IJCTA, 10(03), 2017, pp. 63-72
© International Science Press

Dynamic Scheduling and Resource Allocation in Cloud

Anusha Bamini* and Sharmini Enoch**

Abstract : Recent years the necessary of resources is increased based on demand of the user. Day by day the internet world is converted into a cloud world by the rapid change in the computing techniques. Allocating resources is considered to be a tedious task when several copies of the similar tasks are assigned to different nodes. This paper proposes the efficient allocation of resources based on skewness measure, which measures the irregularity in the practice of resources. To reduce the skewness the overall usage of server resources is enhanced. The allocated resource is exactly managed and overload resources have to be used appropriately in the place where it is actually needed. This process of minimizing the skewness save the energy and prevent the overload in the system. To enhance our resource allocation system we proposed a single agent skewness algorithm, which is used to split and allocate the tasks from master node to child node. In future, we develop a multi-agent based VM migration system, it may avoid overloading of jobs in cloud environment efficiently and save the energy of virtualized cloud environment.

Keywords : virtualization, scheduling, skewness, migration

1. INTRODUCTION

Virtualization is a novel technology in cloud computing; it is used to utilize the cloud resources effectively. Cloud datacenter is having lot physical machines with pool of physical resources. These physical resources are converted into a virtual resource in cloud [17]. From the cloud provider point of view each VM and virtual resources are considered as number of files. The user belief a dedicated physical machine is allocated as a virtual machine. When a user submitting a request new virtual machine is allocated quickly with dedicated resources. The virtual machine is able to migrate from one VM to other when the load exceeds. It will not create any interruption at the time of migration. Dynamic load balancing is very important in the data center at the time of failure occur in the VM. The dynamic load balancing will carry out the migration process of failed machine. Administrating the assignment and relocation of virtual machines in a data centre is a major confront; previous tools mainly concentrating the central management of service, which merge the presentation of every machine [1]. If the present allocation is not satisfied then the cloud agent will try to allocate a new virtual machine by performing necessary migration process.

Virtual machine migration is the practice of moving virtual machines and its applications into another physical location. At the time of transferring virtual machine into physical location, the resources like memory, network and storage [15] of the virtual machine are moved from the exclusive host machine to destination machine. By focusing on the "VM migration", at data centres, where there are lots of node and VMs that are running need to be managed from failure as soon as possible. The important concept of network virtualization is the migration. It transfers virtual resources to physical resources over the network. The agreement process is involved at the time of migration. We can use the physical resources beyond the limit of resource availability by using memory over commitment technique. If there is any

* Assistant Professor Department of Computer Science and Engineering Noorul Islam University, India. *E-mail: anushabamini@gmail.com*

** Professor Department of Electronics Communication Engineering Noorul Islam University, India. *E-mail: sharminienoch@gmail.com*

physical resource failure occurs then the virtual machine running on the particular physical machine must be migrated. In a data centre thousands of physical machine and virtual machines are running.On the other hand if there is any failure occurs in the virtual node, we need to migrate the virtual machine into another datacenter. This migration process may take little time. While applying migration process several data packets are transferred from source to destination. It may result traffic and power consumption.

Resource allocation [1] constrains having set of restrictions on scheduling resources, and allocating resources to particular task. The machines are situated in dissimilar regions and each machine are having different processing abilities, and different characteristics such as processors, CPU core, size of main memory, cost, etc. The parameters of cost, time and processing capacity are considered at the time of scheduling and allocation. In order to achieve a reduced time and cost with optimized result the task scheduling and resource allocation must be matched carefully. From the optimized result only we can set the effective cloud environment. Computing time is calculated in task scheduling with set of times such as receiving, processing and waiting time. Main objective of this work is to attain optimal task schedules using skewness algorithm to minimize total time.

The paper structured in the subsequent sections. Section 2 organizes the related works which are mostly supporting to this work. Section 3 represents the PM to VM migration, which depicts the migration process from physical to virtual machine. Section 4 analyze about the scheduling process. Section 5 describes the resource allocation process with load prediction algorithm and skewness algorithm. Section 6 demonstrates the resource management module. Section 7 addresses the results obtained from the simulation. And the conclusion and future work mentioned in section 8.

2. RELATED WORK

Scheduling of classical jobs is the difficult problem in scheduling theory. The Ant Colony System algorithm [16] is a scattered algorithm which is widely worn to resolve NP-hard combinatorial optimization problems. Foraging and recruiting activities defines how ants determine the world in hunt of food sources, then discover their way reverse to the nest and indicate the food source to the other ants of the colony. Performance of the ACS for the JSP mostly depends on variables value and the number of ants. Adjusting these parameter values takes a big change of time, for the optimal parameter values depend on the problem to solve, and it is complicated to discover an all-purpose locating of parameters for all problems.

Particle Swarm Optimization [2], [3], [5] is a self-adaptive global search optimization technique. This PSO algorithm is parallel to further population-based algorithms like Genetic algorithms but, there is no direct combination of individuals of population. PSO was applied for many real time problems in all situations. The algorithm of PSO [3], [5] emulates from behavior of animals societies that don't have any head in their cluster or swarm. PSO has turn into accepted due to its unfussiness and its usefulness in broad choice of application with low computational cost [11], [12].

For optimizing result genetic algorithms and bacterial foraging algorithms [4] was combined to create a hybrid approach. A Genetic Algorithm [18], [19] is used to calculate the best values of predefined set of free parameters connected with either a process representation or a control vector. From regular optimization the global optimization global optimization [18] is distinguished by its focus on finding the maximum or minimum over all input values, as contrasting to verdict local minima or maxima. Since a foraging organism/animal takes actions to maximize the energy utilized per unit time. For various problems, the designer frequently has to be fulfilled with local optimal or suboptimal solutions.

