Minimizing Response Time in Cloud Computing Based upon Optimized Fuzzy Based Artificial Neural Network

Navjit Kaur, Rachhpal Singh
Punjabi University Regional Centre for Information Technology and Management, Mohali

***Abstract*—** Cloud Computing is an emerging technology in the field of Information Technology. This research aims towards the establishment of performance qualitative analysis on load sharing in VM to VM and then implemented in CloudSim with Java language. Here major stress is given on the study of resource allocation algorithm with heterogeneous resources of the cloud, followed by comparative survey of other algorithms in cloud computing with respect to scalability, homogeneity or heterogeneity and process migration. A previous study also indicates change of MIPS will affect the response time and increase in MIPS versus VM decreases the response time. When image size of VM is implemented against the VM bandwidth then no significant effect is found on response time and it remains constant for which these parameters are investigated. But in case of Cloudlet long length versus Host bandwidth a pattern is observed in which response time increases in proportionate manner. Using the modified approach the reduction in the down time of the various processes is achieved as shown in results.

Keywords - Cloud, VM, Host, VM Placement Schemes

I. INTRODUCTION

Cloud computing takes virtual foundation and expands upon look into in circulated computing, lattice computing, utility computing, autonomic computing, organizing, web administrations and programming administrations. It has demonstrated huge potential to strengthening, dexterity, multi-occupancy, dependability, versatility, accessibility, execution, security and upkeep. Through Cloud condition Email, Instant informing, business programming, and web content administration can be advertised. It fuses many existing innovations, for example, data and framework comprising of pools of PCs, systems, disseminated administrations application, data and capacity assets [1]. As of late, numerous Internet specialist organizations have constructed their own particular server farms to process the consistently expanding enormous information. It is evaluated that the server farms will devour more than 100 billion kWH every year [1], and the vitality cost of server farms will twofold at regular intervals [2]. The most famous arrangement is to unite VMs to as couple of servers as could be allowed, with stayed sit servers stop. Truth be told, substantially more power can be spared if the power utilization of each VM can be precisely measured. Along these lines, VM control metering is critical for the power sparing of green server farms. Furthermore, it is sensible to charge clients as indicated by the power utilization of their VMs. In any case, there are a few difficulties to overcome for VM control metering. VM is running at the level of programming, with the goal that conventional equipment control meter can't be utilized. It is difficult to quantify the power utilization of virtual gadgets like CPU, memory and circle that has a place with a VM. Furthermore, it is important to recognize the extent of equipment assets utilized by each VM.[2]

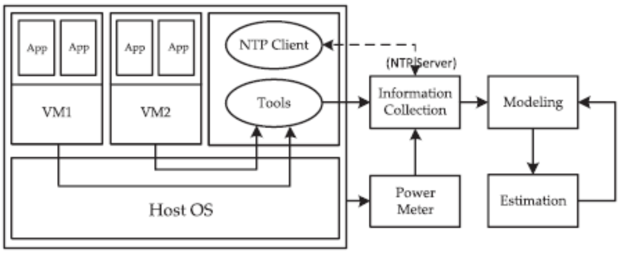

Fig 2.1: Architecture of VM Power Metering[3]

II. VM PLACEMENT MODEL USING BIN PACKING ALGORITHMS

As we know the quality of the IaaS layer in cloud computing can be evaluated by keeping consideration of both power consumption and quality of service (Qos), in this work, it was tried to put our focus on minimizing power consumption without making drastic alterations over the other areas, to meet the quality of IaaS. We will try to follow some heuristics for dynamic consolidation of VMs based on the previous data of resource usage. We have followed exact similar procedures for VM allocation to detect both underloaded and overloaded hosts and VM selections tools for selecting VMs which are needed to be migrated from those hosts as discussed in [5]. Now when it comes to VM placement, instead of using Best Fit Decreasing algorithm, we are going to propose some additional algorithms based on the solutions of bin packing problem which are likely to decrease the power consumption as well as maintaining quality of services. In the following section we have discussed very briefly about some popular solutions for bin packing problem.

