ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Mutual Resource Allocation on Compute Nodes in the Cloud

Pankaj Kadam, Nilesh Alone

Post-Graduation Student, Dept. of Computer Science, R H Sapat College of Engineering, Nasik, University of Pune, Maharashtra, India

Assistant Professor, Dept. of Computer Science, R H Sapat College of Engineering, Nasik, University of Pune, Maharashtra, India

ABSTRACT: Cloud allows users to deploy their application and infrastructure in terms of virtual machine. Computing resources are made available to the users in form of virtual machines. These virtual machines are placed according to the constraints and availability of resources accessed by the cloud provider. Current virtual machine placement algorithm only considers constraints from cloud provider and do not understand the layout or virtual machine arrangement required by the end users. Current system does not allow users to mention any interdependencies in between any virtual machines hence they cannot communicate their requirements effectively to the cloud providers. In this paper we propose mutual cloud scheduler which mutually understands the requirements and constraints placed from both the cloud providers and end users. Users can pass various constraints and hints like association and dissociation of virtual machine or favoring certain required feature like backup or private networking. This allows users to place their virtual machine effectively as they know the purpose of the virtual machine and its use. Cloud administrator can also place the constraints on physical nodes, our constraints satisfaction formula assigns scores to those available nodes and then yields the final virtual machine to host mapping. We tested the system with our prototype and it shows that our mapping system always select best possible nodes satisfying the constraints from both the cloud provider and end users. We have compared our placement scheme with other well-known placements schemes like round robin, First Fit algorithm and ranking algorithm. Our analysis shows that other algorithm consumes resources on the nodes unevenly and affects application performance as they do not consider any hints from cloud users.

KEYWORDS: cloud computing, virtualization, virtual machines, placement algorithm, simulated annealing

I. INTRODUCTION

Over past 5 years cloud industry is being in demand due to its flexibility, reliability and scalability. Cloud infrastructure allows users to deploy or get the required resources very easily at very reasonable cost. Costing model of the cloud have played very important role in the popularity of the cloud. Most of the cloud providers use pay per use model which shifts cloud users investment from capital cost to operating cost. Operating cost is always very low compared to the capital cost required to setup the cloud infrastructure. Many small organization or Silicon Valley start-ups are using cloud to launch their applications or infrastructure as it allows them to keep their investments in the check. Cloud consists of many high power physical nodes which constitutes these resource pools. These resources pools have many gigs of RAM, large storage and multi core cpupower [2]. Users resources request are served from these resource pools and resources are allocated to the end users.

Cloud computing is abstraction of physical resource pool to virtual resources [3]. Cloud consists of various physical servers which are located in various data center across the globe. These physical hosts created huge amount of resources pool of memory, computing power and disk space. To virtualize physical resources cloud uses virtualization technology to virtualize the computing resources. Hypervisors act as middleware in virtualization and sits on physical hosts. Currently there are some popular hypervisors available in the market which includes VMware, Xen, Hyper V and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

KVM. Cloud computing also changed the way how services are deployed as the infrastructure management is solely pass to the cloud provides now and user just need to figure the required resources for his applications.

II. Related Works

Pricing model used by the cloud providers is pay per use so this ease of initial investment burden from the cloud users and covert them to just operating cost. This also transfers lot of risk to cloud provider instead of the risk on end users. Cloud can be categories in three types which include the public cloud, private cloud and hybrid cloud. Public clouds are generally maintained by the big players like Amazon, Google and Microsoft Cloud which have their own huge infrastructure to host million numbers of servers. Public clouds are very affordable way to get started with the cloud as users need to only spend the operating cost on them instead of long term investments. Many start-ups companies are using public cloud as their launching pad for applications. Private cloud on other hand is built privately by organization itself. Private cloud provides a sense of security and data privacy as data is kept in house instead of public servers. Private cloud also requires huge amount of financial investment as the hardware needs to be purchased by the organization. Private cloud also requires trained management staff to maintain the cloud up and running. Hybrid cloud is combination of the public and private cloud which allows users to use features of both public and private cloud. On the basis of services provided by the cloud can be categories in infrastructure as service, platform as service and software as service. In infrastructure as service, virtual machines are basic unit of resources. Virtual machines are provided to end users in infrastructure as service. Platform as service provides platform to the users to develop their application on the cloud. Such platforms are generally integrated development environments which includes the entire necessary tool for particular application. Software as service is most abstract model of the cloud computing and provides only particular service to end users. Sales force CRM service is example of software as service where only CRM application is provided to the end users.

