International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 3, Issue 7, July 2014 **ISSN 2319 - 4847**

Machine Learning based decision making by brokers in cloud computing

Kiran Bala[1] and Sahil Vashist[2]

[1]CEC Landran(Pb)

[2]CEC, Landran(Pb)

ABSTRACT

Cloud Computing is in its naive stage as a lot of research is going to schedule the problems in best possible method. With scheduling the cost effectiveness is also looking up for the effectiveness of the system. As a result, in this paper, we have proposed a machine learning based methods to make the system proactive. This machine learning system may play an important role in scheduling the requests in an efficient manner. Simulation results show the effectiveness of the work.

Keywords: Proactive, machine learning, scheduling

1. INTRODUCTION

Cloud Computing provides the services to the consumer on the internet. In the course of cloud computing we can access anything that we want from ubiquitously to any computer without perturbing anything likes about their management, cost and storage, Infrastructure and so on. Cloud computing is delivering the services to the consumers through the Internet connection. As a replacement for maintaining data or updating applications on your personal hard drive according to your requirement and necessities, you can utilize this service via the Internet and access the data from any location and also use its applications whenever required. The cloud provides the services consent to individuals, enterprise and businesses utilize the infrastructure, storage, resources, hardware and software that are managed by providers from different geographical locations. Examples of cloud services like social sites and online file storage, Email clients like Yahoo, Gmail, Wikipedia, YouTube, Skype or Bit Torrent and many more. Cloud computing provides resources from a pool like servers, storage, services, processing power, network bandwidth and particular many user applications [1]. The cloud computing provides access to application, information, data and resources at any time from any geographical location, merely necessitate a connection of internet. This is implemented with virtualization technology where one or more servers can be configured and partitioned into many self-sufficient virtual servers and perform functionality independently and appearing to consumers as a single physical device and consumers are rented virtual machines (VMs) dynamically as they require. Cloud Computing helps production organizations, organizations, company, government institutions, academic organizations in cutting down operational costs. The following definition of cloud computing has been developed by the U.S. National Institute of Standards and Technology (NIST) [2] "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models"Cloud computing can be distinct as a distributed and parallel system with a set of virtualized and dependable computers that are provisioned animatedly [3] and easy to get as one or many resources accessed by conciliation among the provider of cloud and consumers based on service-level agreements. Various examples of the Cloud computing infrastructures/platforms are Microsoft Azure, Aneka, Amazon EC2 and Google App Engine and so on. National Institute of Standards and Technology (NIST) have explained cloud computing uniqueness that illustrate and distinguish the services of cloud from conformist approaches of cloud computing. The characteristics of cloud computing, consist of on-demand self service, measured service broad network access, rapid elasticity and resource pooling.On-demand self service means that customers and typically organizations can request the service provider for resources and also manage resources for computing. Broad network access, provides services to be accessible anywhere and anytime over the public networks or private networks.Pooled resources mean that consumers access number of resources from a cloud where number of resources are available, generally in distant data center. Pay per use measured service Cloud provider has different policies to produce a bill. According to the use of services are measured. Consumers are billed as per usage basis. Services can be measured as larger or smaller depend upon usage of consumer [4]. If we move towards scheduling the efficient strategies are still under research. A lot of strategies we have discussed in the next section. The sequence of the paper is that the next section 2 contains the related work, after related work we will go to our proposed framework under Section 3 and later we will see the experimental setup and results as section 4, further with a conclusion and future work as final section i.e. section 5

