homogenized and standardized as possible[8]. The main features is that it fully managed by service provider including the operating system and Infinite resources available.

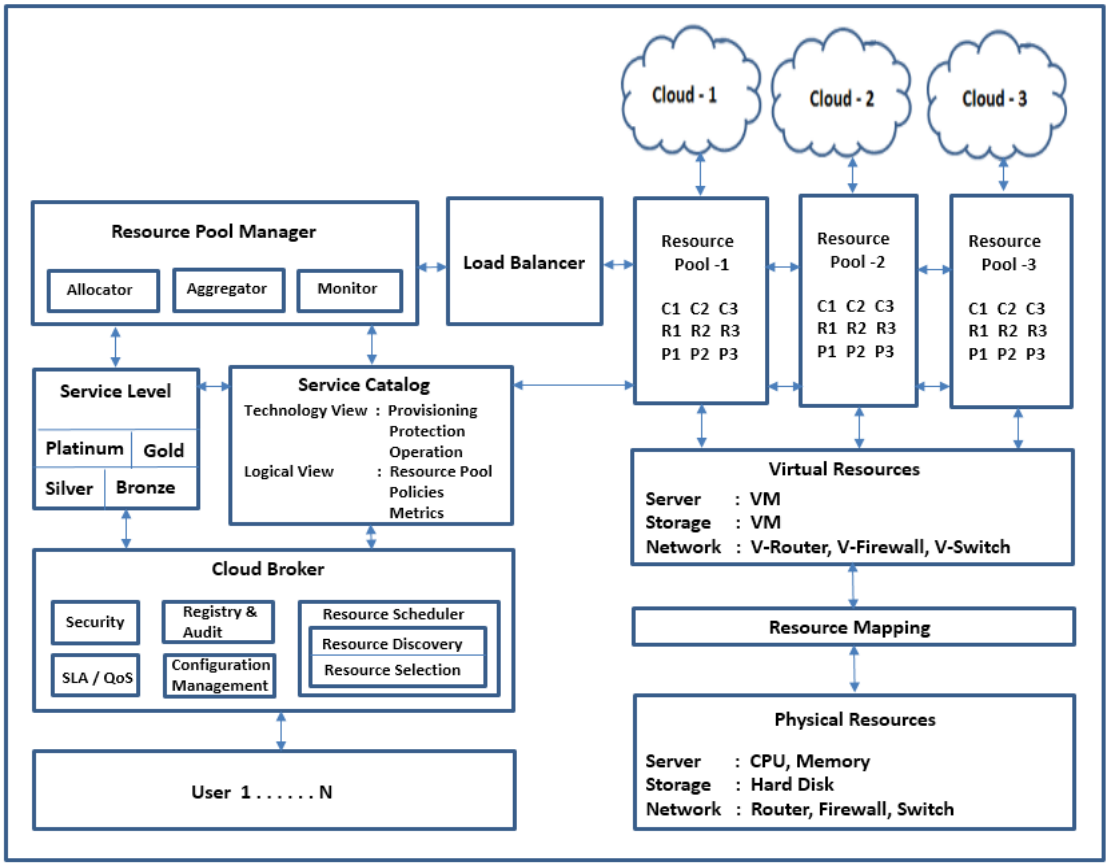

Figure 3. High Level view of Proposed Model

VM: Virtual Machine is the most fundamental and manageable computing element in a Cloud. Each VM, based on user's requirements, can have number of CPUs, memories, storage volumes and an OS defined.

VM pool: is a resource container from which VMs are created. It usually contains a large amount of computing resources which are sliced into smaller resource units based on the user requirements. Many VM pools can be deployed in parallel in a data center in order to provide sufficient computing resources. At implementation level, the concept of VM pool may be mapped to a physical server or a cluster of servers.

Cloud Controller: The Cloud Controller coordinates a collection of services such as VM scheduling, authentication, VM monitoring and management. When a Cloud Controller receives a request from the user to create a VM, it requests its corresponding Cluster Controllers to provide a list their free resources. With this information, the Cloud Controller can choose which cluster will host the requested virtual machine. A Cluster Controller is composed of a collection of Node Controllers, which consist of a pool of Servers that host Virtual Machine (VM) instances. The Cluster Controller handles the state information of its Node Controllers, and schedules incoming requests to run instances. A Node Controller controls the execution, monitoring, and termination of the VMs through a Virtual Machine Monitor (VMM)

which is the one responsible to run VM instances. The Cloud Controller retrieves and stores user data and Virtual Machine Images (VMI)[9]. The Virtual Machine Image Repository contains a collection of Virtual Machine Images that are used to instantiate a VM. The Dynamic Host Configuration Protocol (DHCP) server assigns a MAC/IP (Media Access Control/Internet Protocol) pair address for each VM through the Cloud Controller, and requests the Domain Name System (DNS) server to translate domain names into IP addresses in order to locate cloud resources.

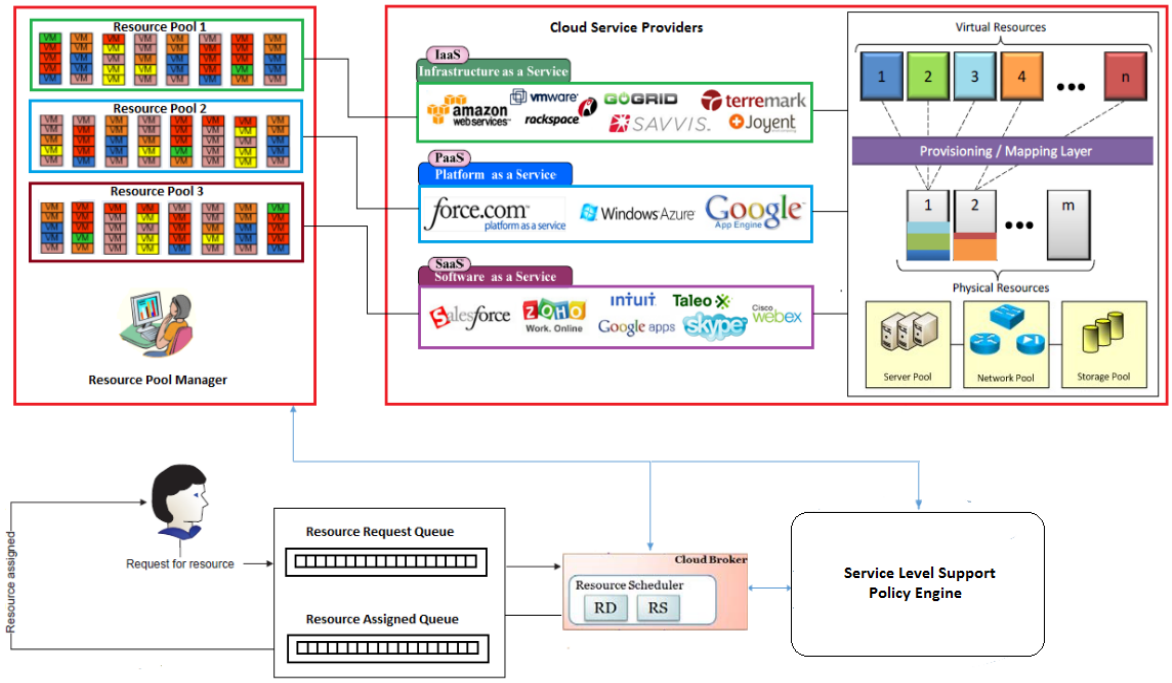

Figure 4. Implementation view of Proposed Model

Cloud Broker: It enables business value of cloud as an intermediary for consumers & providers. It also manage the complexity of cloud services implements, integrates, aggregates, customizes cloud services provides simple & easy access to cloud different solutions on different platforms. The main function of cloud broker is given below.