Resource sharing [17] is the major issue of concerns for cloud computing is concerned. This work focuses on the skewness metrics which is the measure of unevenness in the usage of resources. Unevenness may mean the unplanful usage of resources. For example a minimum amount of resource is needed but the available bulk amount of resource is been allocated, in that case this excess amount of resource that is allocated is wasted unnecessarily. In such cases if resources are allocated appropriately it will save a remarkable amount of resources. This work takes this in to consideration and tires to reduce the overhead.

3. PM TO VM MIGRATION

In proposed system a VM migration concept was presented that reduce the overloading in the structure effectively. From that the energy utilized in the virtualization environment was reduced and ensuring high fault tolerance capacity. For dynamic resource allocation, virtual machine live migration is a widely used. To compute the virtual machine to physical machine migration first-fit approximation algorithm was used. For the offline use only the algorithm is used. But it is necessary to predict the large number of migration in online environment, because the resource requests of cloud user and virtual machines vary dynamically. Lot of experiments has to be conducted to decrease the energy utilization in the data center.

When an application is running it is important to adjust the mapping among virtual machine and physical machine. In a virtualized environment virtual machine migration is a frequently applied method for dynamic resource allocation. It sorts the record of physical machine depends on volumes of data and virtual machines in every physical machine in their volume-to-size ratio (VSR). Later physical machines and virtual machines are considered in presorted order. The capability of physical machines is a heterogeneous system because large number of hardware resources is available in the data center. Additionally Fault tolerance in a VM can be minimized by dropping the migration time.

4. SCHEDULING PROCESS

Figure 1 shows the number of physical server resources are combined to create a virtual machine. The allocated job/load is transferred to the VM scheduler. That VM scheduler may contain number of process to perform migration operation.

- Deployment
- Hot and cold spots identification
- Hot spot mitigation through VM migration
- Energy consolidation through load balancing

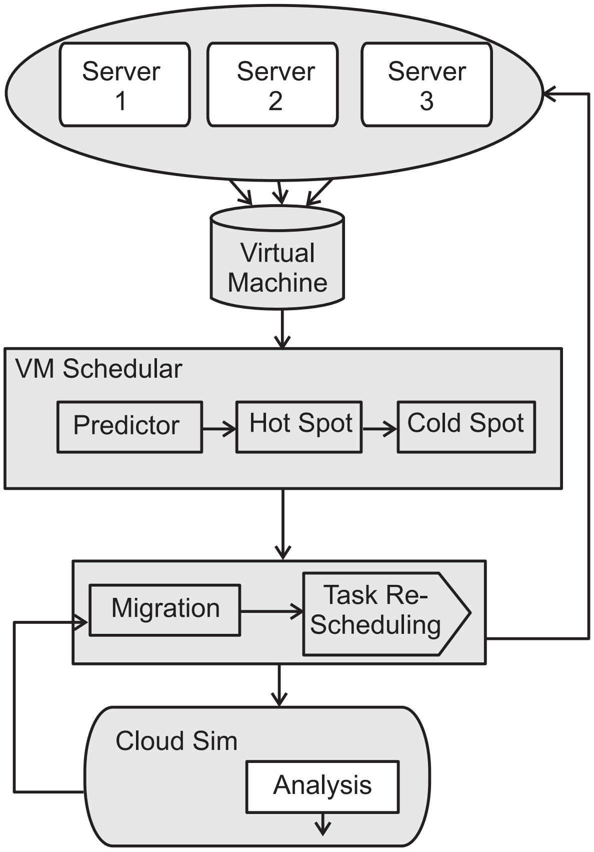


Figure 1: Scheduling System Architecture

4.1. Deployment

This stage involves the establishment of multiple data centres and list of VM's within each data centre. Each primary machine working inside the Xen hypervisor (VMM) works with a restricted domain 0 and one or more domain U. All virtual machine in field U encapsulates one or more applications. While monitoring the scheduling actions in Xen the CPU and network usage will be calculated. The memory used inside the virtual machine is not known to the hypervisor. Memory shortage of the virtual machine is calculated by observing the swap activities. Guest OS is compulsory to install a part of swap partition. Several components are included in the scheduler. Based on precedent information the predictor predicts the upcoming resource load of virtual machines and the future load of physical machines. The load of the virtual machine is calculated by calculating the entire load of virtual machine. Hot threshold value is fixed for identifying the overload of virtual machine. If the value is above the threshold value the virtual machines are migrated to reduce their load.

4.2. Identification of Hot and Cold Spots

The hot and cold spot of each machine is identified by predicting the resource demands and resource allocation of all virtual machine. If the utilization of a server is above the threshold value it will be marked as hot spot. The hot spot mention if the server is overloaded and some of the virtual machines running on the server should be migrated. Cold spot is identified based on the threshold value of if the utilization value is below the threshold. Server is continuously working if it has at least one virtual machine running. At last, we define the warm threshold to be a level of resource utilization that is sufficiently high to rationalize having the server running but not as high as to risk suitable a hot spot in the features of temporary fluctuation of application resource demands. Each type of resources can have different thresholds at the time of allocation. For instance, we can define the hot thresholds for CPU and memory resources to be 90 and 80 percent. At final a server is a hot spot if each its CPU working is beyond 90% or its memory handling is above 80 %.