2. RELATED WORK

Talebi [5] proposed a consistent scheduling algorithm in a cloud computing environment. In this explains the PPDD and RSDC algorithm.In this scheduling algorithm the job is alienated in many sub jobs. Now the request time and acknowledge time of jobs is calculated independently in order to establish stability in the jobs. For each job the scheduling is completed by calculating the acknowledge time and request time from shared job. So the system efficiency is improved. Entezari-Maleki [6] proposed a RASA algorithm for scheduling the task. It is a combination of traditional two scheduling algorithms; Min-min and Max-min. RASA takes the benefits from both Min-min and Max-min algorithms and also help to remove their disadvantages. To attain this, Firstly RASA calculates the time of completion of the tasks according to available grid resources. After that, apply the both algorithms. For small task execution RASA uses the Min-min algorithm and for the tasks that is large, apply the Max-min algorithm for execution the large tasks so that avoid the delay of large tasks. RASA maintain the concurrency in the executions of both types of tasks large or small tasks. While the limit of each task, tasks arriving rate, communication cost, task execution cost of each of the resources that are not measured. The results defined that RASA is performing better in large level than existing scheduling algorithms. Guruprasad [7] has proposed a scheduling algorithm depend upon on Quality of service and priority system. In this scheduling algorithm serving the consumer requests according to priority that is assigned to each admitted queue. Calculating the tolerable delay and service cost of the requests and then added into the admission queue. This policy achieves high service completion time rate and guaranteed QoS with the planned architecture of cloud. Also improve the throughput and service response time of the request when it is in queue. This policy gives the uppermost preference for the users of extremely paid, in general the cost of other servicing in the also increases. Al-rahamawy [8] has proposed an algorithm depend upon the RASA algorithm. In this Max-min algorithm improved the execution time of the expected task based on selective basis instead of complete time of request. Petri nets describe the simultaneous performance of dispersed systems. Max-min algorithm gaining schedules better than original Max-min and RASA algorithm with similar lower makespan. Bansiwal [9] has proposed a scheduling differentiated algorithm in a cloud environment. The scheduling algorithm is used for reducing the load and improve the utilization of resources. The scheduling algorithms are static and dynamic. This algorithm is applied on static load balancing and uses non-preemptive priority queuing model for behavior estimation. In this algorithm one application is created. This application is a web application to do a number of actions. These actions are mainly file storing, uploading and downloading. For storing a file, if in one folder there is not sufficient space, then find out the folder which has less space and stored the file. By clicking the button, downloading any of file, text, audio and video. Uploading can also be done by clicking the button of uploading. For this there is requiring of the efficient scheduling algorithm. The QoS requirements and maximum profits provide by cloud providers are gained by users with this algorithm. Samir Youcef [10] has proposed complementary three bi-criteria approaches. These are cost-based, time-based and cost-time based. In cloud uses the two workflows, business and scientific workflow. In this algorithm use scientific workflow that explains with the help of the directed acyclic graph. In DAG, the nodes define the tasks and edges define the dependencies. The tasks mainly have NP-complete problems mean the problem of allocation and scheduling of tasks. So it is beneficial to use a heuristic algorithm than deterministic algorithm. The three approaches enhancing the overall the cost and execution time. This approach provides the consumers to choose the desired schedule and flexibility to estimate their preference for execution of tasks. HaijunCao [11] has proposed two major algorithms, dynamic-tuning algorithm based and evaluation-scheduling Algorithm. These algorithms are implemented in Hadoop. Hadoop is used by many companies like Microsoft, Facebook, etc. In the dynamic-tuning algorithm, the task is divided into map and reduce phase and map phase further divided into two phase map and sort. Reduce phase divided into three phases shuffle, sort and reduce. These phases record the values at every phase. Based on historical values of the map and reduce phase it map and reduce the tasks, estimate the progress of a task properly and also used for calculation of time for execution of tasks. Some time tasks may not get the resources due to load or other reasons and launch straggler task. In the evaluation-scheduling algorithm, before initiation of a straggler task on the node, it finds the idle node and assigns this task to the idle node. Late scheduler is used to find out the idle node and when, choose the task, then uses the largest left over time approach. So it reduces the resource wastage by finding the free slot. With this algorithm reduce the resource wastage and calculate the execution time. in-Guang Sang [12] has proposed algorithm UFPS (Urgency First Parallel Strategy) uses the local and cloud preferred strategy. In local strategy when the task arrives, then the task scheduler checks the resources at local whether resources are free or not. If resources are free at local then allocate the task at local server, otherwise check resources on cloud servers. My resources are free at the cloud then allocated the tasks at cloud. If resources are not found at both then put it on a tail of the queue. In cloud preferred strategy is opposite to local. Firstly check the cloud server and then locally. If idle resources are not found, then putting it in the queue. In UFPS algorithm uses both strategies. UFPA define the emergency of tasks. According to emergency the tasks put in front of the queue. The resources are pretreated and tasks are divided into a number of tasks after the tasks put into the buffer. Then these tasks are divided into many classes. The classes are divided into special kind require large computation and less computation. According to this special task put in one group and allocate to a special server, tasks that require larger computation put into another group and allocate for the server that have the ability of larger computation and similar smaller computation tasks put in one group and allocate to a smaller

