International Journal of Current Engineering and Technology
©2015 INPRESSCO®, All Rights Reserved

E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

*Research Article*

# Threshold based Dynamic Resource Allocation using Virtual Machine Migration

**Ravikumar U. Ighare†* and R. C. Thool†**

†Dept. of Information Technology, SGGSIE & T, Nanded (MS), India

Accepted 23 July 2015, Available online 27 July 2015, **Vol.5, No.4 (Aug 2015)**

***Abstract***

*Cloud Computing has emerged as a technology that provides flexibility, availability and scalability of resources. All these features are provided by virtualization technology which can be considered as a heart of cloud computing model. Virtualization technology multiplexes virtual resources onto the physical resources. Cloud computing model provides on demand resources to users as per their varying needs. Hence efficient resource allocation strategies are needed for data centers that reduce cost. We developed a system that effectively reduces a system overload and reduces number of physical servers used. We have simulated the algorithm which achieves better performance.*

***Keywords:*** *cloud computing; virtualization; resource allocation; green computing; resource management; virtual machine migration.*

## 1. Introduction

In recent years, Computational power demand of IT infrastructure has grown continuously. Many modern businesses evolved with huge demand of the computational, storage and network resources. It became headaches for companies to provide varying demands and manage the software and the hardware. Hence companies are moving from traditional software model to the cloud based solutions. Cloud computing model provides convenient, on-demand network access to resources. These resources could be servers, storage, networks, applications, and services. All the network, storage, server, computing capabilities are accessed over the internet (M. Armbrust, *et al,* (2009). Cloud computing follows Utility Computing model which charges customers for exact usage of the services (Peter M. Mell, *et al,* 2011).

Cloud datacenter contains thousands of severs consuming large amount of power. In 2006, a research estimated that electrical energy consumed by IT infrastructure in US was near about 61 billion kWh which lead to 4.5 billion dollars as cost for electricity (R. Brown, *et al,* 2008). Under present trend it is highly likely to increase three times by 2016. The reason behind such enormous power usage is not only the power used by servers for computations but also inefficiently used servers in data center. The reason for such enormous power consumption is not only servers used for computation but also inefficiently used servers. A study has shown that in a large scale datacenter many servers has resource utilization in range 10%-50% of their total capacity. Such underused servers increase the expenses of datacenter for electricity (L. A. Barroso, *et al,* 2007).

Cloud data centers use virtualization technology to multiplex their virtual resources onto the physical resources. From many years virtualization has been used in IT industry as a strategy consolidate servers. Virtualization is a technique that shares a single physical instance of software or hardware resource among multiple users. Virtual machine is environment (usually program or OS) which does not exists physically but created within another environment. Many hypervisors like Xen has a mechanism that maps their virtual machines to physical resources in data centers (P. Barham, *et al,* 2003).

In this paper we first discuss the need for efficient dynamic resource allocation strategies. Then we have made a survey on the different existing strategies that make use of virtual machine for resource allocation for cloud computing environment. We discuss advantages, disadvantages and proposed a method for efficient dynamic resource allocation strategy. Rest of paper is organized as follows, Section 2 gives overview about resource allocation and its significance. Related work is discussed in Section 3. Section 4 gives overview about our proposed system. Section 5 describes the system model. Section 6 describes allocation policies. Simulation results are presented in Section 6 and Section 8 concludes the work.

## 2. Resource Allocation & Its Significance

Resource Allocation is the process of assigning available resources efficiently as per the need. Resource Allocation System (RAS) is a mechanism that

*Corresponding author: **Ravikumar U. Ighare** guarantees the requirements of user applications are served correctly (Glauco E. Gonçalves, *et al,* 2011). When user submits a job to the cloud, job is partitioned into many tasks such as,

- Which resources to be allocated to the tasks
- When resources will be allocated
- Which VM will be used as source or target for allocation
- How many resources will be allocated to the task

To resolve these issues there is need of better task scheduling and resource allocation mechanism. Efficient resource provisioning involves two basic steps, (1) static planning: assigning the resources initially to VMs and then classify the VMs and deploy them to physical hosts. (2) Dynamic Resource Allocation: This step involves the allocation of additional resources, VM creation and migration dynamically (Timothy Wood, *et al,* 2007).

Following issues should be considered by an optimal Resource Allocation Strategy

A. Resource Contention : It is the situation where two or more applications try to access the same resource at the same time.

B. Scarcity of resources : This situation arises due to Limited resources.

C. Resource Fragmentation : This situation arises when resources are isolated. In such situation, there are enough resources available but still RAS can not fullfill the application requirements.