4.3. Hot Spot Mitigation through VM Migration

Managing the record of hot spots in the system in descending temperature. We needs to eliminate the hot spot at the maximum. Otherwise the temperature is kept as low if possible. We have to decide which virtual machine is migrated for each server p. Virtual machines are sorted based on the resulting temperature of the server if that virtual machine is migrated away. Our aim is to migrate the virtual machine which can reduce the server's temperature the more. We are trying to catch the destination server to accommodate the virtual machine in the list. After accepting the virtual machine the server must not become a hot spot. From all the available servers, we choose one whose skewness can be reduced the most by accepting this virtual machine. This reduction of skewness can be leads to negative if the server skewness increases the least. The predicted load of selected server updated and recorded if a destination server is found. Otherwise, we move onto the next virtual machine in the list and try to find a destination server for it.

4.4. Energy Consolidation through Load Balancing

The VM should be migrated from one physical location to other. Virtual machine migration is checked for the cold spot p. Destination server is identified to accommodate a virtual machine on the cold spot p. After the acceptance of VM the resource utilizations of the server should be lower than the warm threshold. By consolidating the underutilized servers the exaggerated energy can be saved. To prevent this warm threshold is created. Multiple servers assure the exceeding condition, we choose one that is not a current cold spot. Increasing load on the cold spot can eliminate the resource request. The cold spot would be acknowledged in the destination server compulsorily. We select a destination server based on the reduction in the skewness value. If the destination servers were for all VMs on a cold spot, record the progression of migrations and update the predicted load of related servers. Otherwise, we do not migrate any of its VMs.

The list of cold spots is also updated because some of them may no longer be cold due to the proposed VM migrations in the above process. The extra loads can be added to the related server based on above consolidation. This is not as serious a problem as in the hot spot mitigation case because green computing is initiated only when the load in the system is low down. Restrictions in the number of cold spot can be eliminated at every execution of the algorithm and it cannot be move over the limit of algorithm. This can be termed as consolidation limit.

5. RESOURCE ALLOCATION

Resources on the cloud are not frequently shared by multiple cloud users other than dynamically re-allocated for every demand. Resources can be allocated and reallocated after the utilization of the user. The demand of the user may vary in time. But our aim is allocate the resources efficiently. The mentioned algorithm is able to work for allocating resources to cloud users in dissimilar time.

5.1. Load Prediction Algorithm

Future need of the resources preserve predicted using this algorithm. For calculating exponentially weighted moving average (EWMA) with a equation,

$$\mathrm{E}(t) = \alpha * \mathrm{E}(t-1) + (1-\alpha) * 0\ (t) \leq \alpha \leq 1, \qquad ()$$

Equation (1) states E(t) and O(t) are the approximate and experiential load at time t and α means an employment off among durability with responsiveness. The parameters in the parenthesis are α values. The length of the measurement window is termed as W. The “median” error is derived by the equation:

$$|\mathrm{E}(t) - \mathrm{O}(t)|/\mathrm{O}(t) \qquad (2)$$

The percentage of predicted values is high or low than the predicted value compared to observed value. To reflect the “acceleration,” we obtain a new method through setting α as a negative value. While $-1 \leq \alpha < 0$, the higher formula can be modified into the following:

$$\begin{aligned}\mathrm{E}(t) &= -|\alpha| * \mathrm{E}(t-1) + (1 + |\alpha| * \mathrm{O}(t) \\ &= \mathrm{O}(t) + |\alpha| * (\mathrm{O}(t) - \mathrm{E}(t-1)),\end{aligned}$$

5.2. Skewness Algorithm

For the measurement of irregularity of using resources in the server the skewness algorithm was introduced. Let n be the overall amount of resources utilized and r_i is the utilization of i^{th} resource. Where p is the skewness of the resources, it can be defined as,

$$\text{Skewness } (p) = \sqrt{\sum_{i=1}^{n}\left(\frac{r_{\mathrm{r}}}{\overline{r}} - 1\right)^2}, \qquad (3)$$

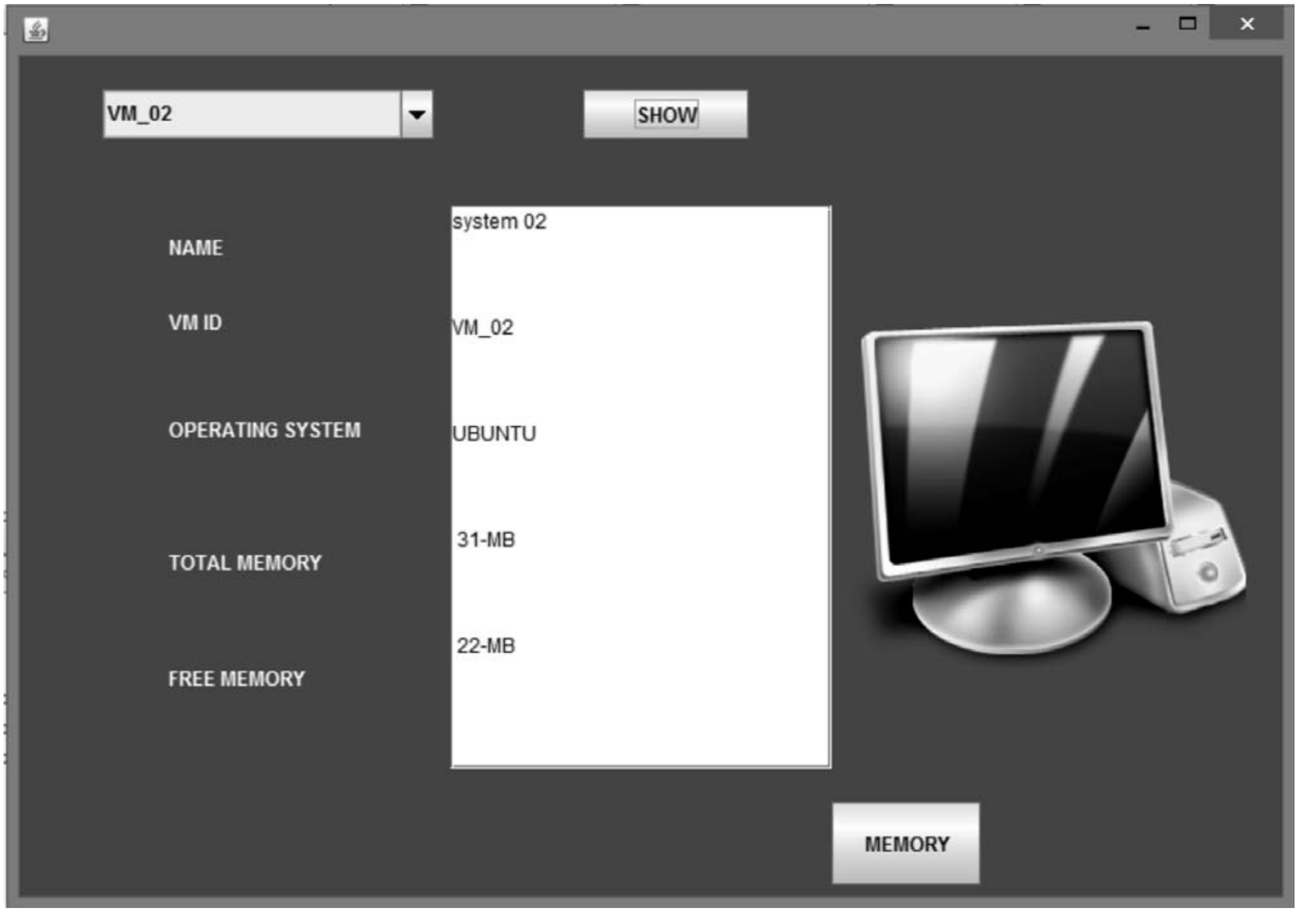
Where $\overline{r}$ is the average utilization of entire resources for server p. In preparation, not every types of resources are concert significant and therefore we only require to judge restricted access resources in the exceeding calculation. The minimization of skewness can combine different types of workloads accurately and reduce the overall utilization of server resources. Above equation (3) is used to calculate the skewness.

