

An Intelligent VM Placement for Optimal Resource Utilization in the Cloud Computing Environment

¹P. Aruna and ²S. Vasantha

¹Department of Computer Technology and Applications,

²Department of Physics, Coimbatore Institute of Technology,
Coimbatore, 641014 Tamil Nadu, India

Abstract: Cloud computing is the emerging technology that enables the user to avail the IT resources on pay for use basis. It facilitates the user to access the IT resources, without spending on capital expenses, through internet in an on demand manner. The users access the required resources by creating virtual machines on the physical machines available at the provider's site. This VM provisioning gains importance as the appropriate physical machine is to be selected for launching of the virtual machines. It would be appreciated by the users, if the VM Scheduling algorithm considers the user's placement requirements (co-location, anti co-location, trusted scheduling, etc.) too besides the VM configuration. At the same time efforts should be made to utilize the resources in an effective manner for the benefit of the providers. In this study, researchers present the study of VM Scheduling algorithms used in the private cloud define an architectural framework and policies for VM provisioning to make it an efficient one for the users by considering their placement constraints propose a solution to utilize the resources effectively by schedule or co-locate a mix of compute-intensive, memory-intensive and bandwidth-intensive VMs on a host in an intelligent manner.

Key words: Bandwidth, co-location, VM, physical machine, VM configuration

INTRODUCTION

The NIST definition of cloud computing is cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models and four deployment models. The characteristics of cloud as listed by NIST are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. The service models of cloud computing includes Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Mell and Grance, 2011).

SaaS provides software that is deployed over the internet. With SaaS, a provider licenses an application to customers either as a service on demand, through a subscription, in a "pay-as-you-go" model, or (increasingly) at no charge when there is opportunity to generate revenue from streams other than the user such as from advertisement or user list sales. SaaS is a rapidly

growing market as indicated in recent reports that predict ongoing double digit growth (Cooter, 2010; Williams, 2010). Platform as a Service (PaaS) brings the benefits that SaaS brought for applications but over to the software development world. PaaS can be defined as a computing platform that allows the creation of web applications quickly and easily and without the complexity of buying and maintaining the software and infrastructure underneath it (Rackspace, 2011). Some examples of PaaS include Google App Engine, Microsoft Azure Services and the Force.com platform. Infrastructure as a Service (IaaS) is a way of delivering cloud computing infrastructure-servers, storage, network and operating systems as an on-demand service with the help of virtualization. Rather than purchasing servers, software, data enter space or network equipment, clients instead buy those resources as a fully outsourced service on demand (Diversity Limited, 2010). Some examples of IaaS include Amazon Web Services and Rackspace (2011).

As depicted in Fig. 1 among the major three service models offered by the cloud computing, infrastructure as a service is the foundation for the other two layers. Infrastructure stands for the computing resources (servers, memory, network and storage). In cloud virtual