- The First Fit (FF): FF starts with the most active bin and tries to pack every item in it before going into next bin. If no suitable bin is found for the item, then the next bin is selected to put in the new bin.
- First Fit Decreasing (FFD): In FFD the items are sorted in non-increasing order and then items are processed as the First Fit algorithm. It is actually the First Fit algorithm with the items are decreasingly sorted.
- Best Fit Decreasing (BFD): Like FFD, BFD also sorts items in non-increasing order and then for packing items it chooses a bin with minimum empty space to be left after the item is packed.

- Worst Fit Decreasing (WFD): it works exactly same as BFD except in one thing, instead of choosing bin with minimum empty space it chooses bin with maximum empty space to be left after the allocation of item on that bin.
- Second Worst Fit Decreasing (SWFD): Same as worst fit, it just choose bin with second minimum empty space. It is also known as almost worst fit decreasing (AWFD).

In this system the main emphasis is on to calculate the resources of network and node so that Service Level Agreement (SLA) may be maintained. For implementation of improved SPA resources are to be used in such a way that the task in the queue may be placed in a manner so that the load, delay and throughput i.e. QoS parameters may be achieved so that SLA may be maintained.

III. PROBLEM FORMULATION

Interesting models that tackle the problem of virtual machines placement are already proposed in the literature. However, to the best of our knowledge, no work tackles the problem of placing a full network as one package. In this context, placement models should provide for performance optimality for the whole parties involved in the placement process. However, to the best of knowledge, no work solves the problem of placing a full network as one package. In this context, placement models should provide for performance optimality for the whole system involved in the placement process as to check the network resources as well as node resources. Indeed, where is not a optimal solution to place a virtual machine in a way that fulfils its performance requirements but causing problems to others. Considering the placement costs is also a crucial factor in such decision, however, other factors like resource allocation to tasks and QoS guarantees are also important.

IV. FUZZY APPROACH IN WEB BASED QOS

Fuzzy set theory is a suitable system for modeling uncertainty arising from mental phenomena, which are neither random nor stochastic. In this paper, we use fuzzy inference system (FIS) to evaluate the cloud computing user’s satisfaction A fuzzy inference system is a rule based system with concepts and operations associated with fuzzy set theory and fuzzy logic. This system is a rule based system which is mapping input spaces to output spaces. Therefore, they allow constructing structures to be used to generate responses (outputs) by certain simulations (inputs) based on the stored knowledge of how the responses and simulations are related. The knowledge is stored in the form of a rule base, that is, a set of rules that express the relation between inputs of a system and expected outputs [30]. A “membership function” is a curve that defines how the value of fuzzy variable is mapped in a degree of membership between 0-1. In step one, membership functions are used to calculate the degree of fuzzy user’s satisfaction in different values expressed by linguistic term such as low, low to medium, medium, medium to high and high. IF-THEN expression is the most common way for representing human knowledge. This form generally is referred to as deductive form. It means that if we accept on a fact (premise, hypothesis, antecedent), then we can infer another fact called conclusion (consequent). The fuzzy inference system is a popular way for wide range of science and engineering. In step two, for making rules the verbal options of experts regarding the effects of different factors such as security, efficiency and performance, adaptability and cost are gathered and processed for generating a rule base and using them as inputs of our fuzzy inference system