6. RESOURCE MANAGEMENT MODULE

Resource management is an important concept used in cloud scheduling. In place of considering migration process dynamic resource management applied. The cloud resources are used by the cloud user based on the cost and availability of particular resource. So effective and efficient management of resource is necessary for the cloud user and cloud provider. The resource management is based on the flexibility of resources. To get the efficiency of resource organization we need to isolate the cloud resources. Victorious resource management explanation used in virtual cloud environments wants to supply a prosperous pool of resource controls for improved isolation, for doing preface assignment and matching of load for competent exploitation of fundamental resources.

7. RESULTS AND DISCUSSION

The cloud environment was executed in Java. In this cloud environment we created set of virtual machines with different resources. The fig. 2 shows the VM selection from the set of VM's. Fig. 3 shows the datacenter allocated in the cloud. The datacenter is having X86 architecture, Linux operating system, Xen hypervisor and 40GB memory was allocated. The skewness process is performed in fig. 4. Based on the threshold value skewness operation is carried out. The status of three servers is showed in fig. 5. If the workload is overloaded in one server migration process is performed to balance the load. Fig. 6 shows the migration process. If the VM on the server is fully loaded, its color is changed into black. At last in fig. 7 the heap memory usage over time is showed. The time of the load balancing process was minimized compared to other works.