- Collecting and indexing all the resources available from different cloud service providers.
- Translating the service requirements in terms of high-level parameters such as execution time, throughput, transaction rate into low-level criteria related to computing, storage and network distributed resources.
- Estimating the capacity needs of Virtual Machines
- Managing the available resources in order to ensure specific requirements of QoS.
- Monitoring the usage of allocated resources in order to guarantee the SLA with the user
- Performing load balancing based on resource consumption.
- when services request to allocate extra resources, checking whether they are authorized

Cloud Service Catalog: To access different type of cloud services, there is a cloud service catalog which is a list of services associated with SLAs and costs that a user can request. In cloud systems, a task usually needs a cloud service to execute. This cloud service is registered along with other cloud services by service providers on the cloud service catalog server. Usually, the items in the catalog are maintained by their corresponding service clouds via the heartbeat mechanism. Cloud Service Catalog also called as Cloud Resource Registry[10].

Resource Pool Manager: The Resource Pool Manager in the cloud service provider's datacenter is the prime entity to distribute the execution load among all the VMs by keeping track of the status of the VM. The Resource Pool Manager maintains a data structure containing the VM ID, Request ID of the users that has to be allocated to the corresponding VM and VM Status to keep track of the load distribution. The VM

Pool Status represents the percentage of utilization. The Cloud Pool Manager allocates the resources and distributes the load as per the data structure. The Resource Pool Manager analyzes the VM Pool status routinely to distribute the execution load evenly. In course of processing, if any VM Pool is overloaded then the users are migrated to the VM Pool which is underutilized by tracking the data structure. The Cloud Pool Manager automatically updates the data structure. Table 1 illustrates the sample data structure maintained by the Resource Pool Manger. Resource Pool Manager also manages the resource pools of different cloud services and predict the future resource usage according to the number of requests[11]. Resource Pool Manager also decide from which pool the resources have to be allocated. In addition to this, Resource Pool Manager balance the load among all resource pools by using a load balancing algorithm.

Load Balancer: The load balancer forwards the requests to the server pools based on their available capacities. When delivering the service, server pools may be overloaded with too many requests working simultaneously. Therefore, making sure that the workload is spread out across all servers is quite integral to maintaining an interruption-free environment for the cloud users.

Resource Mapping Layer: Resource Mapping layer is responsible for mappings one PM to multiple VMs using software called a hypervisor/VMM. The Virtual Machine Monitor (VMM) or hypervisor has the complete control over the resource pool allocated to the VMs. It provides the physical resource abstraction to virtual machines deployed on the host, while virtual switches provide connectivity between virtual machines residing on the same host. The cloud Infrastructure is distributed across different geographical data centers, we assume a single data center for our target system model.

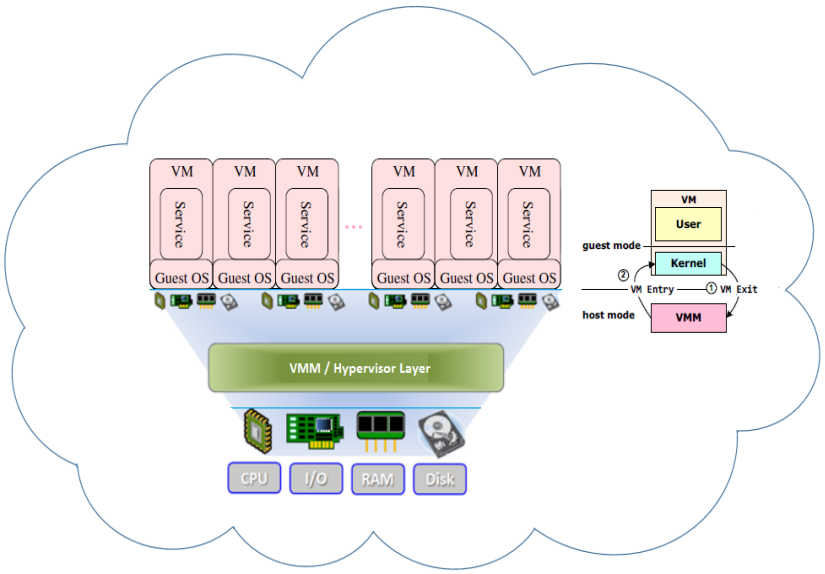

Figure 5: Mapping of virtual to physical resources

In cloud infrastructure environments, a cloud provider services requests from a client on shared physical resources. The client is typically offered the resources that can be configured according to their needs. The parameters that can be configured are typically a number of virtual machines, each of which can be individually configured for the number of virtual CPUs (vCPUs), RAM and disk space[12]. Since many configuration parameters have been simplified from the client's view, virtualized resources that have been provisioned to a client can have ambiguous physical characteristics. For example, when a client requests virtual machines from a cloud provider, these virtual machines can be provisioned in various physical layouts. The client receives the virtualized resources but has no knowledge of the physical ones. Identical client requests could be satisfied using different physical resource layouts. The resource contention in each physical layout could differ substantially. Hence, this mapping can potentially impact runtime performance of the client's application. We term the variation due to ambiguity in the mapping of virtual resources to physical resources in a cloud computing environment as provisioning variation. Different application domains put different loads on available physical resources. Examples vary between compute, memory or I/O bound applications[13]. The performance of an application is limited by the resources available and the contention for these resources. When applications compete for the same physical resource, serious performance degradation may occur. We conjecture that the mapping of identical virtual resources on different physical layouts could have significant variations in physical resource contention. This could lead to

performance variation of the same application on identical virtual resources due to the variation in resource contention[14].

