IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.6, Special Issue.11, Dec 2018 E-ISSN: 2347-2693

Comparison between Round Robin and Virtual Migration Algorithm Based on their Energy Efficiency

P.S.S. Akilasri[1*], K. Meenakshi[2]

[1,2]Department of Computer Science, National College, Trichy, India

[*]*Corresponding Author: kmoorthymeena@gmail.com*

Available online at: www.ijcseonline.org

Abstract— Cloud Computing is an important field in today's computer world. Data are entered the server through various algorithms and each algorithm has its own advantages and disadvantages. In this paper two algorithms are compared, and their energy efficiency based on their time taken is found and a graph is tabulated. The algorithms are Round Robin and Virtual Machine migration algorithms.

Keywords— Cloud Computing, Machine Migration, Virtual data Round Robin

I. INTRODUCTION

The term Cloud refers to a Network or Internet. In other words, we can say that Cloud is something, which is present at remote location. Cloud can provide services over network, i.e., on public networks or on private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM), all run in cloud.Cloud Computing is one of the latest trends in Information Technology sector, where the computational requirements are provided as a Service to end users via Internet. Since Internet is denoted with cloud symbol, this type of computing is said to be Cloud Computing. It is also called as Internet Computing and On-Demand Computing. All the Services provided to the end user in "On-Demand basis". All the on-premise service are migrated and served via Internet in Cloud Computing.

DEPLOYMENT MODELS

Private Cloud

Private cloud is the basic cloud deployment model, which is implemented for a limited size organization. The number of users, using the private cloud will be limited. Private cloud is more secured than other deployment models. Private cloud is generally built for internal purpose. Microsoft Azure, Amazon Web Services are the leading service provider for private cloud.

Community Cloud

Community cloud is just an advanced version Private Cloud. Community Cloud is generally built to connect two or more organizations and share their computational requirements. Community cloud enables clients deliver projects across various organizations. Community cloud cannot be accessed beyond the users of the organization which has formed the community cloud.

Public Cloud

Public cloud is the biggest version of all the cloud deployment models. Public cloud is generally built by large organization to deliver their services to large group of audience. With proper authentication, anyone can use the services provided by the public cloud. Public cloud is little insecure than other forms of cloud deployment models. Since the cloud is open to the public, security breaches and security vulnerabilities are more in Public cloud.

Hybrid Cloud

Hybrid cloud is one of the biggest solution for the security and performance issue of the public cloud. Hybrid cloud is the combination of one or more cloud deployment models. Hence it has the property of all the cloud deployment models. Hybrid cloud enables the service providers to provide certain resources to the general audience and limit certain services to the particular group of audience.

II. LITERATURE SURVEY

D. Breitgand(et.al.,)[4],describes implement our proposal computationally and storage efficiently, while maintaining sufficient accuracy, we propose a simple method of estimating total effective nominal demand ofa cloud and use it for capacity sizing and placement reservation plan that is compliant with eSLA. In approach, a momentary nominal demand of a workload is calculated as the number of VM instances of different types being run by the workload at a specific time instance. Summing up the momentary nominal demand of all users, obtain the momentary nominal demand of the cloud and propose a simple method for calculating

effective nominal demand from the time series of the momentary nominal demand. While conceptually similar to an over-commit strategy that over-subscribes hosts' resources based on the actual utilization levels of VMs that received more attention in the literature recently, treat an entire pool, datacenter or distributed cloud as a single overcommit domain, while respecting nominal capacity allocations of VM types and proposed method can handle multi-dimensional resource allocation simultaneously by reserving placement slots for VMs according to the nominal capacity specification. While such approach may result in lower over-commit ratios than those achievable with per host actual utilization over-commit, our method offers much more robust performance guarantees (via respecting nominal allocations), requires dramatically less amount of monitoring (actual demand over-commit methods require per VM monitoring of utilization), keeps VM migrations to the minimum, requires less computational power and is more transparent to the cloud customer.AlthoughPareto-efficient strategies have been investigated before indifferent contexts, they are generally considered too computationally-intensive for online scheduling scenarios.However, we show here that even low-resolution searches for Pareto-efficient strategies benefit our scenario of scheduling large numbers of tasks online and allow for loose connectivity between the user scheduler and the hosts. In such loosely connected systems (e.g., desktop grids using the BOINC BoT runtime system), the exact time of a failure may be unknown.