VM_02

NAME system 02

VM ID VM_02

OPERATING SYSTEM UBUNTU

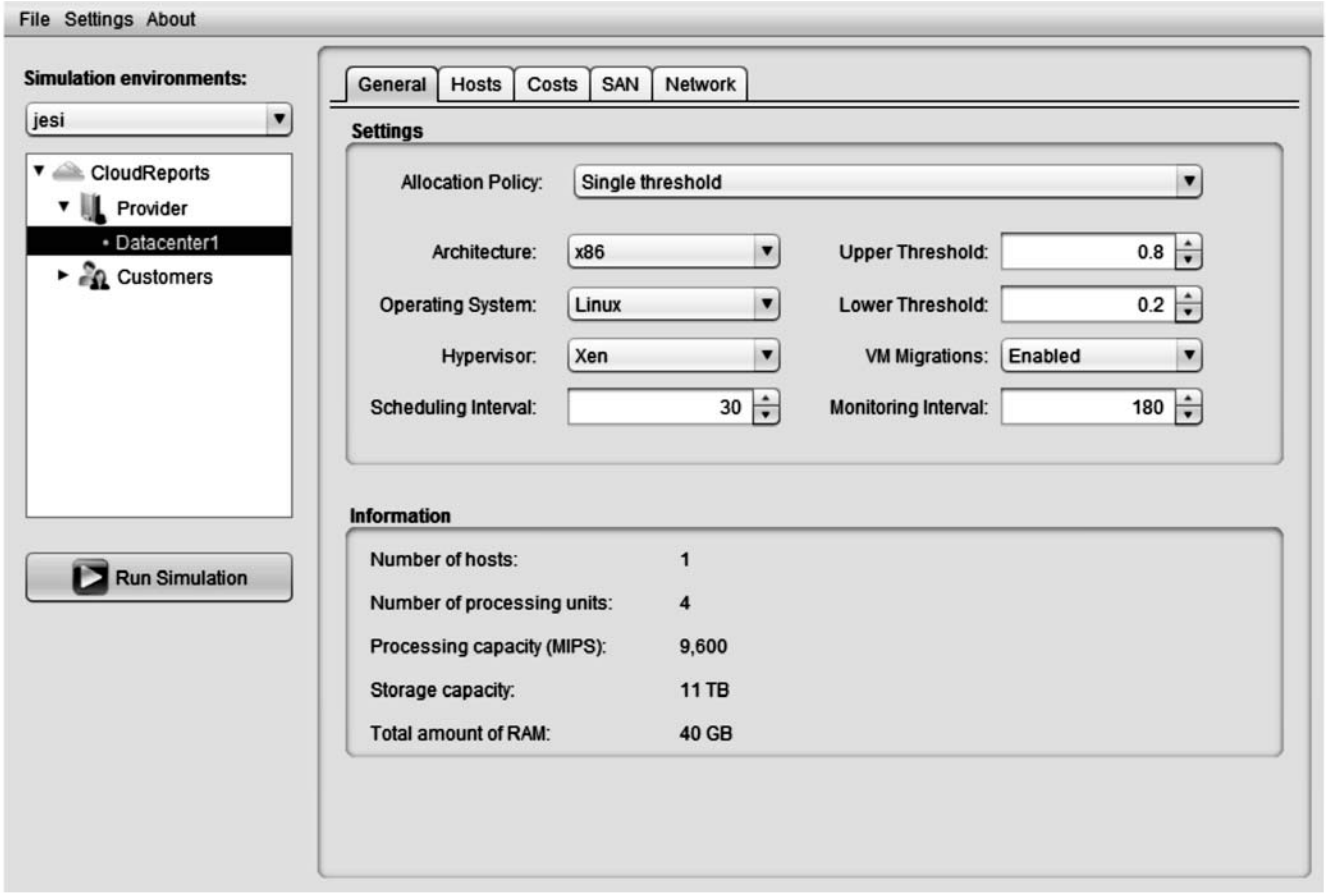
TOTAL MEMORY 31-MB

FREE MEMORY 22-MB

Figure 2: VM selection

8. CONCLUSION AND FUTURE WORK

Cloud virtualized environment using the virtual machine live migration process is working for dynamic resource allocation. Sometimes the service level agreements are violated, while performing migration process. In this paper metrics considered is skewness it is used to combine and well utilize the virtual resources. Skewness operation is performed with different threshold values. The skewness algorithm achieves overload avoidance with less time. In future we propose a multi-agent based VM migration system. When the single agent based migration is converted into a multi agent based migration the energy is saved and we may get the high fault tolerance capacity. At the time of considering huge data center virtual machine migration becomes a very Virtual Machine Migration become tough process. If any one specific datacenter virtual machine from one place to another place number of data packets are also need to be migrated. At that time the network elements are configured to reach the appropriate destination. To overcome these drawbacks multi-agent based migration will be used.