server whose ability is smaller than others. After that, these tasks scheduled both at local and cloud cluster simultaneously. If the tasks find idle resources at local then allocate the resources at local cluster and will quit scheduling at cloud cluster. Similarly, if found in the cloud cluster, then will renounce scheduling at local cluster. If not, find idle resources at both then put it in a tail of the queue. UFPS enhance the efficiency, reduce time and economic increases.

3. PROPOSED FRAMEWORK

In the proposed framework we have used learning techniques to acquire the proactive behavior and we try to find the type of analysis to the systems which helps in finding the right mapping system (request-to-resource) as shown in figure 1. In this framework we have adopted three different approaches for decision making (proactive) also shown in:

3.1 One step or single learning

This system is used when a new type of request is generated from users. In the single learning concept, it first randomly puts the requests to any of the virtual machine for processing and notices the performance and output. In the next time when the same type of requests has generated, based on the performance of older one it is decided whether to go for the same or another.

3.2 Use of predefined rules

In order to refine the behavior of our learning system, a set of rules for the decision on the certain requests can be predefined and our algorithm has the capability to find out the can be used to figure out more specifically in comparable cases which may go against the rules. In this framework design certain type requests like priority requests, compute intensive requests, etc. can be granted by special privileges and govern under special rules.

3.3 Use of Multi learning

By feeding multi-learning into the cloud system the broker got the capability to find the intersection of all the single learning results. In multi-learning the broker tries to learn from all the results from that was previously mishandled during single learning. In this learning system the accuracy of the right decision is of high probability.

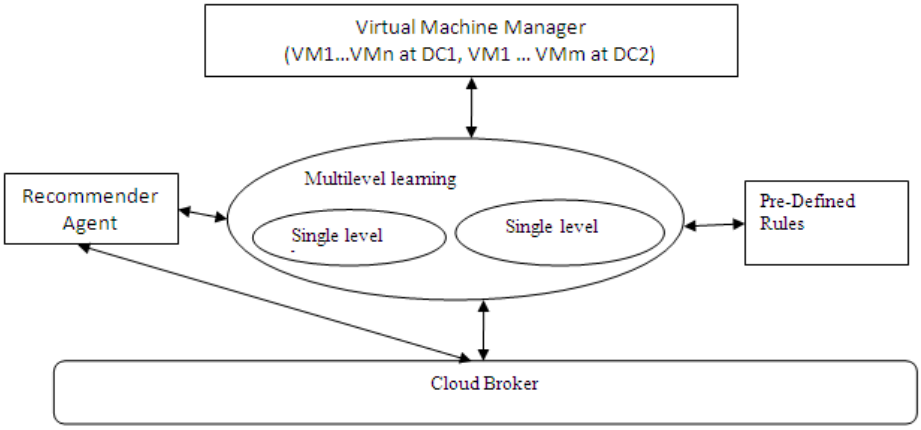

Figure 1: A descriptive view for proactive behavior in Cloud Computing

```
// Pseudocode : Proactive algorithm
A= getVMinfo(CPU,ram,bandwidth);
C = getcloudlets(fileSize);
For every request
D= Initialize_virtual_machine_manager(C);
Allotment_of_vms (D);
// initialization method for virtual machine manager
vm Initialize_virtual_machine_manager(Cloudlet val);
{
Process last _n _requests (get_n);
Call_recomendder_agent();
schedule_jobRequests();
return(vm)
}
```