V. FLOW CHART

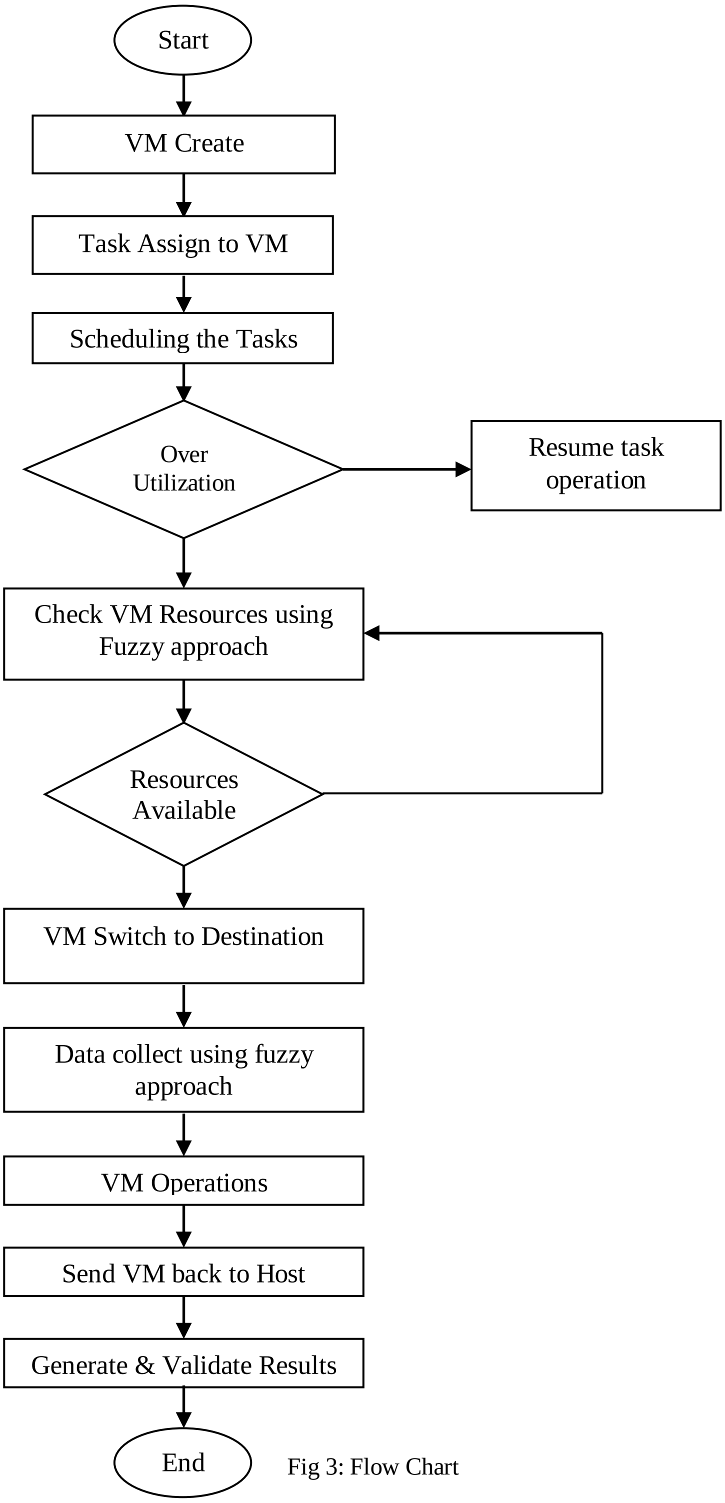

Fig 3: Flow Chart

Fig 3 is the description of the proposed approach in which various steps are defined about the proposed approach. As in the proposed approach first of all a virtual network is created on the physical resources. After the network creation the tasks are enqueued for processing on the virtual machine. To process the task list the resource vector is generated and check if the resources are available to process that task. If the resources are available on network as well as on the host and VMs then the tasks are executed on that machine using fuzzy approach.

Algorithm for Fuzzy based ADP:

Input: VMs, task

Output: Migration of VMs

Step 1: create VM
Step 2: queue ← task[i];
Step 3: Schedule task according to length;
Step 4: if resources available
Step 5: VM ←task
Step 6: if overutilization == true
Step 7: check resources of physical_machine[i] – physical_machine[current];
Step 8: if pm[j].resources > threshold
Step 9: transfer vm[current] to pm[j]
Step 10: apply loop transformation and loop fusion
Step 11: process task of vm[current];
Step 12: pm[current] ← vm[current]
Step 13: end

VI. RESULTS AND DISCUSSION

Make Span: Make span may be defined as the total time that is taken by the machine in which machine is in the idle state. In this proposed research the make span is lower i.e. 140 sec as compare to existing i.e. 160 sec.

Table 1: Comparative Study for Make Span (msec)

Technique / Host	Existing	Proposed
1	70	60
2	155	135
3	160	140
4	165	150
5	200	160

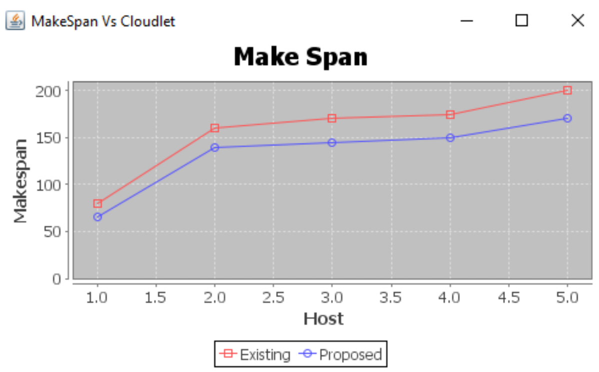

Fig 4: Make Span

Response Time: Response Time may be defined as the total time that is taken by the machine to respond to a task in cloud. In this proposed research the response time is lower i.e. 9sec as compare to existing i.e. 12 sec.