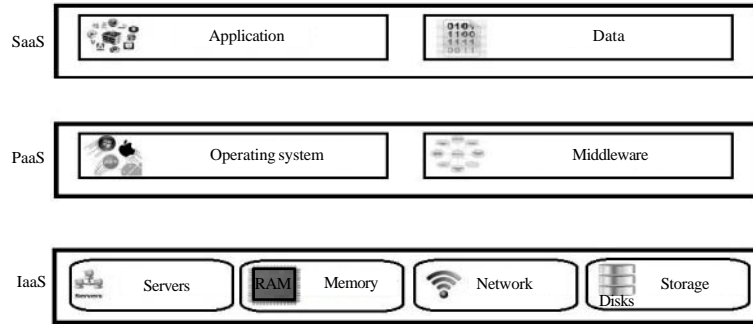


Fig. 1: Service models of cloud computing

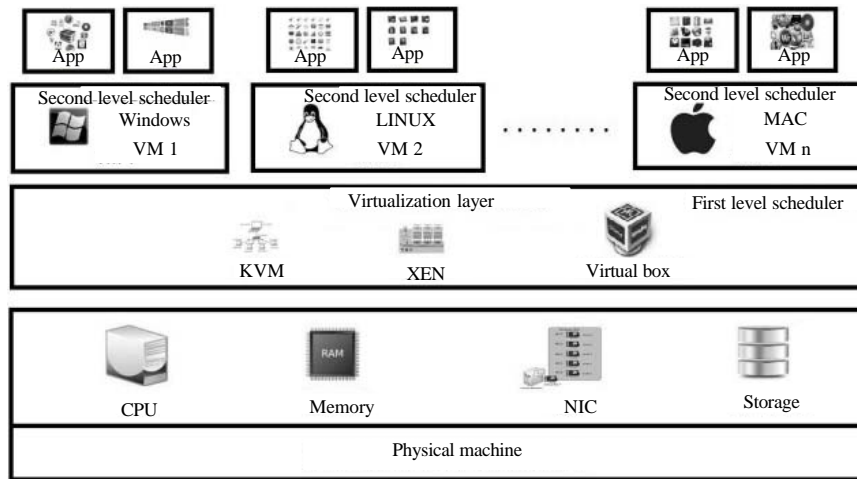


Fig. 2: Virtualization

computing, the computing power is offered via machines, storage is offered via virtual disk and network is offered via virtual switches. This study focuses on the infrastructure as a service cloud which facilitates the customers to work with the servers in a virtualized environment.

Virtualization is the technique which enables the cloud computing technology an existing one. Cloud provides rapid provisioning of virtual machines on the physical machines and allows scaling up and down quickly with the help of virtualization. Different kinds of servers can be run in the same platform known as server consolidation, making the better utilization of resources. VT enabled machines with hypervisor software provides an environment for virtualization. On top of the hypervisor virtual machines can be created each with different platform and for different purposes. The hypervisor acts as a middleware between the physical machine and the virtual machine, virtualizes the resources available in the physical machine known as the host, to

make it available for the virtual machine. Thanks to virtualization and consolidation techniques, applications can be embedded in Virtual Machines (VMs) and deployed on demand on a reduced number of nodes (Ristenpart *et al.*, 2009). Figure 2 depicts the concept of virtualization.

VIRTUAL MACHINE PROVISIONING

The main benefit of virtualization is better resource utilization; it can be obtained effectively through a good virtual machine provisioner available in the virtualization layer. Virtual machine provisioning is mapping a virtual machine to a physical machine according to the user's requirement and the resource availability. According to the objective specified, the criterion is to place the virtual machine on a physical machine.

There is a basic difference between the traditional job scheduling algorithms and Virtual Machine Scheduling (provisioning) algorithms. Scheduling decides the order

of execution of jobs waiting in the queue according to some specification. In cloud, virtual machine provisioning is assigning a physical machine to the virtual machine. In Scheduling algorithms while deciding the order of jobs those are to be allocated to the resources; each job may be waiting state until it gets the resources. But in contrast to this traditional scheduling in virtual machine provisioning waiting time for the creation of virtual machine is not entertained since the cloud has invariant number of resources. It is not placing the virtual machine in sequencing instead choosing the best host for the virtual machine to launch on depending upon some objective. The objective could be from the user's perspective or provider's perspective.

Cloud user's perspectives:

- On demand resource provisioning
- Transparent elasticity of resources
- Quality of service, no disruption of services

Cloud provider's perspectives:

- Better resource utilization/eliminate the unused resource holes occurrences in data centers
- Minimizing the power consumption
- Maximizing the revenues

As depicted in Fig. 2, virtualization introduces two level scheduler one from the virtual machine monitor where vcpus are mapped to the physical cpus the other from the guest OS where the tasks are scheduled to vcpus. While mapping a virtual machine to a physical machine, the assignment can be made according to the optimization criteria like compute intensive, memory intensive, affinity aware, power aware, load balanced, to minimize the migration latency, high performance, etc.

A STUDY ON VM PROVISIONING IN PRIVATE CLOUD BY IMPLEMENTING OPENSTACK

Cloud can be classified into four categories such as public, private, community and hybrid. Among the many available deployment models of cloud computing, the most widely used ones are public and private clouds. Both of them are using the same underneath technologies but differed in the nature of demand, availability of resources and provisioning of resources. Public cloud is beneficial for the start up industry to cut down its capital expenses. But for the institution which have already invested their money in the infrastructure they can utilize their resources efficiently with the help of private cloud setup. Whenever, there is any demand for higher end resources, they could do cloud bursting (hybrid cloud) to

meet the demands. This implies that they need not invest the money on infrastructure once again. Nowadays the private cloud is getting popular, e.g., virtual classroom, virtual desktop to make better utilization of resources already they have invested in.