In our approach

Traditional algorithm for scheduling, all the information regarding the job is already known and static, but now when faced with dynamic behavior then need to a proactive approach. Proactive approach studies the dynamic behavior of jobs and makes solutions that can attain a performance that is acceptable at execution time. Proactive behavior involves acting in advance of a future situation, rather than just reacting. In this consumer behavior is studied before, according to behavior of consumer algorithm create and execute. The consumer behavior is almost same. Consumer request for similar

application and require similar resources. So that proactive algorithm studies the behavior and attains the best performance. In this, there is a one datacenter, host, virtual machine created. When cloudlets given to a broker, broker schedule the request according to the algorithm. In the algorithm broker first get the information about virtual machines-virtual machine status, memory, storage etc. After that check the size of cloudlets. The broker gets the list of last processed cloudlets from Call_recomendder_agent and Call_recomendder_agent store the information of all last processed cloudlets and send information to the broker. The broker receives the list and then schedules the cloudless by schedule_jobrequest. After that, all cloudlets executed and produced the results.

4. EXPERIMENTAL SETUP AND RESULTS

In this section we use cloudsim as our simulating tool, the experimental setup we have used 30 virtual machines (VM1 to VM30) under 1 data center (DC1). Configuration of virtual machine as shown in Table 1.

Type	CPU cores	RAM	B/W
Small Instance	2	1 GB	1000 kbps

Further 20 user bases have been used to launch requests to the virtual machines. Since to cause the heterogeneity, we have taken the different priority levels of random selection clauses. These results are compared with the [13] as shown in figure 2 as per their success rate of scheduling. As shown in figure for initial requests PISA is working well as during this time PSA is learning the requests and putting them into database for future uses.

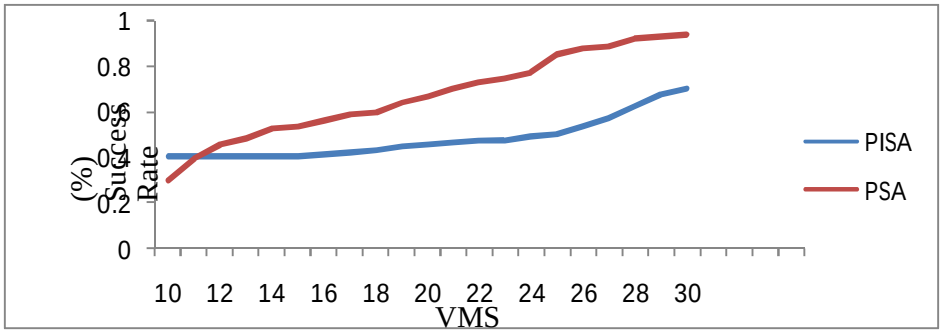

Figure 2: Success Rate PISA vs PSA

In the Figure 2 the comparison of both algorithms, from starting PISA algorithm performs better because this algorithm allocates the cloudlets according to priority and executes the cloudlets. PSA algorithm schedules the cloudlets according the randomly generated priority and use machine learning algorithm. By this at start point it allocates the cloudlets randomly to the virtual machine and then learns which virtual machine performs better. So after that allocate the cloudlets according to performance of virtual machines. So at starting PISA perform better, but the PSA success rate is higher than PISA.