D. Over-provisioning : This arises when application gets more resource than it needed.

E. Under-provisioning : This arises when fewer resources are assigned to application than it needed.

It is impractical for the cloud service provider to predict the dynamic nature of user demands and the workload. Cloud resources are limited and heterogeneous, also the resource demand is dynamic. Hence we need an efficient dynamic resource allocation system.

## 3. Related Work

Virtual Machine migration is widely used technique for dynamic resource allocation. Many hypervisor like Xen (C. Clark, *et al*, 2005) and VMware (VMware, 2006) provide support for live migration of VMs.

In paper (Ying Song, *et al*, 2013) proposed a novel two tiered resource allocation strategy that includes local as well as global optimization. The focus is given on global as well as local optimization. Local and global scheduler are used to implement the proposed resource allocation mechanism. The main intension behind both schedulers is to focus on the resource allocation. Local Resource Scheduler controls allocation of resources to VMs within local server. Resource utilization is optimized by adjusting the CPU time slots, memory assigned based on the resource utilization, quality and activity of application. Activity is defined as the threshold of resource allocation. Frequency of execution of local reso`14urce scheduler is short (example, 1 second). When resource utilization reaches the threshold, few additional resource are allocated to the VM. Resource allocation of the application in the entire system is controlled by Global Resource Scheduler. It adjusts the activity of applications in each local scheduler. It optimizes the resource allocation at longer intervals (example, 30 second). The efficiency of this resource allocation strategy depends on how well algorithm can predict the resource utilization and number of requests arrived. The proposed system do not reduc3e the number of servers used and hence there is no support for green computing.

Sandpiper system uses a Xen VMM which migrates VMs based on single threshold (Timothy Wood, *et al*, 2007).. Nucleus runs in domain0 of Xen and collects the usage statistics of that physical machine. Statistics of processor monitored by Monitoring Engine, memory, and network. Sandpiper control panel consists of Profiling Engine, Migration Manager and Hotspot detector. Profiling Engine makes profile for each physical server based on the usage statistics. Hotspot detector detects hotspot by monitoring these profiles. Migration manager decides the migration of VMs. The work does not have support to green computing.

Authors in (A. Beloglazov, *et al*, 2010) proposed heuristics for dynamic provisioning of VMs to the servers in Data Centers. Our work also based on these heuristics. Optimization of virtual machine allocation is done in two steps. In first step, VM to be migrated is selected and in second part the destination server is selected based on Modified Best Fit Decreasing (MBFD) algorithm. Heuristics are based on the upper and lower utilization threshold. VM migration is done based on threshold violation.

*Measure-Forecast-Remap (MFR)* technique is proposed (Bobroff, *et al,* 2007) for dynamic resource allocation for virtualized environment. Algorithm measures the historical data and based on historical data algorithm forecasts future demand and remap virtual machines to physical machines.

In paper (Pawar, C.S., *et al,* 2013), dynamic resource allocation strategy is proposed based on the priority. Tasks have been assigned the priority. When a task arrives with higher priority, execution of task with lower priority is preempted. If preemption is not possible then new instance of virtual machine is created. Task will be placed in the waiting queue if resources are not available. When the VM will be free then task from waiting queue is selected and executed. Priority assignment is static and preemption is done based on priorities which is not efficient when tasks are critical.

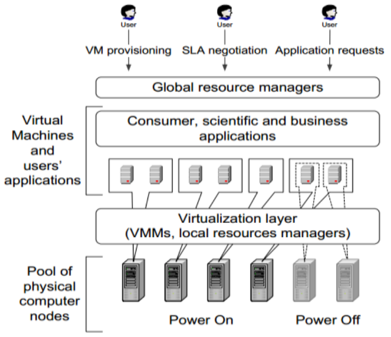

User
User
User
VM provisioning
SLA negotiation
Application requests
Global resource managers
Virtual Machines and users' applications
Consumer, scientific and business applications
Virtualization layer (VMMs, local resources managers)
Pool of physical computer nodes
Power On
Power Off

**Fig.1** System View

## 4. Proposed Method

We propose an automated resource allocation system that effectively does the remapping of virtual machines to the physical machine. Our algorithm uses the utilization threshold to make decision of the virtual machine migration. We have used heuristics proposed in paper (A. Beloglazov, *et al,* 2010) Following goals are effectively achieved by our algorithm:

- Overload Avoidance: PM should be capable of providing resources all VMs running on it. Overloaded PM leads to performance degradation. VMs are migrated from such overloaded PMs to other PMs to avoid performance degradation.
- Green Computing: Number of physical machines used should be minimized, still they satisfying all the needs of VMs. Idle and underutilized PMs are turned off (Fig. 1.).