Private cloud implementation: In the previous research (Aruna *et al.*, 2013; Devi *et al.*, 2013), researchers have implemented two different private cloud setup using the Open Source Cloud Operating System OpenStack and Eucalyptus to explore the VM Scheduling algorithms. To deploy a minimal cloud infrastructure, at least two dedicated systems are needed: a front end and one or more node (s) with minimum 2 GB memory. The scalability of the cloud can be extended through adding any number of compute nodes with the cloud controller. Figure 3 gives the implementation diagram of the cloud which researchers have implemented. Table 1 describes the hardware details of the private cloud setup researchers implemented. The methods of VM Scheduling algorithms used by major cloud service providers are proprietary. Hence, to explore VM scheduling in data centres, researchers will refer to the Scheduling algorithm implemented in OpenStack (Aruna *et al.*, 2013). OpenStack is Open-Source Cloud Software supported by thousands of developers, researchers and the open-source community, in addition to hundreds of leading companies.

VM scheduler in OpenStack (nova_scheduler): Open Source Software for setting up a private cloud is widely available. OpenStack, Eucalyptus, OpenNebula are the main competitors in the private cloud area. Among them OpenStack has a more user-friendly GUI. OpenStack is a cloud computing project for providing the services. It was founded by NASA and RackSpace in 2010. It is a Cloud Operating System, a new management layer that adds automation and control for pool of resources such as computation, storage and network, empower administrators and users via self-service portals and empower developers to make apps cloud-aware via APIs. The project is managed by the OpenStack Foundation, a non-profit corporate entity established in September 2012 to promote, protect and empower OpenStack Software and its community. Though, researchers have implemented the private cloud using two open source technologies such as OpenStack and Eucalyptus, Eucalyptus is dual licensed and some of the features are only available in the commercial version as VMWare support and Windows guest OS. Since, OpenStack is customizable and much research works are being done in OpenStack, researchers have analysed and explored the VM Scheduling algorithms in OpenStack.

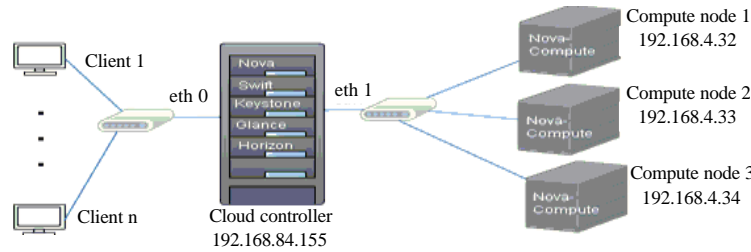


Fig. 3: Implementation diagram of OpenStack

Table 1: Hardware details used for private cloud setup

Hardware	Cloud controller (server 1) VT enabled 64 bit dual core	Compute node 1 (S2) VT enabled 64 bit dual core	Compute node 2 (S3) VT enabled 64 bit dual core	Compute node 3 (S4)VT enabled 64 bit dual core	Client VT enabled 64 bit dual core
CPU	4 GB	16 GB	4 GB	4 GB	1 BG
RAM	320 GB	500 GB	320 GB	320 GB	320 GB
Hard disk size	1 Gbp	1 Gbp	1 Gbp	1 Gbp	1 Mbp
Network					

OpenStack (Essex) has five components namely Keystone, Glance, Nova, Swift and Horizon. Keystone is used to authenticate the user, enforce policies for different users and maintain the service catalog. Glance keeps a database of metadata associated with an image for registering. It is used to retrieve and discover the image when needed. It is built on top of swift where images are stored. Glance has two services, Glance-api: public interface for uploading and managing images; Glance-registry: private interface to metadata database. Glance supports multiple image formats like iso, ami, vdi, etc. Horizon is the user interface dashboard of OpenStack.

Among the current core projects of OpenStack, Nova project is the core of the cores. The central component that manages the allocation of virtual resources to the cloud's physical resources is known as the cloud scheduler. Openstack refers to its cloud scheduler component using the name "nova-scheduler" (OpenStack, 2012). Nova-Scheduler coordinates all services, determines the placement of new resources requested.

This service is used to determine how to dispatch the compute and volume requests. The scheduler only influences the initial placement of the VMs, it does not do virtual machine live migrations. When it receives any request for VM, it chooses the compute node for the VM to be launched. The local scheduler available in the compute node creates a disk image for the VM and asks the hypervisor to boot the VM. The parameter for this call are the number of cores, the amount of memory, the hard disk image to boot from and the local CPU allocation policy. Each compute node updates its status to the cloud controller via AMQP every periodic interval (default 60 sec) or upon instance creation or deletion. There are many servers (compute nodes) available in the cloud environment for providing computing services. A virtual machine can be deployed on any of the available compute nodes from the resource pool based on the

scheduling algorithms used by the nova_scheduler. A scheduler can base its decision on various factors such as load, memory, physical distance of the availability zone, CPU architecture, etc.

