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

Reducing Energy Efficiency through Virtual Machine Migration

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

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

Available online at: www.ijcseonline.org

Abstract — Cloud Computing is one of the important fields in today's computer world. Many data are being entered in to the cloud network from time to time. As a result, the traffic in the cloud system is also increased. To reduce the energy many algorithms are used. The Virtual Machine Migration algorithm is an algorithm which helps in reducing the energy to a greater deal. The data passed through the cloud through various users are measured and the time efficiency is found and discussed.

***Keywords*—** Cloud Computing, Machine Migration, Virtual data.

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 organization. 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

L. Wu,(et.al.,)[3], provides solutions to the above questions by proposing an innovative cost-effective admission control and scheduling algorithms to maximize the SaaS provider's profit. Our proposed solutions are able to maximize the number of accepted users through the efficient placement of request on VMs leased from multiple IaaS providers. We take into account various customer's QoS requirements and infrastructure heterogeneity. The key contributions of this paper are twofold: the proposed system and mathematical models for SaaS providers to satisfy customers. And proposed three innovative admission control and scheduling algorithms for profit maximization by minimizing cost and maximizing customer satisfaction level. In ProfminVM algorithm, a new user request does not get priority over any accepted request.

The inflexibility affects the profit of a SaaS provider since many urgent and high budget requests will be rejected. Thus, ProfRS algorithm reschedules the accepted requests to accommodate an urgent and high budget request. The advantage of this algorithm is that a SaaS provider accepts more users utilizing initiated VMs to earn more profit. The MinResTime algorithm selects the IaaS provider where new request can be processed with the earliest response time to avoid deadline violation and profit loss, therefore it minimizes the response time for users. Thus, it is used to know how fast user requests can be served. The StaticGreedy algorithm assumes that all user requests are known at the beginning of the scheduling process. In this algorithm, we select the most profitable schedule obtained by sorting all the requests either based on Budget or Deadline, and then using ProfPD algorithm. Thus, the profit obtained from StaticGreedy algorithm acts as an upper bound of the maximum profit that can be generated.

III. PROPOSED SYSTEM

The proposed system considers the process of resource management for a large-scale cloud environment. Such an environment includes the physical infrastructure and associated control functionality that enables the provisioning and management of cloud services. The perspective we take is that of a cloud service provider, which hosts sites in a cloud environment. The cloud service provider owns and administrates the physical infrastructure, on which cloud services are provided. It offers hosting services to site owners through a middleware that executes on its infrastructure. Site owners provide services to their respective users via sites that are hosted by the cloud service provider. Therefore, the user demands are transformed to this virtual cloud server. Through this efficient method, the user's demands will be satisfied successfully by serving the customer without waiting. Therefore, the resources will be allocated dynamically.

In this paper, 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.

Scalability is obtained, which indicates that all metrics considered for this evaluation are independent of the system size. In other words, if the number of machines grows at the same rate as the number of sites, while the CPU and memory capacities of a machine, as well as all parameters characterizing a site, such as demand, number of modules, etc., stay the same then it expects all considered metrics to remain constant.

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

Advantages

1. Allows voice calls at zero cost.
2. It is highly secured.
3. Does not need applications to work.
4. Easy to work and implement.
5. Does not require any extra hardware or software to installed in the device.
6. Handover is not monitored.

IV. RESULT AND DISCUSSION

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.

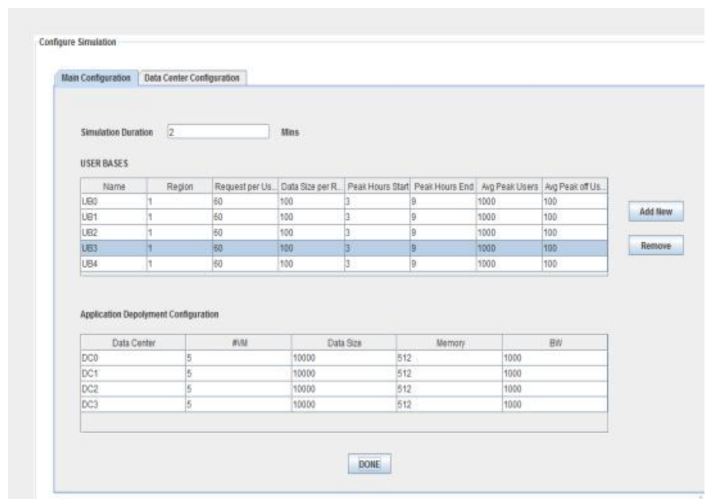

Fig:5.1 n-users- virtual migration

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. 5.2. The report sof the time taken for the data are shown in fig. 5.3.

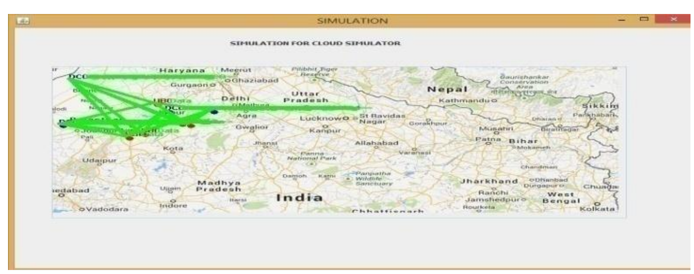

Fig: 5.2 Simulation in virtual migration technique

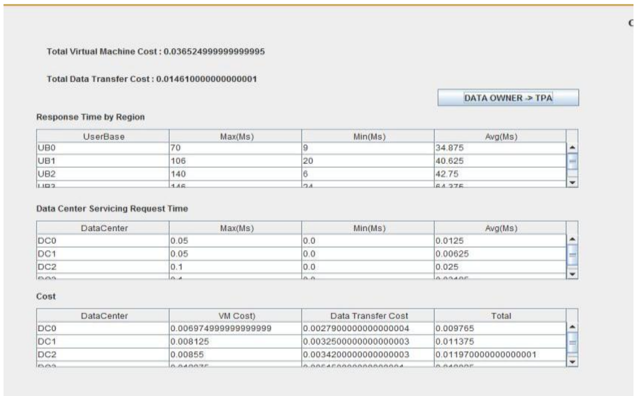

Fig: 5.3 Virtual migration response time

V. CONCLUSION

In this paper the time through which the data is passed increased. More data can be passed at lesser time. Only text data are checked in this paper. It can be further modified tot video and audio files. The main objective is to minimize the energy resource that is being spend in the data server and data passing maintenance.

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.