There is trade-off between above mentioned goals when the resource needs of VMs is always changing. Utilization of PMs should be kept low so that they can perform better, For Green Computing, server utilization should be kept high to effectively use PM.

## 5. System Model

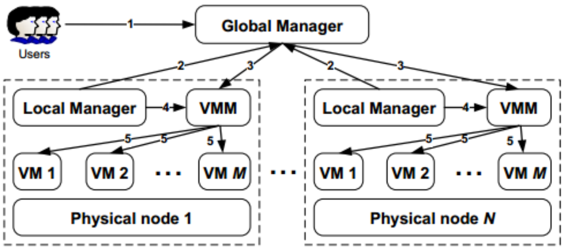


**Fig.2** System Model

The IaaS environment is represented as a target system in Figure 2. The datacenter consists of *N* number of hetergoeneous nodes. Each node *i* has node confihuration such as CPU capacity defined as Millions Instructions Per Second (MIPS), network bandwidth, Disk storage and amount of RAM. The environment proposed has dynamic workload. Virtual machine has configuration such as amount of RAM, network bandwidth,disk storage and CPU capacity.

The software layer of system consists of two tiers, Lokam Manager and Global Manager.Local Manager resides in each physical node as a part of VM Monitor (VMM). Local manager does tasks of monitoring of CPU, RAM utilization of server, Allocating resources to VMs as per varying resource demands and deciding the VM to be migraded from host in case of overload. Global Manager residing in Master node is responsible for maintaining overview of resource allocation of all hosts. It collects the statistical information from Local Manager. Global Manager is responsible for issuing commands for optimization of VM allocation to hosts. VMM does the task of resizing of virtual machines, virtual machine migration from one host to another host. Our algorithm turns of the idle nodes as well as underutilized nodes.

## 6. Allocation Policies

Allocation policies are divided into three parts: (1) selection of the virtual machines that have to be migrated for distribution of workload; (2) Distribution of virtual machines to the destination host; and (3) Consolidation of VMs and turning off idle servers. We discuss these parts in following sections.

### *6.1 Virtual Machine Selection*

We are migratng VMs based on its utilization threshold. We have set two thresholds, *upper utilization threshiold* and *lower utilization threshold.*. Upper utilization threshold is the maximum resource utilization threshold for a server. Hosts violating upper utilization threshold are identified as *Hotspot*. Virtual machines from such hotspots need to migrated to another host to keep some resources free to avoid SLA violation. *Lower utilization* threshold is the threshold for the minimum resource utilization for a server. Hosts violating lower utilization threshold are identified as *Coldspots*. Virtual machines from such coldspots need to migrated to another host so that number of hosts used can be minimized by turning off such idle and underutilized hosts. We keep the resource utilization of hosts between these two thresholds.

Following is pseudo code for identification of overused hosts and underused hosts.

**Algorithm 1** Overused Hosts
**Input:** HostList, uThreshold
**Output:** overusedHosts list

```
define a new list overusedHosts
get HostList, UThreshold
for each host from HostList
get cpuUtilizationRate and get ramUtilizationRate
  If cpuUtilizationRate > UThreshold || ramUtilizationRate > UThreshold
```

```
        Then add host to overusedHosts
6.    End for
7.    return overusedHosts list
```

**Algorithm 2** Underused Hosts

```
Input: HostList, LThreshold
Output: underusedHosts list
1.    define a new list underusedHosts
2.    get HostList, LThreshold
3.    for each host from HostList
4.      get cpuUtilizationRate and get ramUtilizationRate
5.        If cpuUtilizationRate < LThreshold || ramUtilizationRate < LThreshold
          Then add host to underusedHosts
6.    End for
7.    return underusedHosts list
```

Virtual machines from overused hosts and underused hosts are selwected for migration. Our algorithm periodically run to verify if there is any hotspot or coldspot present in the system, If there are any, then distribute VM over other hosts. Distribution of virtual machines is discuissed in nect section.