Consider we have a cloud environment which consists of a set of M, so data centers $DC = \{d_1, d_2, ..., d_x\}$, which are connected by links of different bandwidths. In a Data Center m number of the independent machine PMs, as **D1** = $\{pm_1, pm_2, \ldots, pm_M\}$. In a Data Center m number of the independent machine PMs, as **D1** = $\{pm_1, pm_2, \ldots, pm_M\}$. Virtualization software installed on each **PM**s that runs a set of **N** independent **VM**s represented by VM = $\{vm_1, vm_2, .., vm_N\}$

Table 1. Key Data Structure of our Proposed Model

Cloud Infrastructure	Data Centers-DC	Physical Machines - PM	Virtual Machines - VM	Virtual Machine Pool - VMP
Cloud 1	**D1, D2**	$\mathbf{PM_{0-8}}$	$\mathbf{VM_{0-100}}$	**VMP1**
Cloud 2	**D3**	$\mathbf{PM_{9-17}}$	$\mathbf{VM_{101-200}}$	**VMP2**
Cloud 3	**D4, D5,D6**	$\mathbf{PM_{17-25}}$	$\mathbf{VM_{201-300}}$	**VMP3**

4. RESEARCH CONTRIBUTION

Our approach includes a user who will request for cloud resource, different cloud providers which will provide their resources to fulfill the requests of users, a Resource Pool Manager which will manage the re- sources of various cloud providers using various resource allocation algorithms, thus helping the Resource Pool Manager for making the decision to allocate or release a VM[15]. Firstly, the user request will submit to resource request queue accessed by cloud broker some time interval. The cloud broker plays the major role here. It manage the service catalog from various cloud providers because different vendors have different management systems, different pricing models and are accessed and used in different ways. Cloud broker then request to the Resource Pool Manager for resource allocation. The Resource Pool Manager will decide which pool's resources would be appropriate to allocate to the user[17]. If required number of resources are available, manager will allocate them to the user. If the request requires additional resources, resource pool manager will submit a request to the appropriate cloud provider to provision additional resources. The resource pool manager will provide pointers and credentials of the allocated resource to the user. The pointers are usually the MAC and IP addresses and sometimes a DNS name given to the VM. The credentials are usually a pair of RSA keys (a public key and a private key, which one uses in the API to speak with the VM). Most often, the VM presents an x86 PC machine architecture. On that VM, one boots a system image yielding a running system, and uses it in a similar manner as one would use a running system in the datacenter.

Several virtual machines (VM) were deployed by using the technology call Virtualization or hypervisor (e.g. Xen, KVM, VMware), VM are configured with different properties (RAM, storage, CPU etc.) On top of the Hypervisor in which Operating System (OS) is installed like normal system. Full virtualization hypervisors provide networking capabilities, allowing the individual guest OSs to communicate with one another while simultaneously limiting access to the external physical network. The network interfaces that the guest OSs support may be virtual, physical, or both. In Network Bridging, the guest OS provides direct access to the host's network interface cards (NIC) independent of the host OS. In Network Address Translation (NAT), the guest OS provides a virtual NIC that is connected to a simulated NAT inside the hypervisor. As in a traditional NAT, all outbound network traffic is sent through the virtual NIC to the host OS for forwarding, usually to a physical NIC on the host system. In Host Only Networking the guest OS provides a virtual NIC that does not directly route to a physical NIC. Guest OSs can be configured to communicate with one another and, potentially, with the host OS. When a number of guest OSs exist on a single host, the hypervisor provides a virtual network for these guest OSs. The hypervisor may implement virtual switches, hubs, and other network devices[18].

Full Virtualization: Hypervisor controls the hardware resources and emulates it to guest operating system. In full virtualization, guest does not require any modification. Example: Kernel-based Virtual Machine (KVM). Paravirtualization: In paravirtualization, hypervisor controls the hardware resources and provides API to guest operating system to access the hardware. In paravirtualization, guest OS requires modification to access the hardware resources. Xen is an example of paravirtualization technology. We have consider four tiers of service support level in this research paper as per given below table 2.

Table 2. Service Support Level Criteria

Service Support Type	Platinum	Gold	Silver	Bronze
Response Time P1-Service Unavailable P2-Critical Impact P3-Minor Impact	P1 - 30 minutes P2 - 2 hours P3 – 24 hours	P1 - 30 minutes P2 - 2 hours P3 – 24 hours	P1 - 30 minutes P2 - 3 hours P3 - 1 business day	P1 - 30 min P2 - 1 business day P3 - 2 business days
Availability	99.999 percent & Zero planned outages	99.99 percent & Up to four hour planned outages (maintenance)	99.9 percent & Up to four hour planned outages (maintenance)	2 Business Days & Moderate outage
Monitoring	24*7*365	24*5*365	12*5	8*5
Performance	90%	90%	Not Measured	Not Measured
Dedicated Account Manager	Yes	No	No	No

4.1 Policy Engine

We proposed a policy engine which is running on the service registry server and using the service level support database where all the resource requirement criteria for all the four tiers of service support level. The policy engine is responsible for tracking the processor, network and memory usage of each VM (Virtual Machine). It also tracks the total resource usage on each PM (Physical Machine) by aggregating the usages of resident VMs. This engine tracks the usage of each resource over a measurement interval and reports these statistics to the cloud controller at the end of each interval. This Policy engine also maintains configuration details of VMs for four types of service support level templates[19].

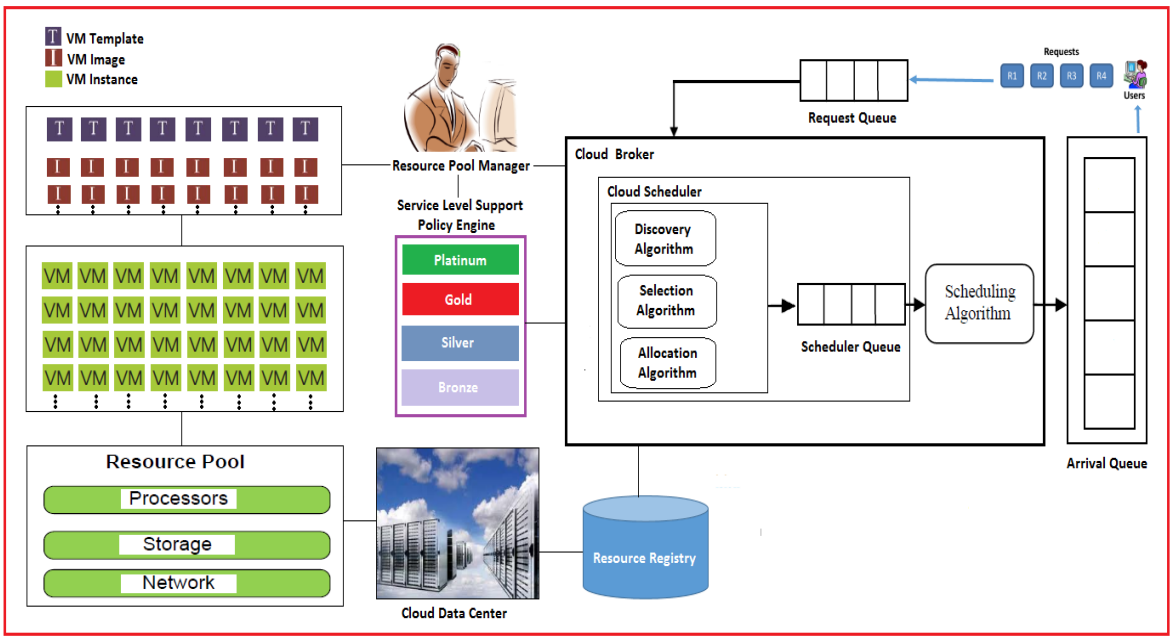

Figure 6. Function of the Resource Pool Manager

Virtual machine configuration is the arrangement of resources assigned to a virtual machine. The resources allocated to a virtual machine (VM) typically include allocated processors, memory, disks, network adapters and the user interface[20]. We have given one sample of the VM configuration file.

Resource pool management involves not only unified management, scheduling, and monitoring of resource pools, but also proper use and maintenance of the cloud platform[21]. A cloud computing management platform can be divided into four layers namely device management, VM management, service management, and tenant management.