Table 2: Comparative Study for Response Time (sec)

Technique / Host	Existing	Proposed
1	10	7
2	11	7.5
3	12	8
4	12.5	8.5
5	13	10

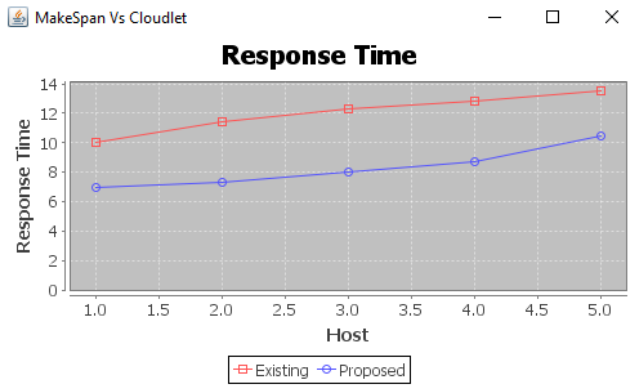

Fig 5 Response Time

Efficiency: As shown in fig 5.7 the efficiency of the proposed approach is better than that of existing one. In the proposed approach efficiency is 93% whereas in existing approach it is 83.4%.

Table 3: Comparative Study for Efficiency(%)

Technique / Host	Existing	Proposed
1	80	90
2	82.4	90.3
3	84.3	93
4	87.8	94.7
5	88.5	96.5

Fig 6: Efficiency

Comparison Table of existing and proposed approach:

Here the comparison takes place between the base paper and the work performed. The results produced by the work are better than the previous work done.

Table 4: Comparison table of existing and proposed approach

Approach	Response Time(sec)	Efficiency (%)	Make Span (msec)
Existing	12	83	160
Proposed	9	93.4	140

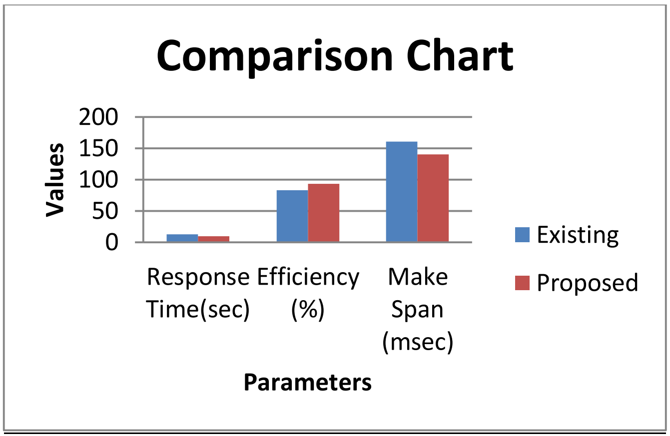

Fig 7: Comparison table of existing and proposed approach

Response Time is defined as the time at which the virtual machines stop executing. It includes transfer of the processor state. In the proposed approach, the Response Time is decreased which results in better performance.

VII. CONCLUSION

Appealing to the requirement of energy savings, many approaches of energy-efficient locating sensing have been explored. Methods beyond the action of locating are somehow auxiliary, and most of the attentions are focused on locating sensing based methods. A class of lightweight positioning systems has been developed to explore a large part of the energy-accuracy trade-off space. When image size of VM is implemented against the VM bandwidth then no significant effect is found on response time and it remains constant for which these parameters are investigated. But in case of Cloudlet long length versus Host bandwidth a pattern is observed in which response time increases in proportionate manner. In the proposed approach the Fuzzy approach is applied to maintain the load balancing and QoS parameters in the cloud. Using the modified approach the reduction in the Response Time of the various processes is achieved as shown in results.

In the future scope this proposed solution may be optimized to a single algorithm in which VM placement and task assignment on the VMs by reducing the operational cost and maintain the SLA may be achieved.

Cloud computing is a significant factor and here as a future course of work comparative different parameters can be examined against each other to discover the impact on protection and information security.