Resource allocation is main core problem in the cloud computing which need to be addressed [3]. User requests virtual machine resources and cloud provides needs to fulfil those requirements. While allocating those resources cloud providers needs to consider many things like current state of available resources, any limitation of the infrastructure and availability of the resources requested by the end users. The provisioning or placement of virtual machines can be considered in some phases like accepting virtual machine requests from the users, accessing the status of resources on the nodes and selecting appropriate node for virtual machine deployment. Such general systems can have time complexity of m*n where m is number of virtual machine requested by the users and n is the number of nodes available with the cloud provider.

Virtual machine allocation is done using various state of the art virtual machine placement algorithms. The problem statement for these virtual machine algorithms can be model as below:

Let NodesList = {N1, N2, N3,Nn} is the list of nodes available with cloud provider.

Let VMList = {VM1, VM2, VM3, VMn} is the set of requested virtual machine from client or end user of the cloud.

Cloud providers then need to find mapping between the virtual machines to nodes and also needs to consider certain goals set by the cloud providers. The goal can be different for each of the cloud provider some focus strictly on the SLA while some focus on energy consumption of the nodes

Various virtual machine placement algorithms are used to solve the virtual machine to node mapping problem. Each algorithm can have entire different goal while doing this virtual machine to node mapping. We can roughly divide the available algorithm strategies in two types primarily known as power consumption based approach and QoS based approach. In power consumption based approach main focus is to reduce the power consumption by satisfying requests in less number of possible nodes. Reduction in number of nodes allows savings in power, hardware cost and overall infrastructure cost. Quality of service based approach is based on delivering quality of service and maintaining service level agreements with the end users. The prime focus of such virtual machine placement algorithm is user specific requirements.

Currently there are many virtual machine placement algorithms which are available. Most commonly used algorithms include First Fit algorithm, Ranking Algorithm, Round Robin algorithm and stochastic algorithm. These algorithms only considers the users requirements and do not allow users to pass any extra information like virtual

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

machine layout and specific required features. Our proposed system considers both the user requirements and physical features of hosting nodes to meet user requirements. This system understands the users requirements and cloud providers requirements mutually hence we called it as mutual scheduler.

III. Mutual Cloud Scheduler

A. *Introduction:*

Cloud users or clients can enter their workload information in the provided interface with help of any modern browser. Workload information generally includes the number of virtual servers required, their association with each other, resources required for each virtual machine and the operating system or application which will be running on these virtual machines. This workload information is then forwarded to the Mutual cloud scheduler to create the possible mapping of these virtual machines to the compute nodes. Mutual cloud scheduler can work with both private cloud and public cloud. In the public cloud the information like node information and its features can be fetched using their API. The public cloud provider can return below information: Node Information: The retrieved information from the API will includes node name, node ID, sizes of the VM it can hold and features which are available on that node. Virtual Machine Information: API also provides the information retrieved about the virtual machines, this information includes hostname of the virtual machine, IP address of the virtual machine and size of the virtual machine. Administrative constraints: Public cloud providers can also return the administrative constraints or features of the particular nodes which can help algorithm to select the proper nodes.

Let H = H1, H2, H3, ..Hn be the set of all the compute nodes available in our infrastructure or present with the cloud providers. Now suppose any client or end user requested some set of virtual machines V = V1, V2, V3, ..Vn. There can be various kind of constraints either placed by cloud provider or placed by user while requesting the virtual machine can be considered as set of constraints say C = C1, C2, C3, Cn Here system will need to map the virtual machine to hosts such that all the constraints are satisfied as well as resource consumption is also optimal.