Settings

Allocation Policy: Single threshold

Architecture: x86

Upper Threshold: 0.8

Operating System: Linux

Lower Threshold: 0.2

Hypervisor: Xen

VM Migrations: Enabled

Scheduling Interval: 30

Monitoring Interval: 180

Information

Number of hosts: 1

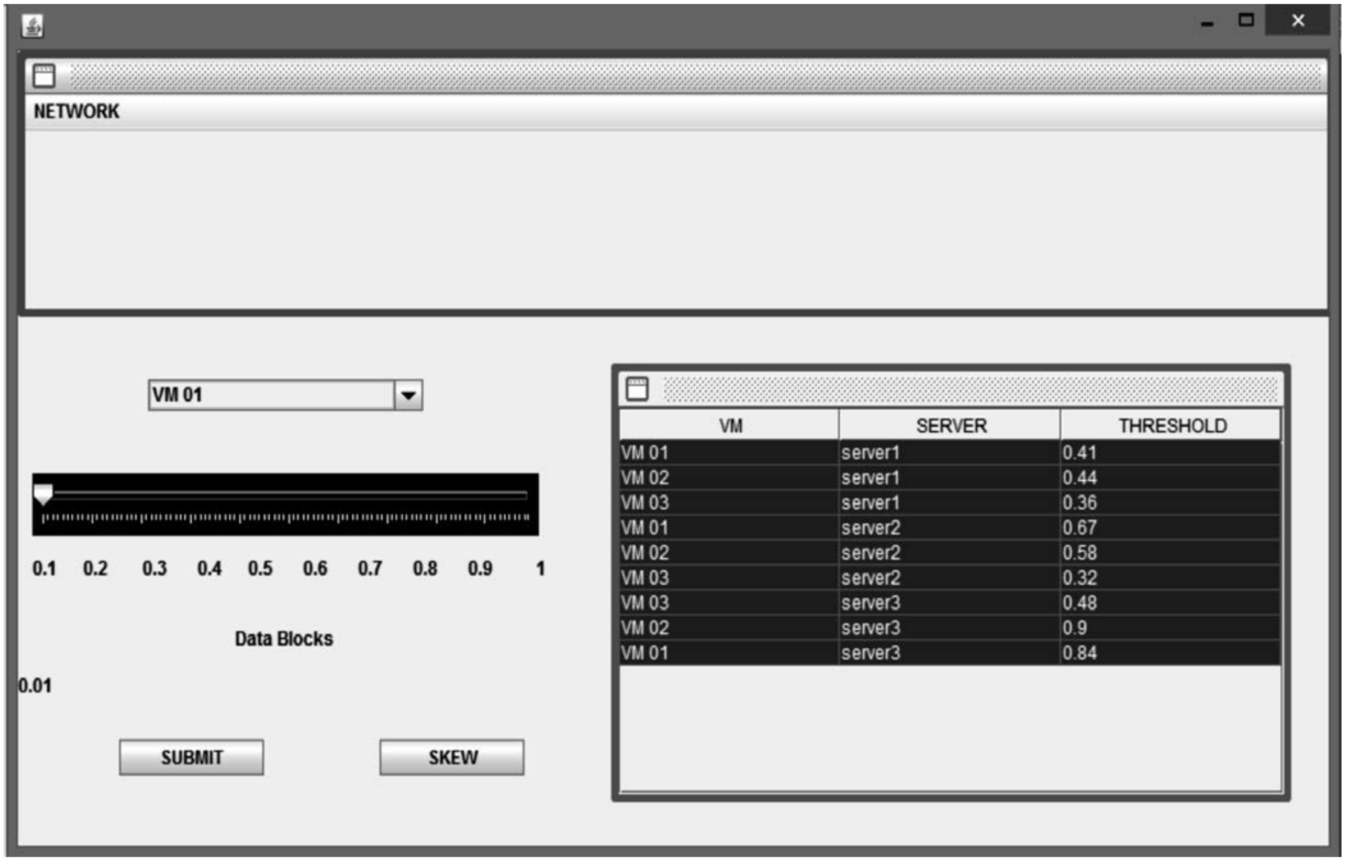
Number of processing units: 4

Processing capacity (MIPS): 9,600

Storage capacity: 11 TB

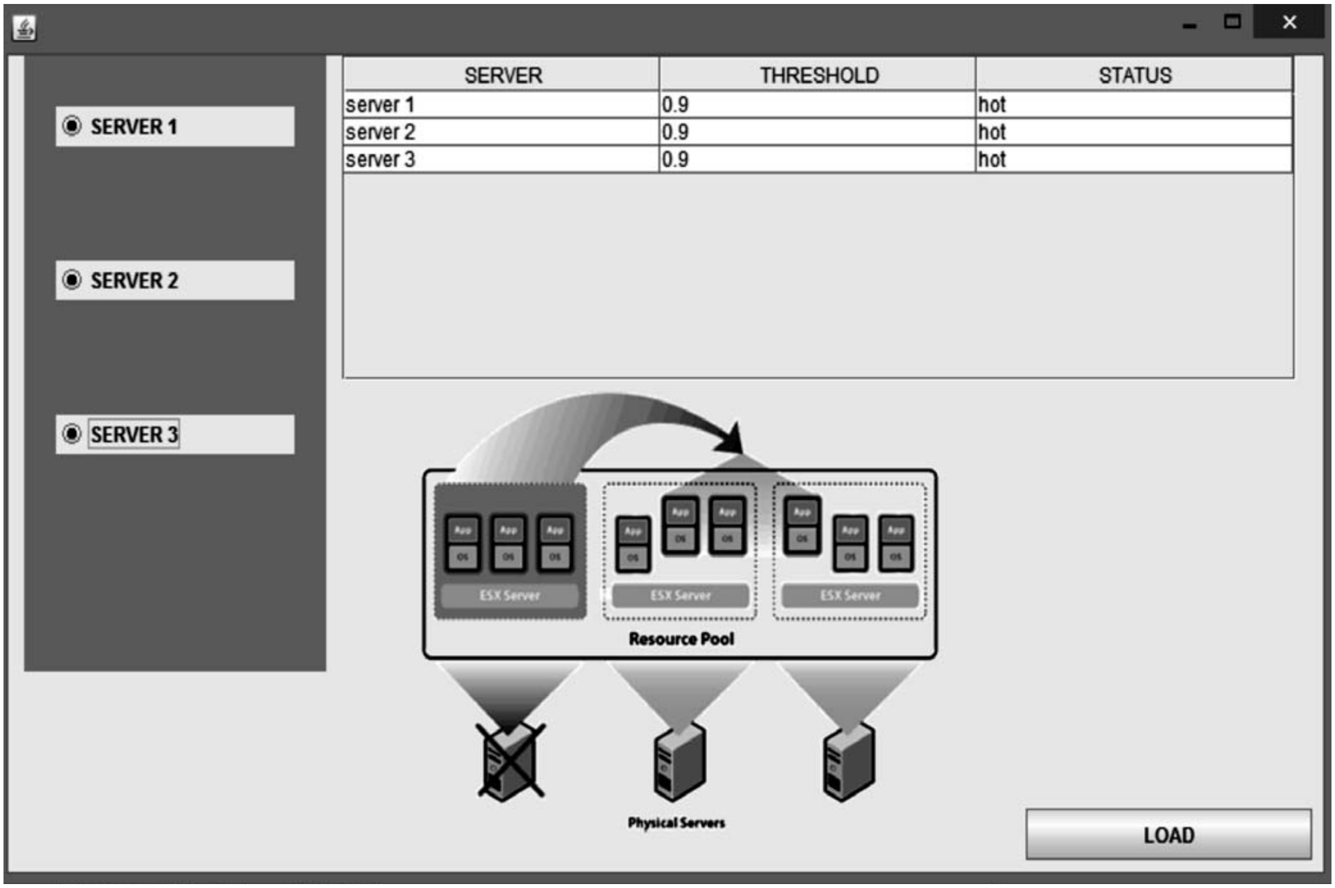
Total amount of RAM: 40 GB

Figure 3: Resource allocation in datacenter



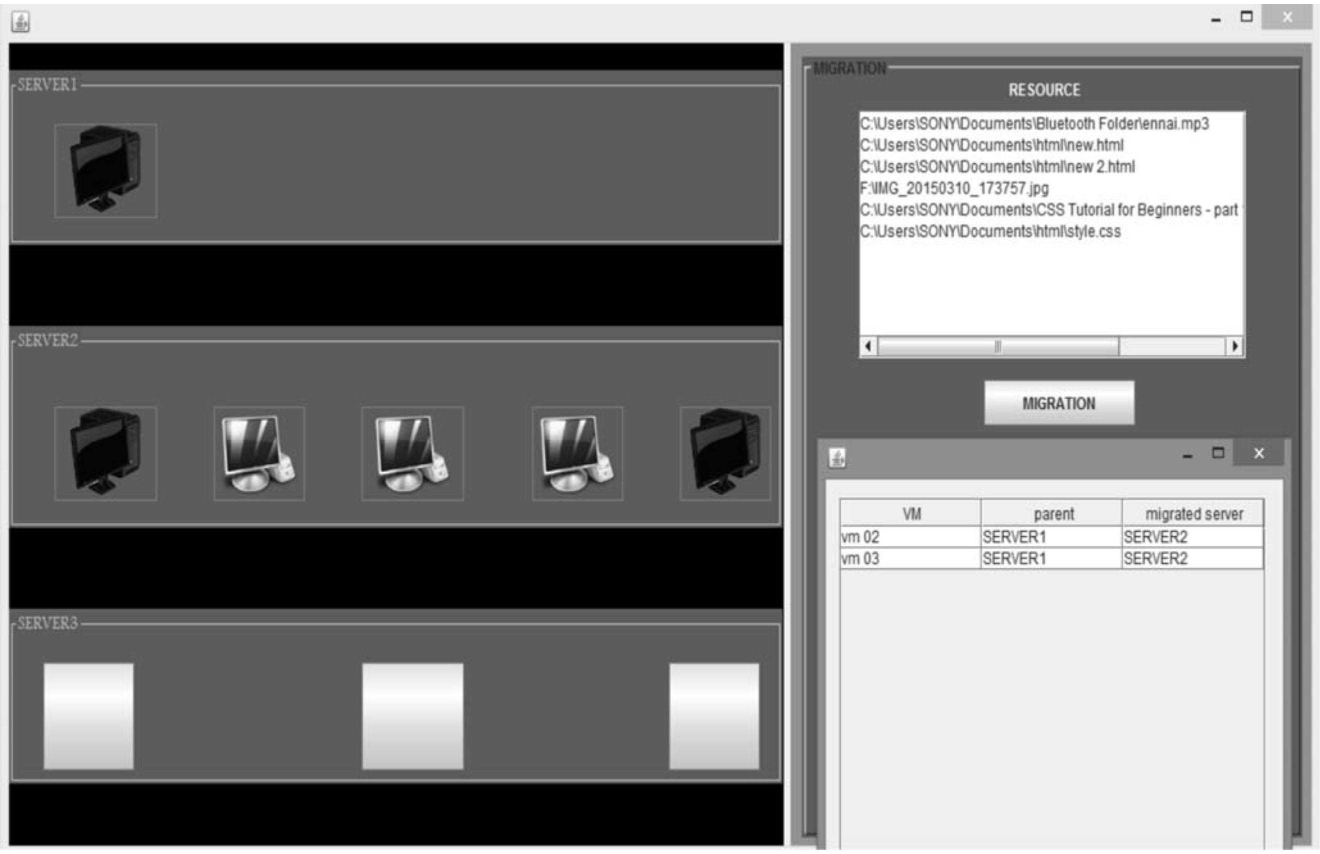
VM	SERVER	THRESHOLD
VM 01	server1	0.41
VM 02	server1	0.44
VM 03	server1	0.36
VM 01	server2	0.67
VM 02	server2	0.58
VM 03	server2	0.32
VM 03	server3	0.48
VM 02	server3	0.9
VM 01	server3	0.84

Figure 4: Skewness



SERVER	THRESHOLD	STATUS
server 1	0.9	hot
server 2	0.9	hot
server 3	0.9	hot

Figure 5: Status of servers



C:\Users\SONY\Documents\Bluetooth Folder\ennai.mp3
C:\Users\SONY\Documents\html\new.html
C:\Users\SONY\Documents\html\new 2.html
F:\IMG_20150310_173757.jpg
C:\Users\SONY\Documents\CSS Tutorial for Beginners - part
C:\Users\SONY\Documents\html\style.css