Nova-scheduler incorporates many scheduling algorithms to assign a host to a virtual machine. The default algorithm for nova_compute is filter_scheduler which has two phases. In the first phase, it selects the host for the virtual machine in the requested availability zone (AvailabilityZoneFilter) which has sufficient Ram available (RamFilter) and capable of servicing the request (ComputeFilter). Then, in the second phase it finds the weight of each host by the cost function and selects the one with the highest weight. The users can configure the other algorithms available in nova-scheduler such as chance scheduler, multi scheduler and simple scheduler. Chance scheduler randomly selects a host from the list of filtered host. Simple scheduler finds the least loaded host for the virtual machine launching. This component is customizable hence the researchers can implement their algorithm and use it as a test bed.

After placing the virtual machine on the host, the virtual machine monitor on host will take care of the virtual machine life cycle management. It is like the referee between all the virtual machines and their host. Researchers have analysed the filters available in the OpenStack private cloud setup environment, there are many filters available. One among them is cpu_core_filter. By default, a single physical core can be mapped with 16 vcpus. But if researchers set cpu_core_ratio = 1, OpenStack will not allow over provisioning. The other available filters are same host filter, different host filter, CIDR affinity filter, etc. Even though the private cloud setup has rich set of schedulers, still there are some measures has to be taken.

However, all of these algorithms fail to achieve higher VMs utilization rate. Therefore, an algorithm must be used which is efficient in allocating VM's instance request and

hence increase the resource utilization. In a private cloud setup resources are not invariant like public cloud, so there should be an algorithm which incorporates the mechanism to avoid the unused resource holes scattered along many hosts in the data center.

There is no component for dynamic consolidation for virtual machine. Initial placement for virtual machine is done by nova_scheduler. Migration can be done by the administrator not triggered automatically or dynamically by the scheduler. There is a need for advanced scheduler that incorporates monitoring and support live migration. Open-source products such as Eucalyptus and OpenStack offer extremely primitive solutions as management infrastructure and lack of more complex mechanisms to enable dynamic service provisioning and rescaling.

While the users raise a VM request, many cloud system ask for the configuration of VM and its resource requirements. They are not giving priorities or consideration for VM placement constraints. If the architecture gets the information from the cloud users what for they need the virtual machine then they can suggest or choose any Scheduling algorithm according to the user's wish.

PROPOSED ARCHITECTURE

Clouds aim to drive the design of the next generation data centres by architecting them as networks of virtual services (hardware, database, user-interface and application logic) so that users can access and deploy applications from anywhere in the world on demand at competitive costs depending on their QoS requirements (Buyya *et al.*, 2009). Fitting various sized virtual machines into different sized physical machines is a challenging problem (bin packing problem). In this study, researchers have undergone the following works.

Define a macro level architectural framework and schemes for virtual machine provisioning in cloud to fulfill the objectives of both users and providers. In this framework, researchers include some schemes which can be adopted by the user for his VM placement.

Investigate and discuss the remedies for the unused resource holes occurred in the data center due to the dynamic nature of demands in resources. Propose a solution to mitigate the unused resource holes and achieve the better utilization of resources.

Suggest that if virtual machine provisioner in the virtual machine monitor has the knowledge of applications that could be run on the virtual machine, according to the nature of the applications the placement of the virtual machine can be made to the benefit of the cloud providers as well as the users. The operations mentioned in the Fig. 4 are:

- The cloud users can make their requests for the virtual machine by specifying the resources needed for the virtual machine, the operating system and the other packages. In this architecture, researchers provide different schemes for the users from which the user can select the needed schemes
- The schemes available are the user can specify the nature of application for which he needs the virtual machine, he can insist for co-scheduling of his virtual machines or he can ask that his virtual machines should be placed in two different disjoint networks. These schemes are optional, the user has the freedom to choose or omit it according to his need
- Considering the resource requirements and the specified schemes, the virtual machine provisioner places the virtual machine in any one of the host in the resource pool
- Dynamic consolidation manager monitors the status of the resource pool. It monitors for power saving, performance and fault-tolerance. It would trigger the power-saving objective without compromising the performance
- The dynamic consolidation manager initiates the virtual machine provisioner for doing the consolidation
 - According to the advice of the dynamic consolidation manager, the virtual machine provisioner reschedule the virtual machine placement

Schemes available: Many VM Provisioning algorithms are asking the users to specify what the resource requirements are but not asking much information about the jobs that indented to be run on the VM. If the purpose of the virtual machine is known then scheduling can be done in such a way to improve the cloud user's work and to increase the cloud provider's revenue by fully utilizing their resources. Here, researchers have suggested some extra options or constraints that can be opted by the users at the time of VM requisition.