Device management: This layer manages hardware devices on the cloud platform and raises an alarm when there is device abnormality[22]. Specifically, the system administrator uses this layer during daily maintenance to check device performance, and to monitor key indexes such as application server CPU usage, memory usage, hard disk usage, network interface usage, storage space availability, and Input/output (IO) status. A user can set the monitor threshold for a physical device based on actual configuration. The system then automatically starts monitoring and raises an alarm when the threshold is exceeded.

VM management: The virtual resource management layer implements unified management, allocation, and flexible scheduling of VM for various applications. As in the device management layer, the system administrator checks the performance of VM's daily maintenance, and monitors key indexes—such as CPU usage, memory usage, hard disk usage, network interface usage, virtual storage availability, and IO status—of virtual machines being used. A user can set the monitor threshold for a VM based on actual configuration. The system then automatically starts monitoring and raises an alarm when the threshold is exceeded.

Service management: The service management layer manages service templates, service instances, and service catalogs. Based on VM's, it promptly provides user-specified operating system and application software to tenants.

Tenant management: The tenant management layer manages the resource clusters of all tenants. Resource type, quantity, and distribution are managed, as well as tenant lifecycle—from application, examination, and normal operation, to suspension to cancellation.

When a user requests for a VM, along with the request it presents a set of attributes that should be satisfied by the VM. First the user request will store in the request queue then it forward to the resource broker. The resource discovery process may be responsible for generating a set of best possible candidates for the given set of attributes using the policy engine. Resource discovery step use the resource registry database that are maintained by policy engine to fulfill the user requests. The optimization of the current VM allocation is carried out in two steps: at the first step we select VMs that need to be allocate and in second step the chosen VMs are placed on the arrival queue using scheduling algorithm. To determine when and which VMs should be allocated using the resource allocation algorithm[23].

Cloud Broker maintains an index table of all Data Centers indexed by their VM Pool. When the Cloud broker receives a request from a resource request queue it sends this to destination Cloud Controller. Cloud Broker retrieves the region of the user of the request and queries for the region list for that region from the Index table. This list orders the remaining regions in the order of lowest network latency first. The Cloud Broker picks the first VM Pool mapped to the data center located at the earliest/highest region in the index table[24]. If more than one VM pool mapped to the data center is located in a region, the data center having less cost (here, considering only VM cost) will be selected. Now user request will be sent to this most cost effective VM Pool mapped to data center. As per this strategy, the data center selection is not made randomly and vm cost in each data center is compared with other data centers in the same region. The data center with lowest vm cost is selected. Now the user requests will be sent to this data center to be processed.

4.2 Network Configuration

Communication service handles the networking issues of all VMs in the system. It is responsible for initializing the local sub-net of a VM pool and allocates MAC addresses to newly created VMs[25]. It keeps tracking of the allocation of IP addresses in the subnet. If more IP addresses are needed, it will generate more items in the IP address and MAC address mapping table and refresh the dynamic host configuration protocol (DHCP) server on each VM pool. In addition, it controls how network packages are sent, routed, and forwarded between VMs that cross different subnet to perform virtual LANs management. In this way, VM users are unaware of the underlying network infrastructure and can be managed based on the sub-network basis.

The initial stage of VMs network configuration is the creation of VM images. The networking infrastructure consists of a mixture of virtual entities and physical entities. Virtual entities include VMs, vSwitches, VLAN taggers, VPN, and gateways. Physical entities include the physical hosts and the physical networking infrastructure, which includes VXLAN-enabled physical switches, routers, and ordinary Ethernet switches. Virtual Ethernet cards or vNICs are the basic building blocks of our design. Each VM can have one or more vNICs[26]. Each vNIC can be associated with at most one VXLAN. Each VXLAN segment is represented by a virtual switch or a vSwitch. A VM appears on a particular VXLAN if one of its vNICs is "plugged" into one of the switch ports on the vSwitch forming that segment. The vSwitch behaves like a

normal physical switch. Ethernet broadcast traffic generated by a VM connected to the vSwitch is passed to all VMs connected to that vSwitch. Like a real switch, the vSwitch also builds up a forwarding table based on observed traffic so that non-broadcast Ethernet traffic can be delivered in a point-to-point fashion to improve bandwidth efficiency. The vSwitch is designed to operate in a distributed fashion. The VMM on each physical machine hosting a VM connected to a particular VXLAN segment hosts part of the vSwitch forming that VXLAN segment[27].

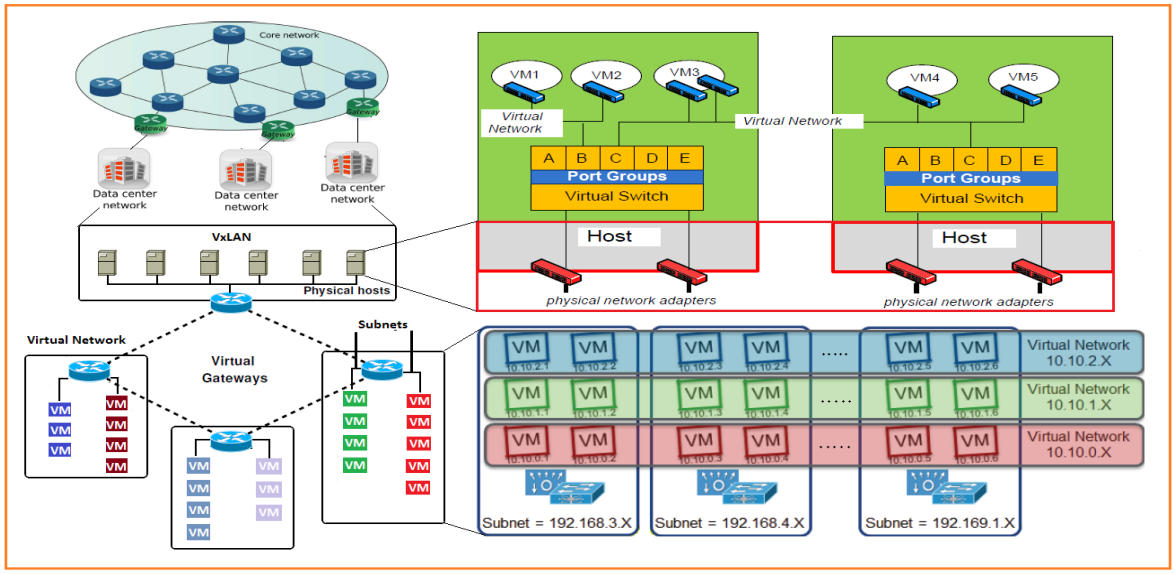

Figure 7. Network Architecture of our Proposed Model

VXLAN (Virtual extensible LAN): VXLAN solves the problem of multiple tenants in a cloud environment i.e communication between VMs of different subnets. It enables the connection between two or more L3 networks and makes it appear like they share the same L2 subnet. This now allows virtual machines to operate in separate networks while operating as if they were attached to the same L2 subnet.