When there is any workload information entered by user then that information is processed by the mutual cloud scheduler. Mutual cloud scheduler checks the available nodes for each virtual machine which is requested by the client via the workload information. Then each virtual machine is mapped to the compute nodes which are called as deployment profiles. Deployment profile M is mapping between virtual machine to compute node.
Deployment Profile m is $V \rightarrow H$

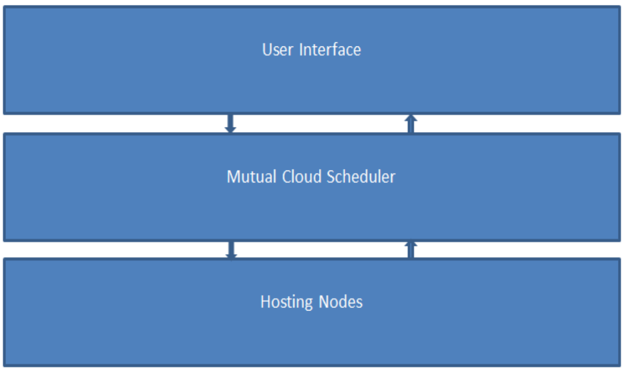

Figure 1Block Diagram of System

B. *Constraints:*

Mutual cloud scheduler then find the satisfaction score of each compute nodes by considering the constraints entered by the users and constraints placed by the cloud admin. These constrains are stored in array say C which holds all the constraints. System calculates the score of each deployment profile and determines the profile with the highest score. Score is calculated on the basis of constraints placed by both users and cloud administrator. The score function can be realized by using:

$$\text{Score}(m) = \sum W * C$$

Where C denotes the set of constraints
W denotes the predefined weight

These scores are calculated for all the deployment profiles. The profile with the highest score is selected as the optimal profile for the VM deployment. Then this information is passed to the cloud provider for actual VM deployment using the cloud API. This process can be repeated for all the virtual machines requested by the user. While deploying the second virtual machine system will also consider the constraints placed for these two virtual machines.

C. *Simulated Annealing:*

Algorithm 1: VM array and Host array Formation
Input: Workload information from client and node information from cloud API
Output: VM array with related constraints and Compute node array
1. N = number of virtual machines give in the workload
2. VM(i) represent the virtual machine where i ranges from 1 to N
3. Store the information entered by the client in form of the array VMarray[N]
4. Get the constraints entered by the user and save them in respective virtual machine entries in the VMarray[N]
5. Fetch the available nodes from the cloud provider and store the available numbers of nodes in Cn.
6. Fetch each compute nodes information and store them in Nodearray[Cn]
7. Return VMarray[] and Nodearray[]

Algorithm 2: Constraint satisfaction score:
Input: VMarray[] and Nodearray[]
Output: Constraints satisfaction Score for each VM to node mapping
1. Get VMid from VMarray[]
2. Get Nodeid from Nodearray[]
3. For each VMid to Nodeidmapping do;
4. Check constraints entered for VMid in the VMarray and match them with features of Nodeid
5. For each constrains from the from VMarray[] do
6. If VMid.constraint(i) == Nodeid.feature(i)
7. Then satisfaction + = 1
8. End if
9. VMid.constraint(i+1) == Nodeid.feature(i+1)
10. Then satisfaction +=1
11. End if
12. Return Satisfaction for each VMid to Nodeid mapping

Algorithm 3: Find optimal deployment profiles on the basis of satisfaction scores
Input: Satisfaction for each VMid to Nodeid mapping
Output: Vmid to Nodeid mapping which have highest score.
1. Compare the score of each Vmid to nodeid mapping
2. maximum = array[0]
3. for (c = 1; c is less than equat to size; c++)
4. if (array[c] is greater than maximum)
5. maximum = array[c];
6. Return the nodeid for particular VMid with max score.