Application specific: The user can request for the application specific VM, that is what type of application the user is going to work on the virtual machine. It may use any one of the resource or more resources very often. Compute intensive applications have a higher ratio of CPU than memory. Those instances are used for running cpu-bound scale out applications. For memory intensive instances, more RAM will be added up to help speed them up. Those applications move lot of data into and out of memory. Bandwidth intensive applications do data transfer very often. They need a good bandwidth.

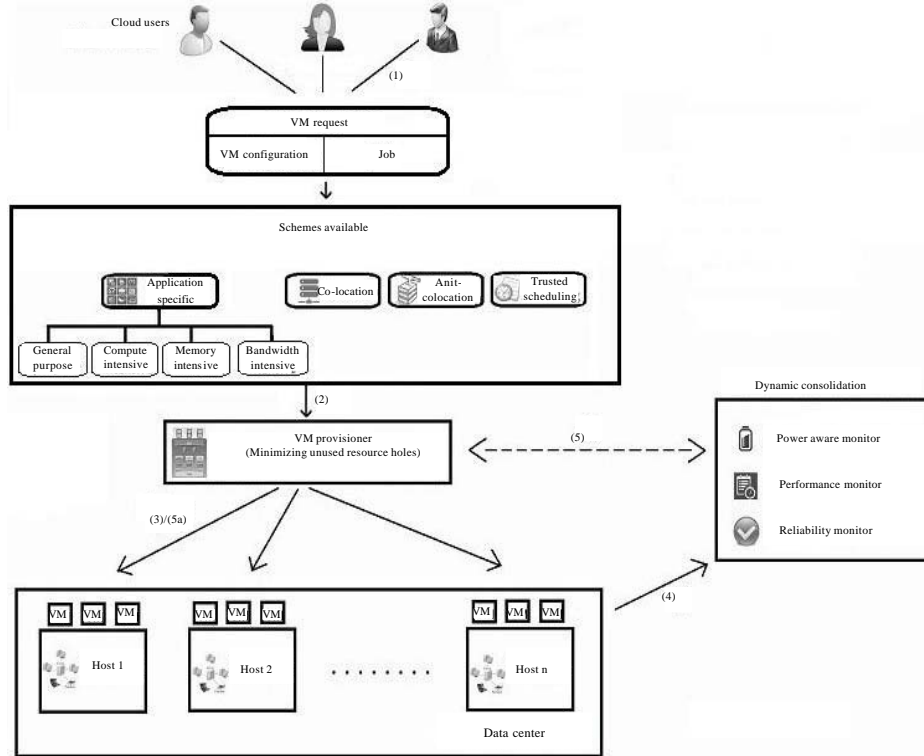


Fig. 4: Proposed architecture

Co-location: Virtual resources are hosted using physical resources that match their requirements without getting users involved with understanding the details of the cloud infrastructure. In some situation two virtual machines of the tenant can be placed in the same host if both of them are communicating with each other. So that, they can communicate through the shared memory and researchers can save the time of precious network resources and use that for other purposes. If two cloud users need their virtual machines on the same host with an acceptance letter from both of them, it can be provided. On the other hand, schedule or co-locate a mix of compute-intensive, memory-intensive and bandwidth-intensive VMs in an intelligent manner will optimize the resource utilization (Fig. 5).

Anti co-location: If a customer asks a bulk number of virtual machines, the provider can make a decision to place them in two different networks. Sometime researchers want to place set of virtual machines in two different disjoint networks in order to be sure that any one set of virtual machines are in safe state. The user may request distinct nodes for VMS running service replicas to achieve fault tolerance during failure of any resources.