VIII. REFERENCES

[1]. Xiong Luo, Yixuan Lv, Ruixing Li and Yi Chen, “Web Service QoS Prediction Based on Adaptive Dynamic Programming Using Fuzzy NeuralNetworks for Cloud Services”, Special Section on Challenges For Smart Worlds, ISSN: 2169-3536, Vol: 3, 2015, pp: 2260-2269

[2]. Ahmad Nahar Quttoum, Mohannad Tomar, Bayan Khawaldeh, Rana Refai, Alaa Halawani and Ahmad Freej, “SPA: Smart Placement Approach for Cloud-service Datacenter Networks”, The 10th International Conference on Future Networks and Communications, Vol: 56, 2015, pp: 341-348

[3]. Chonglin Gu, Pengzhou Shi, Shuai Shi, Hejiao Huang and Xiaohua Jia, "A Tree Regression-Based Approach for VM Power Metering", Special Section On Big Data For Green Communications And Computing, IEEE, June 1, 2015

[4]. Zhaoning Zhang, Ziyang Li, Kui Wu, Dongsheng Li, Huiba Li, Yuxing Peng and Xicheng Lu, "VMThunder: Fast Provisioning of Large-Scale Virtual Machine Clusters", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 12, DECEMBER 2014

[5]. Jiaxin Li, Dongsheng Li, Yuming Ye and Xicheng Lu, "Efficient Multi-Tenant Virtual Machine Allocation in Cloud Data Centers", Tsinghua Science And Technology, Issn: 1007-0214, Volume 20, Number 1, February 2015, pp: 81-89

[6]. Gursharan Singh, Sunny Behal and Monal Taneja, "Advanced Memory Reusing Mechanism for Virtual Machines in Cloud Computing", 3rd International Conference on Recent Trends in Computing, Vol: 57, 2015, pp: 91-103

[7]. Narander Kumar and Swati Saxena, "Migration Performance of Cloud Applications- A Quantitative Analysis", International Conference on Advanced Computing Technologies and Applications, Vol. 45, 2015, pp: 823-831

[8]. Aarti Singh, Dimple Juneja and Manisha Malhotra, "Autonomous Agent Based Load Balancing Algorithm in Cloud Computing", International Conference on Advanced Computing Technologies and Applications, Vol: 45, 2015, pp: 832-841

[9]. S. Sohrabi and I. Moser, "The Effects of Hotspot Detection and Virtual Machine Migration Policies on Energy Consumption and Service Levels in the Cloud", ICCS, Vol: 51, 2015, pp: 2794-2798

[10]. Mohammad Mehedi Hassan and Atif Alamri, "Virtual Machine Resource Allocation for Multimedia Cloud: A Nash Bargaining Approach", International Symposium on Emerging Inter-networks, Communication and Mobility, Vol: 34, 2015, pp: 571-576

[11]. Christina Terese Josepha, Chandrasekaran K and Robin Cyriaca, "A Novel Family Genetic Approach for Virtual Machine Allocation", International Conference on Information and Communication Technologies, Vol: 46, 2015

[12]. Ankush P. Deshmukh and Prof. Kumarswamy Pamu “Applying Load Balancing: A Dynamic Approach” International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), vol. 2, issue 6, June 2012.

[13]. Alam, B. Doja, M.N.; Biswas, R. “Finding Time Quantum of Round Robin CPU Scheduling Algorithm Using Fuzzy Logic”, ICCEE 2008, International Conference, December 20-22, 2008.

[14]. Mohammed Rashid Chowdhury, Mohammad Raihan Mahmud, Rashedur M Rahman, "Study and Performance Analysis of Various VM Placement Strategies", IEEE SNPD, June 2015,

[15]. Chonglin Gu, Pengzhou Shi, Shuai Shi, Hejiao Huang, Xiaohua Jia, "A Tree Regression-Based Approach for VM Power Metering", Special Section On Big Data For Green Communications And Computing, IEEE, June 1, 2015

[16]. Zhaoning Zhang, Ziyang Li, Kui Wu, Dongsheng Li, Huiba Li, Yuxing Peng, Xicheng Lu, "VMThunder: Fast Provisioning of Large-Scale Virtual Machine Clusters", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Vol. 25, No. 12, DECEMBER 2014