VM	parent	migrated server
vm 02	SERVER1	SERVER2
vm 03	SERVER1	SERVER2

Figure 6: Migration process

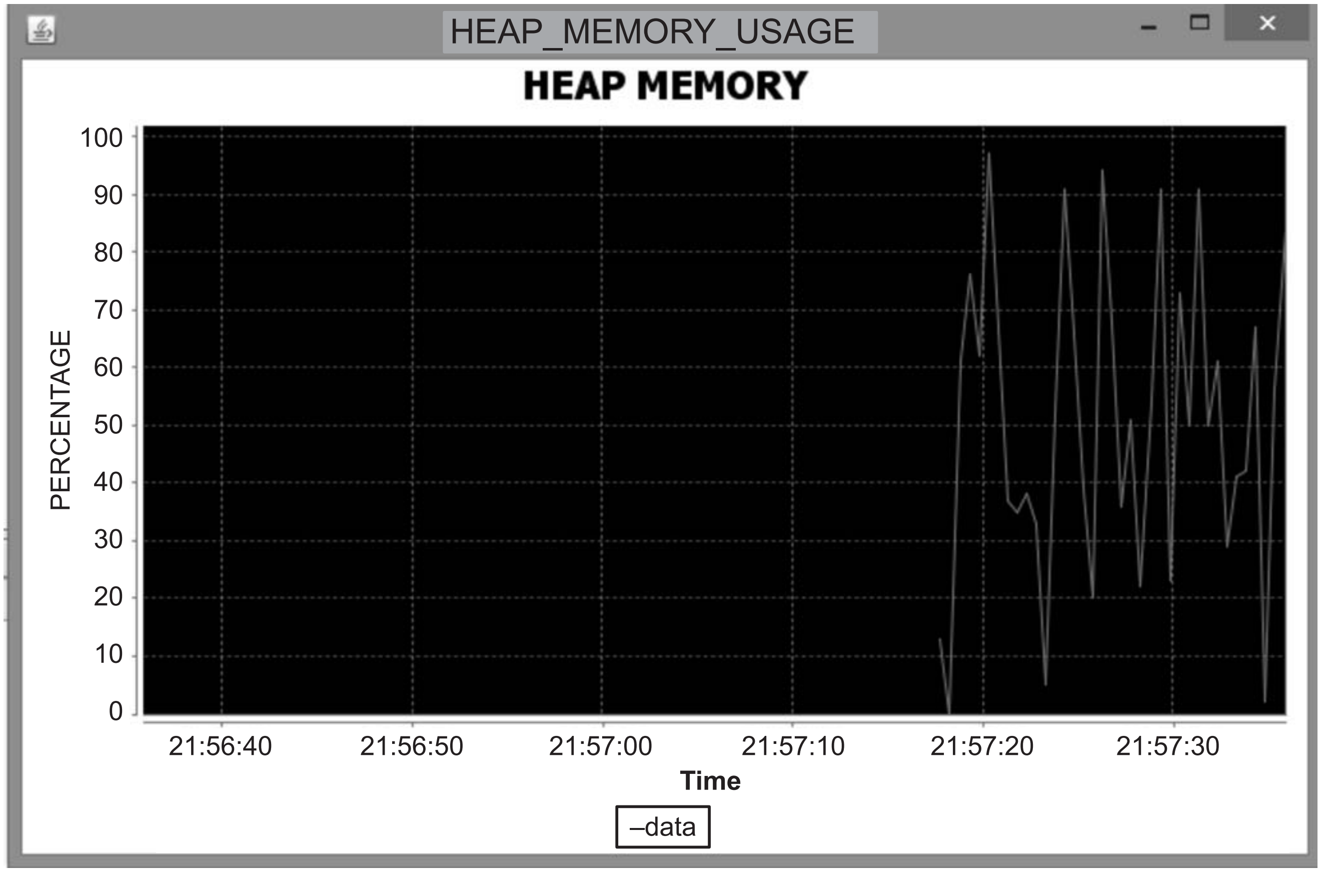


Figure 7: Heap Memory usage over time

9. REFERENCES

1. Zhenxiao, weijia song, and qi chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment",IEEE Transactions on parallel and distributed systems, Vol. 24, No. 6, June2013.

2. Pandey S, Guru SM Buyya R. "A particle swarm optimization-based heuristic for scheduling work flow applications in cloud computing environments", The 24th IEEE international conference on advanced information networking and applications, 2010,pp.400–407.

3. M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flow time minimization in the permutation flowshop sequencing problem", European Journal of Operational Research, March 2007, pp. 1930–1947.

4. Anusha Bamini.A.M, Dr. Sharmini Enoch, "Optimized Resource Scheduling Using Classification and Regression Tree and Modified Bacterial Foraging Optimization Algorithm", International Journal of Applied Engineering Research,Vol.10, No 16, 2015, pp. 37170-37175.

5. Qinghai Bai, "Analysis of Particle Swarm Optimization Algorithm", Computer and Information Science, Vol. 3, I.1, Feb. 2010.

6. SmithaSundareswaran, Anna C. Squicciarini, and Dan Lin, "Ensuring distributed accountability for data sharing in the cloud",IEEE Transactions on dependable and secure computing, vol. 9, no. 4, July/august 2012.

7. J. Octavio Gutierrez-Garcia, KwangMongSim,"A Family of Heuristics for Agent-Based Elastic Cloud Bag-Of-Tasks concurrent Scheduling", Journal of Elsiever, future generation computer systems, 2012.

8. Jaspreetkaur,"Comparison of load balancing algorithms in a Cloud", International Journal of Engineering Research and Applications, Vol. 2, Issue 3, May-Jun 2012.

9. Shaminder Kaur, Amandeep Verma, "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment", International Journal ofInformation Technology and Computer Science, Vol. 10, 2012, pp.74-79.

10. ShamsollahGhanbaria, "A Priority based Job Scheduling Algorithm in Cloud Computing", Journal of Elsiver international conference on advance science and contemporary engineering, 2012.

11. P.-Y. Yin, S.-S. Yu, and Y.-T. Wang, "A hybrid particle swarm optimisation algorithm for optimal task assignment in distributed systems", Computer Standards and Interfaces, Vol. 28, I.4, 2006, pp. 441–450.

12. M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem", European Journal of Operational Research, Vol. 177, I. 3, March 2007, pp. 1930–1947.

[13. Wei Wang, GuosunZeng, Daizhong Tang, Jing Yao, "Cloud-DLS: Dynamic trusted scheduling for Cloud computing",Journal of Elsiver,2011.

14. G.Sireesha, L.Bharathi, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud", 2011.

15. SandeepTayal, "Task scheduling optimization for the cloud computing systems", International Journal of Advanced Engineering Sciences and Technologies Vol. 1, No. 5, Issue No. 2, 2011, pp. 111-115.

16. W. Chen, J. Zhang, "An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements", IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, Vol. 39, No. 1, January 2009.

17. P. Delias, A. Doulamis, N. Doulamis, and N. Matsatsinis, "Optimizing resource conflicts in workflow management systems", IEEE Trans. Knowl. Data Eng., Vol. 23, No. 3, Mar. 2011, pp. 417–432.

18. J. Kolodziej and S. U. Khan, "Multi-level Hierarchical genetic based scheduling of independent jobs in dynamic heterogeneous grid environment," Inf. Sci., Vol. 214, 2012, pp. 1–19.

19. J. Yu and R. Buyya, "Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms," Sci. Program., Vol. 14, 2006, pp. 217–230.