IP Address: set a static IP address to the VM. IP addresses are assigned sequentially and all students of the same year share the same subnet. For the next year, IP addresses will be in the form of 10.0.1.x; Simple socket-outsourcing appears to be like the network address translation (NAT) mode of regular hosted virtual machines[29]. This means the guest OS shares the same IP addresses with the host OS. Occasionally, we need to allocate one or more dedicate IP addresses to each VM instance. To accomplish this, we add these IP addresses to network interfaces in the host in advance. When a guest process creates a server socket and assigns the IP address with system call bind (), we enforce the address by restricting the arguments of system call bind (). When a guest process initiates a network connection as a client, we enforce the source IP address with system call bind () in the host module even though the guest process does not issue bind () in the guest OS[30].

The vSwitch is designed to operate in a distributed fashion. The VMM on each physical machine hosting a VM connected to a particular VXLAN segment hosts part of the vSwitch forming that VXLAN segment.

Netmask: set the netmask, allowing students to communicate only with VMs owned by students of the same year. In particular, they can communicate only with machines that share the same subnet;
Gateway: set the VM common gateway. The gateway is responsible for the management of the packets through

Subnet: The gateway IP is the same than the VMM / Hypervisor bridge, a virtual network interface that supervises all communications between VMs and the external net[31].

4.3 Data Flow Architecture of our Proposed Model

In our proposed resource allocation scheme the Inter-cloud, cloud-to-Internet and Internet-to-cloud data movement starts from data collected from cloud users through cloud services namely Software-as-a-Service (SaaS) and Platform-as-a-Services (PaaS). We have taken an example to explain how data flows

among different levels starting from users to cloud resource providers. Various cloud services like word processor, web-based emails etc specific to a VM where provenance is collected then data specific to a PM where provenance is collected, such as the physical resources status, the physical address of memory/block devices/files, and the mapping of VMs to PMs, then the data related to a cloud, such as customer identification information and intra-cloud networking which is communication between different PMs within a cloud, transfers of data across VMs and PMs located across different geographies in a cloud. Finally data sharing on the Internet, including data transfer between different clouds, and from a cloud to any network node that does not belong to a cloud, which gives the broadest granularity[32].

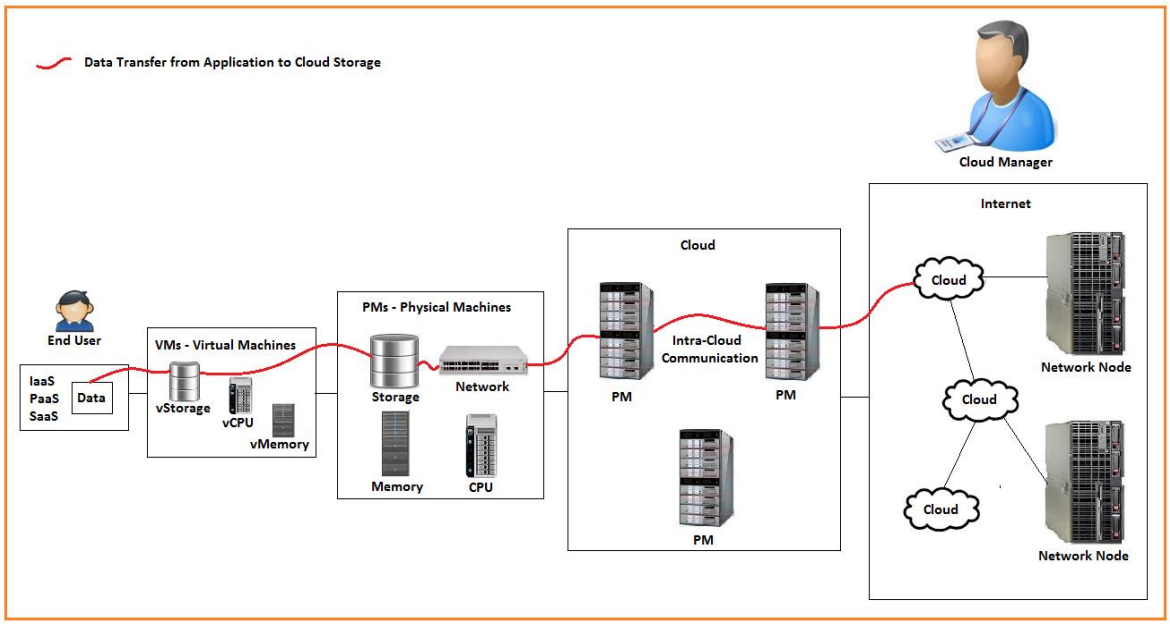

Figure 8. Data Flow Architehture of our Proposed Model

4.4. Key Data Structures & Proposed Algorithms

The resource request queue'RRQ' and resource arrival queue 'RAQ'.
Requests RRQ = {RRQ_1, RRQ_2, RRQ_3, RRQ_4…….. RRQ_n}

Key Data Structure: Consider we have a cloud environment which consists of a set of 'M' PM's (Physical Machines, s o data centers DC ={d1, d2 ,...,dx }, which are connected by links of different bandwidths. In a Data Center m number of the independent machine PMs, as D1 = {pm1, pm2,…, pmM} Virtualization software installed on each PMs that runs a set of N independent VMs represented by VM = {vm1, vm2,.., vmN}. Virtual Resource Pool (VMP) is the set of logical resources in the form of VMs. Let the quantity of logical resources be VMP then we take VMP= {VMP1, ¬VMP2, VMP3….¬VMPn}.

We propose the following five algorithms

VM Creation & Deletion

VM Creation
Step 1: VMM analyses the task and asks for resources required to create VMs from Res_MGR.
VMM_SendRequest [Resources] → Res_MGR
Step 2: Res_MGR checks **RAL**
IF (Resources_available)
Then Provide access to VMM and Exit
ELSE
Continue
Step 3: Res_MGR checks availability of resources with Cloud_P
Step 4: Cloud_P sends resource availability information to Res_MGR
Step 5: Res_Pro selects efficient resource based upon *cost factor (CF)*.
CF = RC+ LCC
Step 6: Res_Pro requests Cloud_P for selected resources.

Step 7: Cloud_P grants access permission to use the resources to Res_MGR
Res_MGR [Resource_Access] ← true
Step 8: Res_Pro updates the VMP
RP → updates [VMP]
Step 9: Res_Pro, based upon the Performance Factor (PF), provides access to resources to VMM.
Step 10: VMM creates VMs
VMM→ creates [VMs]
Case B: VM Deletion
Step 11: Check VM_Load over VMM in VM Pool
IF VM_Load increases
Then goto Step 1
ELSE IF VM_Load decreases
Then
a) VMM searches for VMs with **least PF** and **higher ET**
b) VMM redirects the cloudlets of that VM to other VMs
c) VMM deletes that particular VM.
Step 12: VM executes the cloudlets till the task is completed
Step 13: VMM returns the access to Res_MGR
Step 14: Res_MGR updates VMP

Resource Discovery

```
while (true) {
// user request  incoming message
receive messsage (X) {
// validate the incoming message: this may depend on the policy
// if universal awareness this function is always true
// if neighborhood awareness returns true only
// if the distance to source is less than m
// if distinctive awareness returns true only if the local cloud potential
// is less than or equal to remote induced local cloud potential
if (validate(X)) {
// update the data structures that keep awareness information in the VMP
process(X)
}
// if there are no incoming message then break out the loop to send messages
} or timeout (n)
if (currentTime > lastSentTime + n) {
lastSentTime = currentTime
// send to logical neighbors
get the VMP details
for each Pool in VMP
send status update message
}
}
```