III. PROPOSED WORK

In this paper, we present the design and implementation of an automated resource management system that achieves a good balance between the two goals. We make the following contributions. We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used. We introduce the concept of "fuzzy assessment" to measure the uneven utilization of a server. By analyzing risk assessment, we can improve the overall utilization of servers in the face of multidimensional resource constraints. We design overbooking algorithm that can capture the future resource usages of applications accurately without looking inside the VMs. The algorithm can capture the rising trend of resource usage patterns and help reduce the placement churn significantly. In order to obtain an optimal solution for a simplified version of the resource allocation problem and an efficient heuristic this approach provides the PID controller which gives the important contributions to this proposed system. The controller used in this system executes in middleware platform. The protocol ensures three design goals namely fairness, adaptability and scalability. It evaluates heuristic system through simulation and its performance to be well aligned with the designed goals. In this system, global synchronization can be avoided as there is a single continuous executed instead of sequences of executions with restarts.

1. Virtual Machine Algorithm

Resource overbooking is an admission control technique to increase utilization in cloud environments. However, due to uncertainty about future application workloads, overbooking may result in overload situations and deteriorated performance.. One way of addressing those problems and increasing resource utilization is resource overbooking. In essence, the provider allocates more capacity than the real capacity of the data center. In other words, a new VM is admitted although the sum of requested cores or memory exceeds the number of cores or total memory in the data center. However, such an approach may lead to resource overload and performance degradation. Therefore, besides carefully choosing how to place VMs on physical machines, anew resource management challenge appears: estimating the appropriate level of overbooking that can be achieved without impacting the performance of the cloud services. Admission control techniques are therefore needed to handle this tradeoff between increasing resource utilization and risking performance degradation. Combining statistical multiplexing of resource demands, server consolidation and economy of scales, cloud providers are able to offer users resources at competitive prices. Users often exaggerate the sizes of the Virtual Machines (VMs) they lease, either because the provider forces them to use predefined sizes, common practice, or to compensate for uncertainty. Hence, a provider could practice overbooking: An autonomic admission controller selects whether to accept a new user application or not, based on predicted resource utilization, which is likely smaller than the requested amount of resources. Overbooking is beneficial both to the provider, who can gain a competitive advantage and increase profits, and the user, who may observe lower prices. Although combining overbooking and brownout may seem straight-forward, the two approaches should not be used without thorough evaluation. Indeed, the two autonomic feedback loops, belonging to the brownout application and the overbooking provider, may take conflicting decisions, which may degrade performance. By contrast, if both approaches are effectively combined, the overbooking system may take advantage of the application performance knowledge from brownout, and use both reactive and proactive methods to avoid overload situations.

This algorithm first evaluates the risk associated to the new incoming request by calling the fuzzy risk assessment module. Once the associated risk is known, the admission control obtains the current (new) risk thresholds for the whole data center. Finally, it is checked, for each capacity dimension, if the risk of accepting the new incoming request is below the currently acceptable level and if so, the request is accepted. The process to calculate the service acceptance

risk and the data center risk thresholds. The risk assessment module provides the Admission Control with the information needed to take the final decision of accepting or rejecting the service request, as a new request is only admitted if the final risk is bellow a pre-defined level (risk threshold).Calculating the risk of admitting a new service includes many uncertainties. Furthermore, choosing an acceptable risk threshold has an impact on data center utilization and performance. High thresholds result in higher utilization but the expense of exposing the system to performance degradation, whilst using lower values leads to lower but safer resource utilization. This method of choosing the representative risk thresholds for the data center balances utilization in all capacity dimensions. If capacity is imbalanced, e.g., CPU utilization is greater than memory; the admission control can act on this fact and admit applications that request more capacity of the type that is further from the target utilization level. Algorithm

1. input:
2. Hostlist ,Vmlist //Sorted Desc
3. Curent_Time
4. Link_Speed
5. VmMigration_Time
6. VmMigrationList_Time
7.
8. For i:0 to Hostlist
9. host: Host_LargSize in Hostlist
10. while host>0
11. vm: VM_ LargSize in Vmlist
12. for j:1 to Vmlist

13. If vm>host then
14. vm: vm++ in vmlist

15. else

16. host:host - vm (size)
17. vm is in Migration
18. VmMigrationList_Time:Curent_Time +

19. (vm/Link_Speed)
20. vm:vm++ in Vmlist
21. host:host++ in Hostlist

2. Round Robin Algorithm

Round Robin scheduling algorithm is one of the simplest and most used scheduling algorithm up to this moment. The concept of this algorithm is to share the CPU time among all scheduled tasks on a ready queue. The most important aspect of the Round Robin algorithm is the time slice (Time Quantum) that will be allocated to each task submitted for execution. While the time quantum is a decisive characteristic on the Round Robin algorithm, several proposed Round Robin based algorithms are suggesting static time quantum that segments the CPU time among all submitted tasks, nevertheless, a static time quantum is not always the best solution. A more viable alternative is the use of dynamic time quantum that adapts the CPU time slices to the tasks changes happening on the ready queue for execution. Under the same topic, the Round Robin based algorithm proposed on this paper uses a dynamic time quantum and adds a smarter layer to the existing algorithm in order to adjust the CPU time to different situations.