Finding the optimal profile is NP hard as we cannot determine the ultimate virtual machine to host deployment scheme and we will need to use the approximation scheme to achieve this. We can solve this problem with the simulated annealing, as we are using the score function here to determine best possible deployment profile for the virtual machine to host mapping. We can use simulated annealing to determine the deployment profile with the maximum score.
The optimal profile (mopt) is the one with the highest score:
Score(mopt) > Score(mq)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Algorithm starts generating random profile at the start and start visiting the next profiles randomly to find the high scoring deployment profiles.
Here, V is the set of all VMs, Mall is the set of all profiles, d is the probability for a VM v to be deployed on a hosting node other than the one set by profile m. Increasing d results in wider neighbourhoods and prevents us from getting trapped at local optima. Yet, too wide neighbourhoods result in almost randomly generated neighbours and, thus, deployment profiles of low quality

```
Input: same_iterations: After how many iterations
showing no improvement will we stop our search
T: Temperature
Score(): Deployment profile score function
Output: A near-optimal deployment profile
same = 0
best profile = current profile = GetRandomProfile()
while same <same_iterations do
new profile ¼ GetNeightborOf(current profile)
D = Score(new profile) - Score(current profile)
if ( T> 10-5 AND eD=T > Random()) OR (T < 10-5 AND D > 0) then
current profile = new profile
end if
if Score(new profile) > Score(best profile) then
best profile = new profile
same = 0
end if
same + +
T = 0:99*T
end while
return best profile
```

Our proposed system will help users to deploy their complex workflows in the cloud in more efficient and more effective way also this will allow cloud providers to prevent over expose of the cloud internals. This simply place the requested virtual machines on the nodes according to the users requirements or constraints placed in the workload but also considers the specification of compute nodes so that resource consumption can be optimal. These task flows can be realized with the help of constraints or interrelation placed by the end user. Our system allows following constraints which can be entered by the users while entering the workload information:
AssociateVM: Allocate the same node for the VMs which have AssociateVM constraints placed by the users.
DistantVM: Allocate the different nodes for the VM's which have DistantVM constraints placed by the users.
PrivatenetVM: Enable the private networking in the VM's which have PrivatenetVM constraints placed by the users.
BackupVM: Allocate the virtual machine on the compute node which have backup enabled for the data redundancy.

These users placed constraints can help to realize any virtual machine work flow entered by cloud users. This will help users to communicate the infrastructure needs more effectively even in a public cloud provider. Suppose user want to create high availability apache cluster then user will need to make sure that these virtual machines are deployed on different nodes so that high availability can be achieved. In proposed system user can enter the virtual machines with the DistantVM constraints so that requested virtual machines can be deployed on the separate nodes.

Cloud administrators can control the cloud scheduler by placing some cloud administrative constraints which can priorities some nodes or can rule out some nodes from the process if there is any maintenance activity is going on. This gives cloud admin control over proposed mutual cloud scheduler and allows control in this automatic process. Cloud administrators can place below constraints on the nodes:
EvenNode: Distribute VMs on the compute nodes.
PowersaveNode: Cloud admin can place this constraint on compute node on which no new virtual machines will be setup.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

MaintainanceNode: Cloud admin can place this constraint on the node if there is any maintenance going on the node so that VM will not be allocated on that node.

IV. Evaluation

We can implement this system with both the public and private cloud. Most of the public cloud providers have their own restful API system which allows integration of our code onto their system while in the private cloud we can calls direct hypervisor calls to allocate VM on a particular node.

We have implemented this system on a public cloud provider. We have used the API provided by them and integrated into our control panel. Our control panel allows users to enter the workload information in terms of the number of virtual machines and their inter relationships. This allows users to communicate their workload information more effectively even in a public cloud. The information which can be retrieved from public cloud API includes:

Create a new virtual machine
Fetch the available nodes Information
Fetch the features of the nodes
Fetch the constraints placed by public cloud provide on the compute nodes
Destroy the virtual machine

Our proposed system will allow users to communicate their workload information with a simple control panel. Users can deploy set of virtual machines or workflow in the public cloud via single interface. Consider that user wants to build a load balancer + cache + replication infrastructure in the public cloud which involves set of virtual machines which are associated with each other.
Then this typical system will look like something as Figure 2

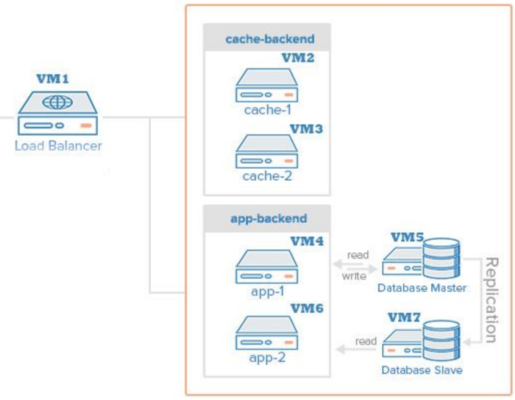

Figure 2 Sample User Workload

To deploy these user requirements in our proposed system user can model this information in terms of number of virtual machines, and there interrelation. To realize this system user can request 7 virtual machines at a time with below constraints in-between:

VM2 and VM3 :DistantVM
VM4 and VM5 :PrivatenetVM
VM6 and VM7: PrivatenetVM

User can enter the above information from the control panel and proposed system will allocate these virtual machines on the compute nodes in such way that these constraints are satisfied as we all resource consumption on the nodes stays optimal too. They can also do this manually but that will involve a lot of work as users will need to check if these features are available or not on a particular node then will manually select the node and will need to manually deploy the virtual machines. Our system saves users from all this hassle and quickly deployed such workloads in the cloud and also preserves the virtual machines interrelations.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

V. EVALUATION

To evaluate the proposed system we have compared the VM allocation patterns of our proposed system to some existing virtual machine placement algorithms. In the given sample user requirement user have requested below 7 virtual machines refer figure 2.

We have modeled various placement strategies to test the performance of our proposed system. We have created hosts and virtual machine requests and calculated the results.

Table 1: Size of compute nodes

Compute Node	Memory
C1	16 GB
C2	32 GB
C3	32 GB
C4	16 GB
C5	12 GB

In table 1, we are considering 5 nodes here having the mentioned RAM with their host machines. Each node can have different features and specifications but we are only considering there RAM as the constraints here.

Virtual Machines Requested:

Table 2: Size of virtual machines

Virtual Machine	Memory
VM1	8 GB
VM2	2 GB
VM3	2 GB
VM4	4 GB
VM5	4 GB
VM6	6 GB
VM7	6 GB

Table 2 represents the virtual machines requested from the client. Single user can requests multiple virtual machines according to his need and workload architecture.

VM to host Mapping:

Table 3: VM to compute node mapping

Virtual Machines	First Fit algorithm	Round Robin	Ranking	Mutual Scheduler
VM1	C1	C1	C2	C3
VM2	C1	C2	C3	C2
VM3	C1	C3	C3	C3
VM4	C1	C4	C3	C4
VM5	C2	C5	C2	C4
VM6	C2	C1	C3	C2
VM7	C2	C2	C2	C2

Table 3 shows mapping of virtual machines to various hosts according to various algorithms mentioned in the table.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

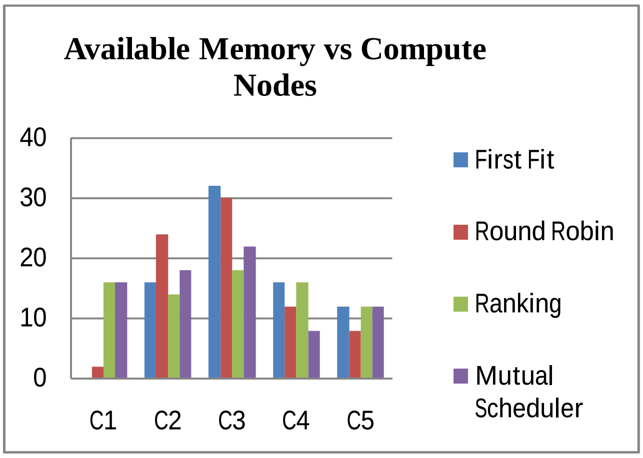

Figure 3 Experimental Result Free Memory vs Host

Figure 3 shows the free memory available on compute nodes after allocation of virtual machines using different allocation strategies. The experiment shows that our algorithm considers user entered constraints i.e. place virtual machine 2 and virtual machine 3 on distinct nodes, virtual machine 4 and virtual machine 5 on the same node for private networking and same for virtual machine 6 and virtual machine 7. Mutual schedulers also utilized all the nodes efficiently while first fit strategy and ranking algorithm utilized certain nodes to full extend while others are kept completely empty. Other algorithms also affect the application performance as they do not consider any virtual machine placement requirements passed by users.