Resource Selection

```
Step 1: Get the VM Pool mapped to datacenter index of selected region
Step 2: regionaList ← regionalDataCenterIndex.get (region) // store regionlist of selected
datacentre
Step 3: if regionaList is not NULL then
Step 4: listSize ← size (regionalist)
Step 5: if listSize is 1 then
Step 6: dcName ← regionalist.get(0)
Step 7: else
Step 8: for all p in dcVMPCostList do
Step 9: if(dcVmCostList.get (smallest)>dcVMPCostList.get (p)) then
Step 10: smallest=p;
Step 11: end if
Step 12: end for
Step 13: dcName ← regionalist.get (smallest)
Step 14: end if
Step 15: end if
Step 16: return dcname
```

Resource Allocation

```
Step 1: Requested are collected pre-determined interval of time.
Step 2: Resources RSi -> {RS1,RS2, RS3,…,RSn}
Step 3: Requests RRQi -> {RRQ1, RRQ2, RRQ3,…, RRQn}
Step 4: Threshold (static at initial)
Step 5: Th= ∑RSi
Step 6: for every unsorted array A & B
Step 7: sort A & B
Step 8: for every RRQi
```

```
Step 9: if RRQi<Th then
Step 10: add RRQi in low array, A[RRQi]
Step 11: else if RRQi>Th then
Step 12: add RRQi in high array B[RRQi]
Step 13: for every B[RRQi]
Step 14: ** allocate resource for RRQi of B
Step 15: RSi= RSi-RRQi; Th= ∑RSi
Step 16: satisfy the resource of A[RRQi]
Step 17: for every A[RRQi]
Step 18: ** allocate resource for RRQi of A
Step 19: RSi= RSi-RRQi; Th= ∑RSi
Step 20: satisfy the resource of B[RRQi]
```

** Best fit strategy is used to satisfy the request alternatively in A[RRQi] and B[RRQi] based on the available VM.

Resource Scheduling

If we use *PMi* to denote the *ith* physical resource, *m* to denote the total number of physical resources and *PM* to denote the set of physical resources, we can get *PM={PM1, PM2, PM3, ..., PMm}*. Similarly, if we use *vmi* to denote the *ith* virtual machine, *n* to denote the total number virtual machine and *VM* to denote the set of virtual machine in the extended virtual machine system, we can get *VM={vm1, vm2, vm3, ..., vmn}*.
M the total number of physical machines in the extended virtual machine system
PM_i the physical resource of the i_{th} physical machine
$vm_i \rightarrow PM_j$ the i_{th} virtual machine schedule physical resource PM_j
t_0 the beginning time that VMs are processed

```
Step-1: Choose the type of Scheduling
1. Priority Based First Come First Serve Scheduling
2. Priority Based Round Robin Scheduling
User enters the choice.
Choice == 1 || 2
Step-2: If choice is 1.
Then
goto Label 1.
If choice is 2.
Then
goto Label 2.
Step-3: Executed VMs are returned to Cloud Co-ordinator.
CC [VM_list]  VMM [executed_VM_list]
Step-4: Cloud Co-ordinator combines all the VMs.
Combine [VMs]
Step-5: Executed VM returned back to User by Cloud Co-ordinator.
User[executed VM]  combine[VMs]
Label 1: FCFS Scheduling
For every VMs received
Executed_list[VMs]  Execution [Actual_VM]
VMM[executed_VM_list]  Executed_list[VMs]
goto Step3
Label 2: Round Robin Scheduling
[initialize] time quantum as tq = 10
Repeat Steps till (actual_VM_list).Length == NULL
{
Execute[actual_VM] till tq
Executed_list[VMs]  Execution[Actual_VM]
actual_VM  VM_list[next_VM]
}
VMM [executed_VM_list]  Executed_list [VMs]
goto Step3
```

5. DESIGN OF OUR PROPOSED MODEL

5.1 Allocation of VMs for user request

- Cloud provider provide details of available VMs to Resource Pool Manger.
- Resource Pool Manger creates a VM Pool containing details of available VMs of Cloud provider.
- Cloud broker asks for VMs from the Resource Pool Manger by sending the requirements.
- Resource Pool Manger checks the VM Pool.
- If VMs are available in VM Pool, then Resource Pool Manger provides access to VMs.

- If VMs are not available in VM Pool, then Resource Pool Manger checks availability of resources with Cloud provider.
- Cloud providers, in return, sends VM availability acknowledgements to Resource Pool Manger.
- Cloud provider selects the VMs considering the cost factor.
- Resource Pool Manger requests for the selected resources from Cloud provider.
- Cloud provider provides access to resources to Resource Pool Manger.
- Resource Pool Manger updates the VM Pool after getting access to VMs.