3. Advantages

1.Allowsvoice calls at zero cost.

2.It is highly secured.

4.Easy to work and implement.

3.Does not need applications to work.

5.Does not require any extra hardware or software to installed in the device.

6.Handover is not monitored.

IV. RESULT AND DISCUSSION

In this project the Round Robin Technique is used to find the minimum, maximum and the average time taken for a bunch of users who sent the data in to the data center. There can be n-number of users who send the data at the same time. The fig 5.1 shows the data entering into the cloud through the user.

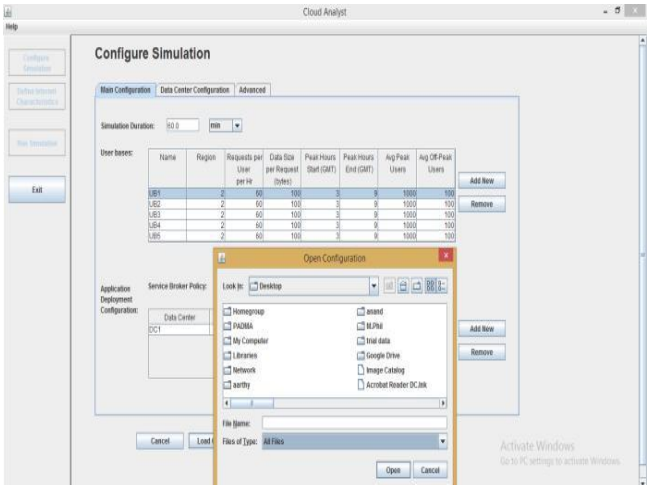

Fig: 5.1 User base

After uploading the data, it must go to the data center where the data can be stored for future use. The data switch from one data center to another. Because if one crashes the user can use another.fig: 5.2 show the simulation. The reports of the round robin method are displayed in the fig: 5.3.

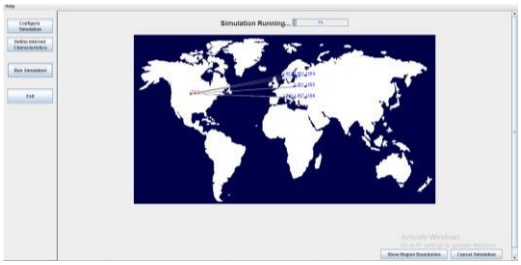

Fig: 5.2 Simulation using round robin

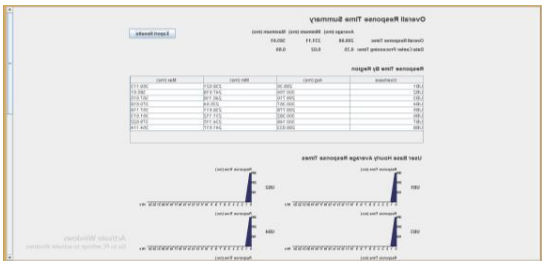

Fig: 5.3 Results of Round Robin Technique

The user assigns the file that must be uploaded. The size of the files and the packets of the file data are displayed. A secret key is used so that the data sent is to be secured.

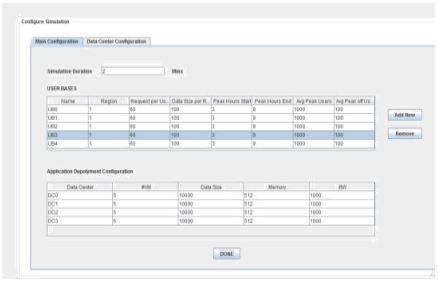

Fig:5.4 n-users- virtual migration

The data are passed from the user to the data server via Virtual Machine Migration the data is passed through several data servers as shown in the fig. 5.5. The report sof the time taken for the data are shown in fig. 5.6.