Trusted scheduling: Hence, the virtualization and clouds are growing faster and faster, the capabilities of physical

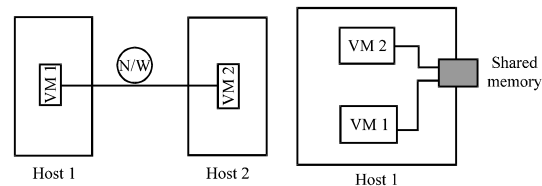


Fig. 5: Co-location

machines are increased to have thousands of virtual machines into it. Since, many VMs are residing in the single physical machine and those VMs are from different users, it is a big challenge to provide security to the VMs and there is an importance of enforcing non-interference between those VMS to avoid cross-VM attacks. Ristenpart *et al.* (2009) stated that “a customer’s VM can be assigned to the same physical server as their adversary”. If the adversary users are hosted on the same hardware, the problem will be worse. The isolation facility for the VM in the cloud should be strong, so the VM Placement algorithm should take these adversaries into account, there is a need of security-aware scheduler while implement the isolation requirements. It is always possible for a VM to explore the accounting of resources provided by the hypervisor (Ristenpart *et al.*, 2009).

Indeed, by placing VMs from adversary users on the same hardware, a cross-VM attack can occur corrupting

the integrity, confidentiality and availability of the attacked VM. Each cloud user can submit a list of adversary users with whom he does not want to share a pc. All the lists of adversary users are then merged to create incompatible groups that are taken into account when scheduling a VM placement (Afoulki *et al.*, 2011).

Dynamic consolidation and migration manager: Virtual servers do not physically exist and can therefore be moved around and scaled up (or down) on the fly without affecting the end user-arguably rather like a cloud. VM migration, especially live migration is a desirable feature in Grid and Cloud Computing Systems for load balancing, elastic scaling, fault tolerance and hardware maintenance. The ability to migrate VMs makes it possible to dynamically adjust data centre utilization and tune the resources allocated to the application environment of a virtual machine. Furthermore, these adjustments can be automated through formally defined strategies in order to continuously manage the resources in a data centre with less human intervention. VMs are loosely coupled with the PM they are running on as a result not only a VM can be started on any PM but also it can be migrated to other PMs in the data centre. In this architecture, researchers have three monitors which trigger the dynamic consolidation. The objectives of the three monitors are power-aware, performance assurance and fault-tolerance.

Power-aware monitor: One of the big challenges in data centres is to manage system resources in a power-efficient way. Data centres consume 10-100 times more energy per square foot than typical office buildings (Scheiing, 2007). They can even consume as much electricity as a city (Markoff and Lohr, 2002). The main part of power consumption in data centres comes from computation processing, disk storage, network and cooling systems. Lowering the energy usage of data centres becomes a challenging issue because computing applications and data are growing so quickly that increasingly larger servers and disks are needed to process them fast enough within the required time period (Buyya *et al.*, 2010). Thus, data centre resources need to be managed in an energy-efficient manner to drive green cloud computing. Here, researchers mention about reducing power consumption taken by the cpu with the help of dynamic consolidation and virtual machine placement. It minimizes the total energy consumption by reducing the number of active physical machines. Reducing the power consumption reduces the operating cost as well increase the resource utilization and profit.

Performance monitor: In practice, a user requests several VMs and express the Quality of Service (QoS) he expects using a SLA. Such a SLA consists in placement constraints asked for a minimum amount of resources to

ensure performance. Even more importantly, if the virtual workloads peak and need more resources, the virtual infrastructure should be able to automatically locate a host server with adequate resources and move the VMs to this host. And the entire process should be completely transparent to end users.

Reliability monitor: A VM can be migrated to other machine when one server goes down unexpectedly connection will automatically be redirected to the other server. Autonomic fault tolerance is implemented with the help of dynamic consolidation.

MINIMIZING UNUSED RESOURCE HOLES

While mapping a VM with the PM, the resources such as no of cores, RAM, bandwidth and data storage has been occupied by the VM. The VM requested by the user may have different purposes. The VM can be categorized such as general purpose, compute intensive, memory intensive, bandwidth intensive and HPC. While placing the VM to the PM, if more number of memory intensive VMs are mapped with the same PM, the other resources such as processing power, bandwidth are underutilized. If the RAM of the PM is filled even though it has other resources those could not be used by other VMs. The reason is RAM is more critical than CPU and bandwidth. Researchers cannot over commit the RAM as an application may fail due to insufficient RAM. If over committing computing power and bandwidth, the insufficiency of resources may just slow down the execution of the applications.