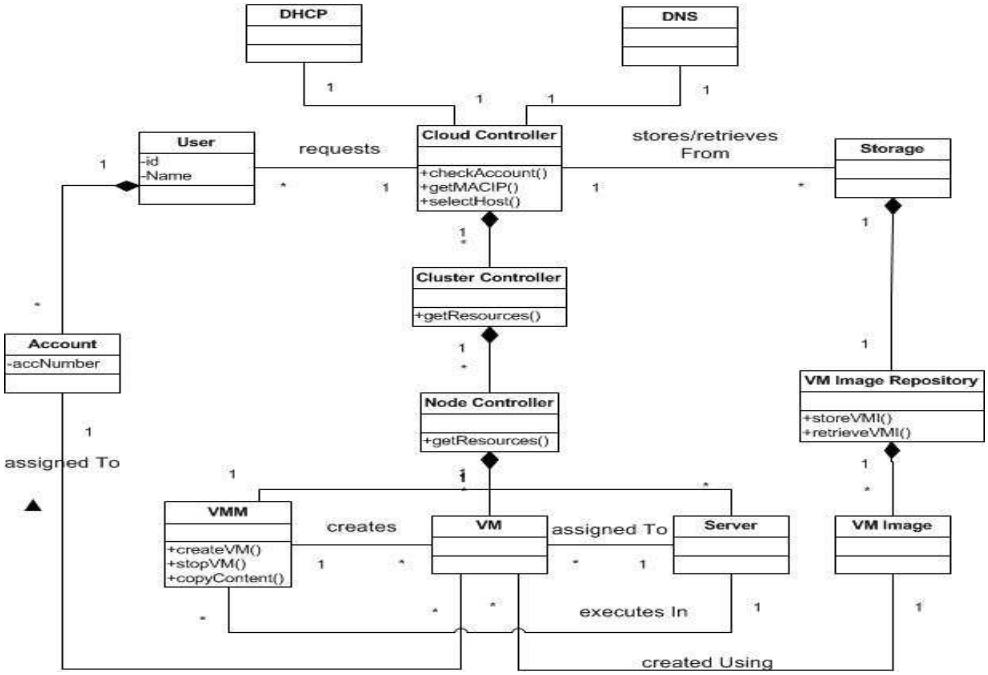

Figure 9. Class Diagram of our proposed Model

5.2 Create a Virtual Machine

- A user requests a VM with some computational resources to the Cloud Controller.
- The Cloud Controller verifies whether the requester has a valid account.
- The Cloud Controller requests the available resources to the Cluster Controllers closer to the location of the user. In turn, the Cluster Controller queries its Node Controllers about their available resources. In the sequence diagram, there is only one Cluster Controller and one Node Controller in order to maintain the diagram simple, but there can be more Clusters and Node Controllers.
- The Node Controller sends the list of its available resources to the Cluster Controller, and the Cluster Controller sends it back to the Cloud Controller.
- The Cloud Controller chooses the first Cluster Controller that can support the computational resources requirements.
- The Cloud Controller requests a MAC/IP pair address from the DHCP server for the new VM.
- The Cloud Controller retrieves a virtual machine image from the repository (VMI)
- The Cloud Controller sends a request to the Cluster Controller to instantiate a VM.
- The Cluster Controller forwards the request to the Node Controller which forwards it to the VMM.
- The VMM creates a VM with the requested resources.
- The VMM assigns the VM to one of the servers.

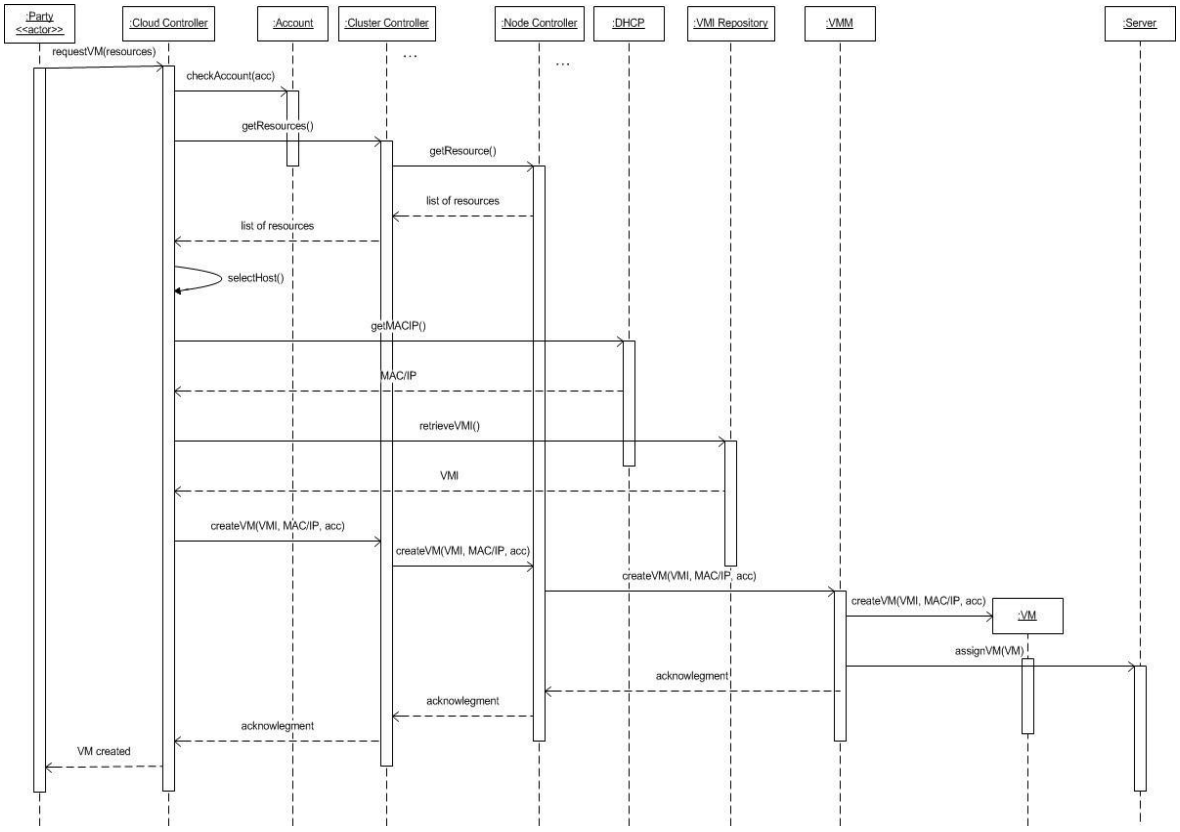

Figure 10. Sequesce diagram for create a virtual machine

6. DISCUSSION AND OPEN ISSUES

We discuss key aspects of our work and open issues. Unfortunately, there are some issues to manage resources in the cloud model. Many organizations view the cloud as a truly distributed model with multiple redundancies built in to maintain the highest uptime possible[33]. In cloud systems based on virtualization, virtual machines (VM) share physical resources. Although resource sharing can improve the overall utilization of limited resources, contentions on the resources often lead to significant performance degradation. Model Enhancements: While our paper helps better understanding of cloud resource allocation strategy, it's only a start[34]. An important feature direction understands the virtual network between multiple local and cloud data centers[35]. Handling Load balancing: An important benefit of public cloud architecture is potential to handle peak workload and handing load balance between server pools[36].

7. CONCLUSION

In Cloud computing different resource allocation strategy required for achieving user satisfaction and maximizing the profit. In this paper, we have made three contributions. First we provide general architecture & design model of cloud infrastructure for a Large IT Enterprise. Secondly, we designed a service support engine that offers cloud resource discovery, selection, allocation, scheduling using different algorithms. Finally we provide the network & Data Flow architecture of our proposed model. In our feature work we will investigate more scheduling algorithms with respect to allocation and utilizing resources. Furthermore we will study maintaining quality of service while provisioning resources. Also in feature we hope to gain experience with our approach using a wider range of enterprise cloud applications.

REFERENCES

[1] W. Chung, R. Chang, "A new mechanism for resource monitoring in Grid computing", Future Generation Computer Systems ,Vol. 25. No.1.,2009,pp. 1-7.
[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz,A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I.