### *6.2 Virtual Machine Distribution*

The main task of this algorithm is migrating virtual machine from one host to another. We have proposed the algorithm for detection of Hotspots and ColdSpots in previous section. Migration is triggered when algorithm detects any hotspot or coldspot. Migration list is calculated based on the overused and underused hostsOverused hosts are sorted by power consumption. One overloaded host is selected from the list, and some of its load is distributed over other hosts. Target host is choosen such that there will be no threshold violation after migration of VM to that host. If source host is still overloaded then algorithm run again untill source host is not overused anymore. The pseudo code for migration is as follows:

**Algorithm 3** Distribution Algorithm

```
Input: overusedHosts list, NotOverusedHosts list, threshold
Output: Migration of VMs.
1.    get list of NotOverusedHosts and assign it to targetHosts list
2.    sort overused hosts by PowerConsumption
3.    for each overusedHost
4.      for each targetHost
5.            create resourcesMap with key as target host id and values as resourcesList with amount of used CPU and RAM
6.            add HostCpuUtilization and HostRamUtilization to the resourcesList
7.      End for
8.      get cpuUtilization and ramUtilization for sourceHost
9.      for each targetHost
10.       if there are enough resources available to migrate VM to host
11.           then update resource utilization of targetHosat
12.           subtract the utilization of the source host
13.           add the entry on the migration map
14.           if ramUtilization && cpuUtilization of source host < uThreshold
15.           then go to line 4
16.           else go to line 9
17.       End if
18.     End for
19.   End for
```

If we could not find appropriate destination server for a VM then we move on to the next VM on source host and try to find destination server for this new VM. As long as we are able to find destination server for VMs from overused hosts, we consider our algorithm works successfully. Consolidation of VMs is discussed in next section

### *6.3 Virtual Machine Consolidation*

We consolidate hosts so that number of hosts used are minimum. Consolidation algorithm migrates virtual machines from coldspots and turns off the server. Turning off idle servers or underutilized servers saves energy. Its challenging to reduce number of servers without degrading performance. We need to migrate all the virtual machines of the host in order to turn it off.

For consolidation algorithm source hosts are underused hosts and destination hosts are active hosts. Hosts having resource utilization above zero is identified as active host. We identify all the coldspots in the datacenter. For each virtual machine on coldspot, we find the destination host. We check if the destination server has enough resources to accommodate the virtual machine and its resource utilization should not cross the upper utilization threshold. Sometimes a coldspot can be choosen as destination server so that its resource utilization goes in acceptable range. If the destination server is likely to become hotspot after accepting the virtual machine then we find another destination srver. Aftern migrating all the virtual machines from coldspot, it is turned off.

## 7. Experimental Result

### *7.1 Experimental Setup*

We evaluated our threshold based virtual machine migration algorithm using CloudSim and CloudReports. CloudReports is a tool that simulates distributed computing environment based on cloud. It uses CloudSim as simulation engine and provides easy-to-use graphical user interface. Further information about CloudSim can be find in paper (R. N. Calheiros. *et al,* 2010).

To check effectiveness of our algorithm, we run our algorithm in CloudReports. Host and Virtual machine configuration is as shown in table 1 and table 2 respectively.

**Table 1** Experimental Host Configuration

| Sr. No. | Parameter | Value |
|---|---|---|
| 1. | Number of processing elements | 4 |
| 2. | RAM (MB) | 40,000 |
| 3. | Storage (GB) | 10 |
| 4. | Power (kW) | 250 |
| 5. | MIPS/PE | 2400 |
| 6. | Bandwidth (mbps) | 1000000 |

**Table 2** Experimental VM Configuration

| Sr. No. | Parameter | Value |
|---|---|---|
| 1. | Number of processing elements | 4 |
| 2. | RAM (MB) | 512 |
| 3. | Image Size (MB) | 1000 |
| 4. | MIPS | 1000 |

### *7.2 Simulation Result*

We have simulated our algorithm with configuration mentioned in table 1 and table 2. First we simulated simple VM allocation policy without VM migration. Simulation parameters are shown in table 3. It is found that many hosts are utilized up to 100 % without VM migration. Utilizing host to 100% results in the degraded performance also the temperature for the server increases as its power consumption increases. Then we simulated using Double threshold Policy with migration enabled. We can see many migrations because of threshold violation. VMs are migrated from one host to another host because of either they are overused or they are underused. Simulation parameters are presented in table 3.

**Table 3** Simulation Parameters

| Sr. No. | Parameter | Simple Policy | Dynamic Policy |
|---|---|---|---|
| 1. | Number of hosts | 4 | 4 |
| 2. | Number of VMs | 30 | 30 |
| 3. | Upper Threshold | - | 0.8 |
| 4. | Lower Threshold | - | 0.2 |
| 5. | Cloudlet sent per minute | 100 | 100 |
| 6. | Length of cloudlet | 50000 | 50000 |

Using above parameters we have simulated algorithm and results for resource utilization are shown in Figure 2. When VM migration is disabled, out of four hosts, two hosts are fully utilized its resources which may result in degraded performance. Host 3 and host 4 are averagely utilized hence not shown here.