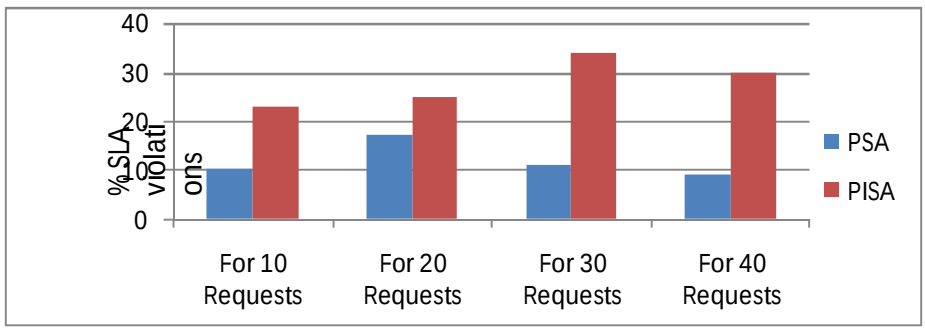

Figure 3: Percentage comparisons of SLA Violations

In the Figure 3 we can observe that there is an improvement of 20% when comparing both techniques and if we look at SLA violations comparisons for a sequence of runs we can see that the random priority does not make any impact on the working of PSA over PISA.

5. FUTURE WORK AND COMPARISONS

This paper has explored the machine learning technique for proactive behavior. In this work three types were used to exploit the past performance of the system. This designed is used to explore the predefined rules impact on the behavior, but in order to move further as future work we shall work on the higher the uncertainties of the users which need to be investigated.

REFERENCES

[1.] Mohamed Abu Sharkh, Manar Jammal, Abdallah Shami, and Abdelkader Ouda "Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges" in IEEE Communications Magazine November 2013
[2.] 2.http://thecloudtutorial.com/related.html
[3.] Linan Zhu1, Qingshui Li2 and Lingna He3 **"**Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm" in IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, September 2012 ISSN (Online): 1694-0814
[4.] 4.Ronnie D. Caytiles and Byungjoo Park " A Study on Analysis and Implementation of a Cloud Computing Framework for Multimedia Convergence Services" in International Journal of Software Engineering and Its Applications Vol. 7, No. 2, March, 2013.
[5.] Arash Ghorbannia Delavar,Mahdi Javanmard , Mehrdad Barzegar Shabestari and Marjan Khosravi Talebi "RSDC (RELIABLE SCHEDULING DISTRIBUTED IN CLOUD COMPUTING)" in International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.2, No.3, June 2012
[6.] Saeed Parsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment" in World Applied Sciences Journal 7 (Special Issue of Computer & IT): 152-160, 2009.Berry M. W., Dumais S. T., O'Brien G. W. Using linear algebra for intelligent information retrieval, SIAM Review, 1995, 37, pp. 573-595.
[7.] Dr. M. Dakshayini, Dr. H. S. Guruprasad "An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment" International Journal of Computer Applications (0975 – 8887) Volume 32– No.9, October 2011Shamsollah Ghanbari, Mohamed Othman "A Priority based Job Scheduling Algorithm in Cloud Computing" International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012)
[8.] El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy "Extended Max-Min Scheduling Using Petri Net and Load Balancing" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012
[9.] Shalmali Ambike, Dipti Bhansali, Jaee Kshirsagar, Juhi Bansiwal " An Optimistic Differentiated Job Scheduling System for Cloud Computing" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue
[10.] 2, Mar-Apr 2012, pp.1212-1214
[11.] Kahina Bessai, Samir Youcef, Ammar Oulamara, Claude Godart and Selmin Nurcan" Bi-criteria workflow tasks allocation and scheduling in Cloud computing environments" , Fifth International Conference on Cloud Computing, 2012, IEEE.
[12.] Xu Zhao, Xiaoshe Dong, Haijun Cao, Yuanquan Fan, Huo Zhu " A parameter dynamic-tuning scheduling algorithm based on history in heterogeneous environments" Seventh ChinaGrid Annual Conference, 2012, IEEE.
[13.] Zheng-Wu Yuan ,Xin-Guang Sang "A Study on Resource Scheduling Strategy in the
[14.] Enterprise Service Cloud" International Conference on Systems and Informatics (ICSAI 2012), 2012
[15.] 13.Hu Wu, Zhuo Tang, Renfa Li "A Priority Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing" in School of Computer and Communication of Hunan University, Embedded Systems & Networking Laboratory, China, ICACT2012.