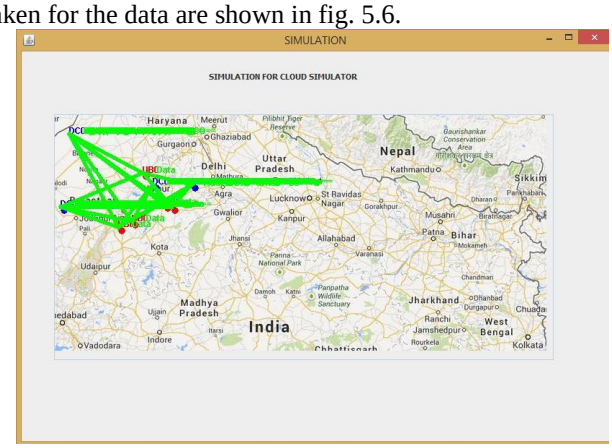

Fig: 5.5 Simulation in virtual migration technique

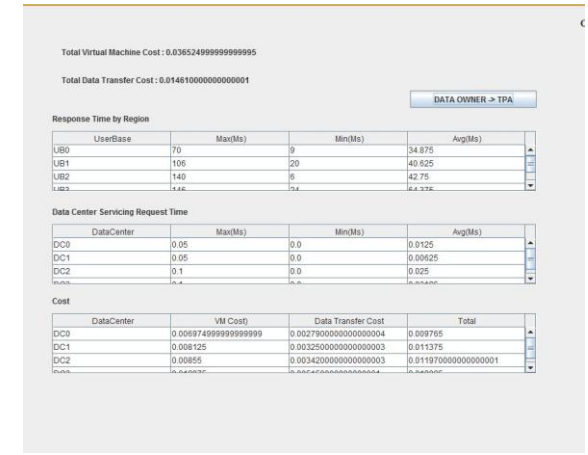

Fig: 5. 6Virtual migration response time

The minimum maximum and the average time taken for the data to move from the used system to data server through both the algorithms are noted. The average data of both the algorithms are tabulated. The graph shows the results of the different users, passing the same data through different algorithms and it is noted that the virtual migration is much faster than the Round Robin algorithm. From there is an increase in the efficiency in the data while it is passed through virtual migration.

Tab: 1.1 Average times in (ms) between virtual migration and round robin

User Base	Virtual migration(ms)	Round Robin(ms)
	34.875	299.466
UB1	40.625	301.029
UB2	42.75	299.277
UB3	64.375	300.564
UB4	50.25	300.207

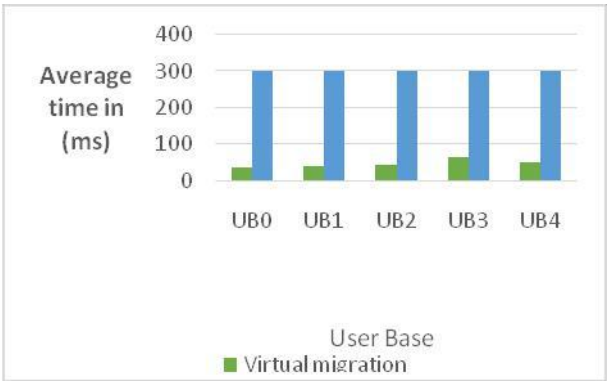

Fig: 5.7 Time difference between Virtual migration and Round robin Algorithm

V. CONCLUSION

Cloud computing is used by many fields in present day. The amount of data that is migrating to the cloud has been increased to a great deal. There are different types of algorithm through which the data can be passed through the cloud. But the efficiency varies upon the algorithm capability. Some algorithms pass the data to the data server much quick then another. The round robin algorithm is used to find the time efficiency for the data. Only text data are used to check the efficiency of the algorithm. The difference in the efficiency of the time between the two algorithms is found to be **256.97**(Round Robin-299.72 & virtual-42.72) for the UB2. The virtual machine migration reveals that the time taken is very less and maintaining a secret key which also helps to secure the file. This project only focuses on the text files which can be further enhanced to image, and video files in the future. These files are of a larger size and maintaining the data through varies data server can be a tedious in case of these files.

REFERENCES

[1] A. Ali-Eldin, J. Tordsson, and E. Elmroth, "An adaptive hybrid elasticity controller for cloud infrastructures," in Proc. of Network Operations and Management Symposium (NOMS), 2012, pp. 204–212.

[2] A. Sulistio, K. H. Kim, and R. Buyya, "Managing cancellations and no-shows of reservations with overbooking to increase resource revenue," in Proc. of Intl. Symposium on Cluster Computing and the Grid (CCGrid), 2008, pp. 267–276.

[3] L. Tom´as and J. Tordsson, "Improving Cloud Infrastructure Utilization through Overbooking," in Proc. of ACM Cloud and Autonomic Computing Conference (CAC), 2013.

[4] "Cloudy with a chance of load spikes: Admission control with fuzzy risk assessments," in Proc. of 6th IEEE/ACM Intl. Conference on Utility and Cloud Computing, 2013, pp. 155–162.

[5] K. J. A° stro¨m and R. M. Murray, Feedback Systems: An Introduction for Scientists and Engineers. Princeton University Press, 2008.