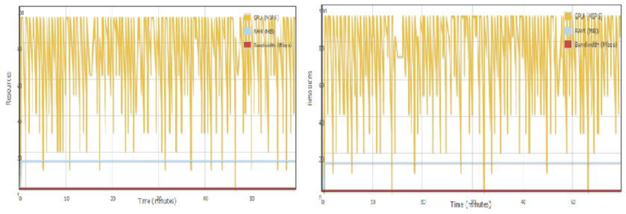
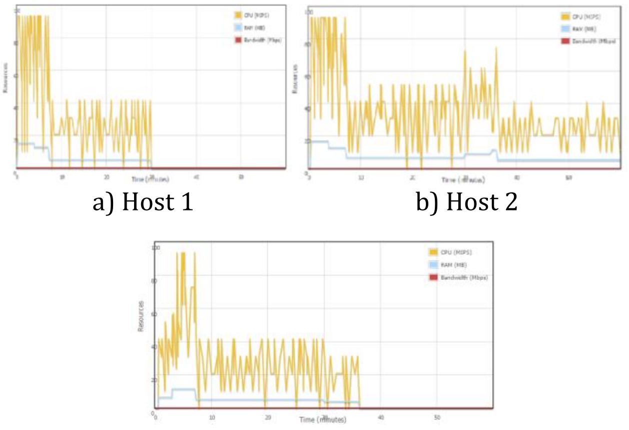


**Fig. 3** Resource utilization for host 1 and host 2

In our next simulation we enabled VM migration and used our double threshold VM allocation policy. Result are shown in Figure 3.

Initially host 1 and host 2 were utilized above its upper utilization threshold for some time. As soon as our algorithm detects these hotspots, some of VMs are migrated to host 3.



a) Host 1

b) Host 2

c) Host 3

**Fig. 4** Resource utilization for Host 1, 2, 3

After distribution of hotspots our consolidation algorithm consolidated host 1 and host 3. All of the virtual machines of host 1 and host 3 are migrated to host 2. After migration, these hosts are turned off to save power and host 2 only keep VMs executing.

## Conclusion

In this paper we have introduced a threshold based dynamic resource allocation algorithm that effectively avoids overload in the system and consolidates the hosts to save power. The algorithm allows hosts to distribute the workload to other hosts if the resource utilization thresholds are violated. Our proposed algorithm achieves both overload avoidance and green computing.

## References

M. Armbrust *et al*. (2009), Above the clouds: A berkeley view of cloud computing, University of California, Berkeley, Tech. Rep

Peter M. Mell, Timothy Grance (2011), The NIST Definition of Cloud Computing, NIST Special Publication 800-145.

P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield (2003), Xen and the art of virtualization," in Proc. of the ACM Symposium on Operating Systems Principles (SOSP'03).

Glauco E. Gonçalves, Patrícia T. Endo , Thiago D. Lamb , André de Almeida Vitor Palhares , Djamel Sadok , Judith Kelner ,Bob Melander , Jan -Erik Mångs (2011), Resource Allocation in Clouds: Concepts, Tools and Research Challenges, Brazilian Symposium on Computer Networks and Distributed Systems.

Ying Song, Yuzhong Sun, Weisong Shi (2013), A Two-Tiered On-Demand Resource Allocation Mechanism for VM-Based Data Centers, Services Computing, IEEE Transactions on , vol.6, no.1, pp.116,129

C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield (2005), Live migration of virtual machines, in Proc. of the Symposium on Networked Systems Design and Implementation (NSDI'05).

VMware (2006) Resource Management with VMware DRS, technical paper.

Pawar, C.S., Wagh, R.B. (2013), Priority based dynamic resource allocation in Cloud computing with modified waiting queue, International Conference on Intelligent Systems and Signal Processing (ISSP), vol., no., pp.311,316.

Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif (2007), Black-box and Gray-box Strategies for Virtual Machine Migration, Proceedings of the Fourth Symposium on Networked Systems Design and Implementation (NSDI), Cambridge, MA.

Beloglazov, A. Buyya, R. (2010), Energy Efficient Allocation of Virtual Machines in Cloud Data Centers, Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on , vol., no., pp.577,578

Bobroff, N., Kochut, A., Beaty, K. (2007), Dynamic Placement of Virtual Machines for Managing SLA Violations, Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on , vol., no., pp.119,128.

R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya (2010), CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, In Software: Practice and Experience. Wiley Press, New York, USA.

R. Brown *et al* (2008), Report to congress on server and data center energy eciency, Public law 109-431, Lawrence Berkeley National Laboratory.

Barroso, L.A.; Holzle, U. (2007), The Case for Energy-Proportional Computing, Computer , vol.40, no.12, pp.33,37.