Sometimes, a virtual machine request cannot be met even though enough resources are in the data centre but scattered across the hosts resulting in failure of VM creation. Hence, the Cloud System needs a VM provisioner which will effectively use the available resource in host and minimize failure in VM creation. Especially, in the private cloud the major objective is to utilize the resources in a better way. If there are many unused resource holes, it cannot be achieved. To avoid this unused resource holes we can place the dissimilar VMs in the PM, so that the fairness is given to all the resources in the PM. Researchers can give a weight to the resources in the PM and the VM. If there is an imbalance in the weight, researchers will make the placement of the VM in the PM to compensate the imbalance. To overcome the problem of fragmentation, the dissimilar jobs with varying ratio of resource needs requested by the VMs could be managed in table format by tracking the resources requested by the VMs. The intensive jobs are routed to other hosts if the problem of fragmentation occurs.

In Fig. 6, VM1 is classified as memory-intensive as it having higher ratio of memory than the CPU. It is mapped

with Host 1. The next request is also a memory-intensive VM. If it is placed in the same host, the other resources are become underutilized. Researchers cannot use those resources since the RAM capacity is filled. This can be overcome using if memory-intensive VMS are placed in physical machine, either bandwidth-intensive or compute-intensive VMS

may be placed on the same physical machine to alleviate the resource fragmentation. In Fig. 7, VM 1 which is memory-intensive is placed in Host 1, again a memory-intensive request VM 2 is not placed in Host 1 as it create the resource hole. So, it is placed in Host 2, to make better utilization of resource utilization.