VI. CONCLUSION AND FUTURE WORK

Public cloud is getting popular these days due to its costing models and scalability. Virtual machines are quickly deployed in the cloud as per user's requirements but sometimes users are not able to communicate their complex needs to public cloud providers due to many abstractions layers. In this paper we have proposed a system called mutual cloud scheduler which allows end users to pass various VM relations so that their requested virtual machines can be placed on the nodes according to their needs. Users can influence the placement strategy used by the cloud provider to deploy custom layout of the virtual machines. Mutual cloud scheduler also considers node side specifications while placing the virtual machine which results in efficient consumption of the nodes resources. We have compared the proposed strategy with existing strategies like first fit, round robin and ranking algorithm and found that proposed scheme efficiently consumed resources and also satisfy users custom virtual machine layout requirements too. In future we can implement this system with multiple clouds API to realize the power of federated clouds.

REFERENCES

[1] Konstantinos Tsakalozos, Mema Roussopoulos, and Alex Delis "Hint-Based Execution of Workloads in Clouds with Nefeli" IEEE transactions on parallel and ditributed systems, vol. 24, no. 7, July 2013

[2] M. Rosenblum and T. Garfinkel, “Virtual Machine Monitors: Current Technology and Future Trends,” Computer, vol. 38, no. 5, pp. 39-47, May 2005.

[3] H.N. Van, F.D. Tran, and J.-M. Menaud, “Autonomic VirtualResource Management for Service Hosting Platforms,” Proc. ICSEWorkshop Software Eng. Challenges of Cloud Computing, pp. 1-8,2009.

[4] Bobroff, N. ; T.J. Watson Res. Center, IBM, Hawthorne, NY ; Kochut, A. ; Beaty, K. "Dynamic Placement of Virtual Machines for Managing SLA Violations" Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE

[5] Chaisiri, S. ; Sch. of Comput. Eng., Nanyang Technol. Univ., Singapore, Singapore ; Bu-Sung Lee ; Niyato, D. "Optimal virtual machine placement across multiple cloud providers" Services Computing Conference, 2009

[6] Remesh Babu, K.R. ; Dept. of Inf. Technol., Gov. Eng. Coll., Idukki, India ; Samuel, P. "Virtual Machine Placement for Improved Quality in IaaS Cloud" Advances in Computing and Communications (ICACC), 2014

[7] Jamali, S. ; Dept. of Electr. & Comput. Eng., Univ. of Mohaghegh Ardebili, Ardebil, Iran ; Malektaji, S. "Improving grouping genetic algorithm for virtual machine placement in cloud data centers" Computer and Knowledge Engineering (ICCKE), 2014

[8] Ricardo Stegh Camati, Alcides Calsavara, Luiz Lima Jr. "Solving the Virtual Machine Placement Problem as a Multiple Multidimensional Knapsack Problem" ICN 2014 : The Thirteenth International Conference on Networks

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

[9] Nskinc study on cloud computing, Nskinc white paper2012
[10] Linlin Wu ; Dept. of Comput. & Inf. Syst., Univ. of Melbourne, Melbourne, VIC, Australia ; Garg, S.K. ; Versteeg, S. ; Buyya, R. "SLA-Based Resource Provisioning for Hosted Software-as-a-Service Applications in Cloud Computing Environments" Services Computing, IEEE Transactions on (Volume:7 , Issue: 3)
[11] Dong Huang ; Inst. for Infocomm Res., Agency for Sci. Technol. & Res, Singapore, Singapore ; Bingsheng He ; Chunyan Miao "A Survey of Resource Management in Multi-Tier Web Applications
[12] Pankaj R Kadam, Nilesh V Alone "Review on KVM Hypervisor" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-3 Issue-4 September, 2014
[13] Mell, P. and Grance, T. 2011. "The NIST Definition of Cloud Computing" NIST Special Publication, 800-145

BIOGRAPHY.

	**Pankaj Kadam** Pankaj R Kadam received his BE degree from Pune University. He is currently pursuing his master degree in computer science. He also has 3 Years of working experience in the cloud and worked on the various cloud technologies. He is currently working with a private firm and managing the cloud servers for the clients

	**Nilesh Alone** Nilesh V Alone is currently working as Assistant Professor in Department of Computer Engineering, R. H. Sapat College of Engineering, Management Studies & Research, Nashik. He is having 13 years of experience in teaching. His area of interest is in Cloud Computing and Machine Learning.