Stoica,and M. Zaharia. "Above the Clouds: A Berkeley View of Cloud Computing" . Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
[3] I. Foster, Y. Zhao,I. Raicu, S. Lu, S." Cloud computing and grid computing 360-degree compared", Grid Computing Environments Workshop,2008, pp. 1–10.
[4] Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2
[5] A. Ernst, H. Hiang, M. Krishnamoorthy," Mathematical programming approaches for solving task allocation problems", Proc. of the 16th National Conf. Of Australian Society of Operations Research, 2001.
[6] G.H. Chen, J.S. Yur, "A branch-and-bound-with-derestimates algorithm for the task assignment problem with precedence constraint", Proc. of the 10th International Conf. on Distributed Computing Systems, 1990, pp. 494– 501.
[7] Zs. Németh, V. Sunderam, "Characterizing grids: Attributes, definitions, and formalisms", Journal of Grid Computing,Vol. 1 . No.1,2003,pp. 9-23.
[8] A. Abraham, H. Liu, M. Zhao," Particle swarm scheduling for work-flow applications in distributed computing environments, in: Metaheuristics for Scheduling: Industrial and Manufacturing Applications," in: Studies in Computational Intelligence, Springer Verlag, Germany, 2008, pp. 327-342.
[9] A. Abraham, R. Buyya, B. Nath, "Nature's heuristics for scheduling jobs on computational Grids", in: Proceedings of the 8th International Conference on Advanced Computing and Communications, Tata McGraw-Hill, India, 2000, pp.45-52.
[10] Y. Gao, H.Q. Rong, J.Z. Huang," Adaptive Grid job scheduling with genetic algorithms", Future Generation Computer Systems,Vol. 21,No. 1,2005, pp. 151-161.
[11] A. Abraham, R. Buyya and B. Nath, "Nature's heuristics for scheduling jobs on computational grids", Proc. of the 8th IEEE International Conference on Advanced Computing and Communications, India, 2000,pp.45-52.
[12] H. Liu, A. Abraham," An hybrid fuzzy variable neighborhood particle swarm optimization algorithm for solving quadratic assignment problems", Journal of Universal Computer Science, Vol.13, No.7, 2007, pp. 1032-1054.
[13] P.Y.Yin, S.S. Yu,P.P. Wang, and Y.T. Wang," A Hybrid Particle Swarm Optimization Algorithm for Optimal Task Assignment in Distributed System", Computer Standards & Interfaces, Vol. 28, 2006, pp. 441-450.
[14] P. Ruth, X. Jiang, D. Xu, and S. Goasguen. "Virtual distributed environments in a shared infrastructure". Computer, Vol. 38, No. 5, 2005,pp.63–69.
[15] ChunyeGong,Jie Liu,Oiang Zhang,Haitao chen and Zhengh Gong, "The Characteristics of Cloud Computing", in Parallel Processing workshops, 2010, pp 275-279.
[16] Pushpendra Kumar pateria, Neha Marria, "On-Demand Resource Provisioning in Sky Environment," International Journal of Computer Science and its Application, 2010, pp 275-280.
[17] Zhang Yu Hua, Zhang Jian, Zhang Wei Hua, "Discussion Of Intelligent Cloud Computing System," International Conference on Web Information Systems and Mining, 2010, pp 319-322.
[18] Jianfeng Zhan, Lei Wang, Bipo Tu,Yong Li, Peng Wang, Wei Zhou and Dan Meng, " Phoenix Cloud: Consolidating Different Computing Loads on Shared Cluster System for Large Organization",In Proceeding of the First Workshop of Cloud Computing and its Application,17 july 2010.
[19] Shu-Ching Wang,Kuo-Qin Yan, Wen-Pin Liao and Shun- Sheng Wang, " Towards a Load Balancing in a Three-level Cloud Computing Network" ,.
[20] Hu, Johnny Wong, Gabriel Iszlai and Marin Litoiu, "Resource Provisioning for Cloud Computing," Conference of Center For Advanced Studies on Collaborative Research, 2009.
[21] M. Noureddine and R. Bashroush, "Modality cost analysis based methodology for cost effective datacenter capacity planning in the cloud," Special Issue on The International Conference on Information and Communication System, pp 1-9. (ICIC 201).
[22] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato,"Optimization of Resource Provisioning Cost in Cloud Computing," pp 1-32.
[23] Gihun Jung, Kwang Mong Sim, "Agent-based Adaptive Resource Allocation on the Cloud Computing Environment," at Parallel Processing Workshops (ICPPW),2011 40th International Conference, pp 345-351.
[24] M. Suhail Rehman Majd F. Sakr, "Initial Findings for Provisioning Variation in Cloud Computing," International Conference on Cloud Computing(2010), pp473-479.
[25] Dongwan Shin and Haken Akkan, "Domain based Virtualized Resource Management in Cloud Computing".
[26] Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, "Independent Tasks Scheduling Based on GeneticAlgorithm in Cloud Computing", IEEE Computer society,978-1-4244-3693-4,2009.
[27] Hai Zhong, Kun Tao, Xuejie Zhang, "An Approach to Optimized Resource Scheduling Algorithm or Open-source Cloud Systems" , IEEE Computer Society, 978-0-7695-4106-8, 2010.
[28] B. Sotomayor, K. Keahey, I. Foster," Combining batch execution and leasing using virtual machines", ACM 17th International Symposium on High Performance Distributed Computing, pp. 87-96,2008.
[29] Daniel Warneke, Member, IEEE, and Odej Kao," Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud", IEEE Transactions On Parallel And Distributed Systems, vol. 22, No. 6, JUNE 2011.
[30] P. Ruth, P. McGachey, D. Xu," VioCluster: virtualization for dynamic computational domain", IEEE International on Cluster Computing, pp. 1-102005.
[31] Anirban Kundu, Chandan Banerjee, Sutirtha Kr. Guha, Arnab Mitra, 4Souvik Chakraborty, Chiranjit Pal, Rahul Roy," Memory Utilization in Cloud Computing using Transparency".
[32] Alexandru Iosup , Simon Ostermann, M. Nezih Yigitbasi, Radu Prodan Thomas Fahringer Dick H.J. Epema,"Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing", IEEE Transactions On Parallel And Distributed Systems, Vol. 22, No. 6, June 2011.

[33] Sandeep Tayal,"Task scheduling optimization for the cloud computing systems", International Journal Of Advanced Engineering Sciences And Technologies Vol No. 5, Issue No. 2, pg no:111 – 115.
[34] Paul Marshall, Kate Keahey and Tim Freeman,"Improving Utilization of Infrastructure Clouds", IEEE /ACM International Symposium on Cluster, Cloud and Grid Computing, 2011.
[35] Ying Zhang, Gang Huang, Xuanzhe Liu, and Hong Mei," Integrating Resource Consumption and Allocation for Infrastructure Resources on-Demand", IEEE 3rd International Conference on Cloud Computing,2010.
[36] Yuan Yuan, Wen-Cai Liu, "Efficient resource management for cloud computing", International Conference on System Science, Engineering Design and Manufacturing Informatization, 2011.

BIOGRAPHY OF AUTHORS

Rabi Prasad Padhy is currently working as a Senior Software Engineer - Oracle India Private Ltd. Bangalore, Karnataka, India. He has achieved his MCA degree from Berhampur University. He carries more than 8 years of extensive IT Experience with MNC's like EDS, Dell, IBM and Oracle. His area of interests includes IT Infrastructure Optimization, Virtualization, Enterprise Grid Computing, Cloud Computing and Cloud Databases. He has published several research papers in national and international journals. He is a certified professional for Oracle & Microsoft Database, Oracle Apps, Oracle Cloud, Linux & Solaris System Admin and also ITIL Certified.

Dr. Manas Ranjan Patra holds a Ph.D. Degree in Computer Science from the Central University of Hyderabad, India. Currently he is an Associate Professor in the Post Graduate Department of Computer Science, Berhampur University, India. He has about 24 years of experience in teaching and research in different areas of Computer Science. He had visiting assignment to International Institute for Software Technology, Macao as a United Nations Fellow and for sometime worked as assistant professor in the Institute for Development and Research in Banking Technology, Hyderabad. He has about 90 publications to his credit. His research interests include Service Oriented Computing, Software Engineering, Applications of Data mining and E-Governance. He has presented papers, chaired technical sessions and served in the technical committees of many International conferences.