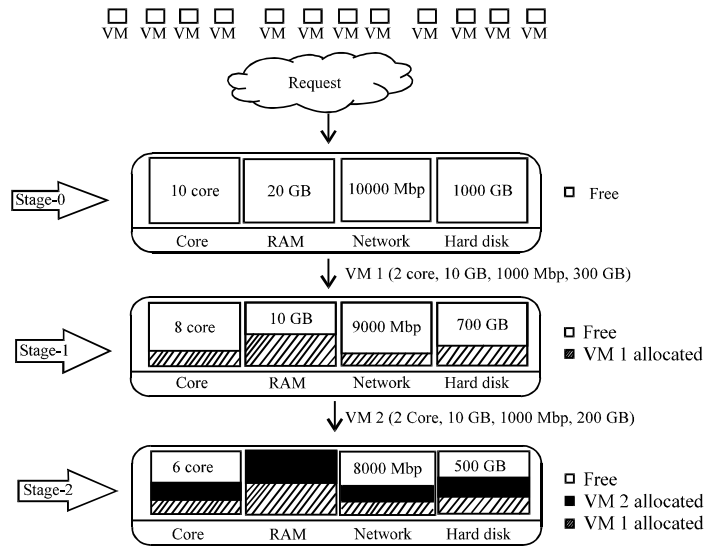


Fig. 6: Occurrence of unused resource holes

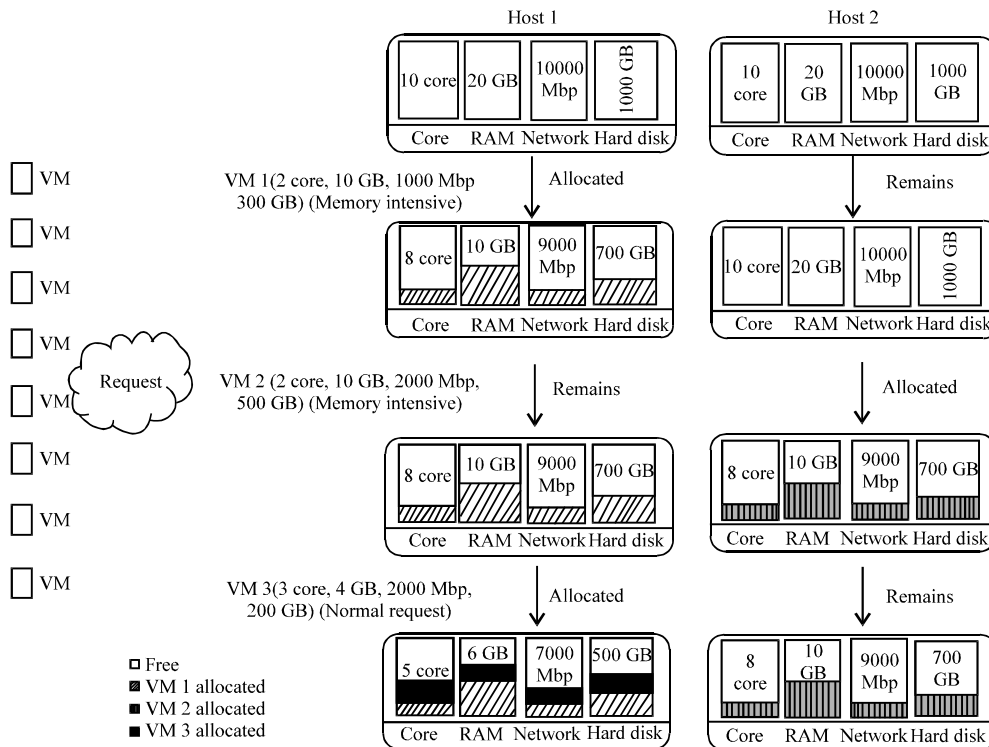


Fig. 7: Minimizing unused holes

CONCLUSION

Researchers address the cloud resource management problem which is to allocate and schedule computing resources in a way that providers achieve high resource utilization and users meet their application's performance requirements. In the previous research researchers have implemented the private cloud setup and analysing the VM Scheduling algorithms available in it. Researchers learned that by knowing the target application of the VM, VM placement can be made in an optimal manner. Unused resource holes can be avoided or minimized by co-locating different application specific VMs in an intelligent manner.

REFERENCES

- Afoulki, Z., A. Bousquet and J. Rouzaud-Cornabas, 2011. A security-aware scheduler for virtual machine on IaaS clouds. Rapport No. RR-2011-08, LIFO, France, August 2011. <http://www.univ-orleans.fr/lifo/prodsci/rapports/RR/RR2011/RR-2011-08.pdf>.
- Aruna, P., L. Yamuna Devi, D. Sudha Devi, N. Priya, S. Vasantha and K. Thilagavathy, 2013. Private cloud for organizations: An implementation for OpenStack. *Int. J. Scient. Eng. Res.*, 4: 82-87.
- Buyya, R., A. Belonglavoz and J. Abawajy, 2010. Energy-efficient management of data center resources for cloud computing: A vision, architectural elements and open challenges. *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, July 12-15, 2010, Las Vegas, NV., USA., pp: 6-20.
- Buyya, R., C.S. Yeo, S. Venugopal, J. Broberg and I. Brandic, 2009. Cloud computing and emerging IT platforms: Vision, hype and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.*, 25: 599-616.
- Cooter, M., 2010. SaaS on a tear says IDC. *NetworkWorld News*, October 18, 2010. <http://www.networkworld.com/news/2010/101810-saas-on-a-tear-says.html>.
- Devi, S.D., Y.L. Devi, K. Thilagavathy, P. Aruna, N. Priya and S. Vasantha, 2013. Private cloud in educational institutions: An implementation using UEC. *Int. J. Comput. Appl.*, 78: 8-12.
- Diversity Limited, 2010. Moving your infrastructure to the cloud: How to maximize benefits and avoid pitfalls. Diversity Limited, Christchurch, New Zealand, pp: 1-20. <http://diversity.net.nz/wp-content/uploads/2011/01/Moving-to-the-Clouds.pdf>.
- Markoff, J. and S. Lohr, 2002. Intel's huge bet turns iffy. *New York Times Technology Section*, September 29, 2002. <http://www.nytimes.com/2002/09/29/business/intel-s-huge-bet-turns-iffy.html>.
- Mell, P. and T. Grance, 2011. The NIST definition of cloud computing: Recommendations of the National Institute of Standards and Technology. Special Publication 800-145. National Institute of Standard and Technology, U.S. Department of Commerce, September 2011, Gaithersburg, MD., USA. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
- OpenStack, 2012. OpenStack compute administration manual. OpenStack™, March 11, 2012. <ftp://ftp.pku.edu.cn/open/cloud/OpenStack/doc/bk-compute-adminguide-trunk.pdf>.
- Rackspace, 2011. Understanding the cloud computing stack: SaaS, PaaS, IaaS. Rackspace US Inc., USA. http://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas.
- Ristenpart, T., E. Tromer, H. Shacham and S. Savage, 2009. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. *Proceedings of the the 16th ACM Conference on Computer and Communications Security*, November 9-13, 2009, Chicago, IL., USA., pp: 199-212.
- Scheihing, P., 2007. Creating energy efficient data centers. *Proceedings of the Data Center Facilities and Engineering Conference*, May 18, 2007, Washington, DC., USA.
- Williams, A., 2010. SaaS market grows spectacularly to the detriment of the IT kings. July 26, 2010. <http://readwrite.com/2010/07/26/sass-providers-challenge-the-k#awesm=~ox2fuv0cC4uY7m>.