

Load Balancing Using Genetic Algorithm in Mobile Cloud Computing

Neha Gupta

*Department of Information and Technology
Chandigarh Engineering College, Mohali, Punjab, India*

Parminder Singh

*Department of Information and Technology
Chandigarh Engineering College, Mohali, Punjab, India*

Abstract- Mobile Cloud Computing is a fast developing technology today that faces the dominant problem of load imbalance due to the high demand of mobile applications. There are lots of techniques available to solve the problem but the load balancing performance can be improved by using more optimized solution. This paper proposes a load balancing scheme based on Genetic Algorithm (GA). The algorithm advances to balance the load of the mobile cloud infrastructure while trying to minimize the processing time or responsiveness of tasks with reduced number of migrations of virtual machines and improving the resource utilization by dividing the computing capacity of a datacenter into n number of virtual machines executing the number of requests at the same time and thereby improving the performance. The proposed load balancing scheme has been implemented using the cloudsim simulator. The simulation results show the efficiency and effectiveness of the proposed algorithm.

Keywords – Mobile Cloud Computing, Load balancing, Virtual Machine, Cloudlets.

I. INTRODUCTION

Cloud Computing has been involved at a very large scale in the outsourcing of computing resources of IT infrastructures, service platforms, and software in the last few years. With the emergence of ultra-fast 3G and 4G mobile networks and highly-featured Smartphone's and tablets, the prerequisites are now met for bringing cloud computing to the mobile domain. An explosion of mobile and handheld devices is also extensively contributing to world IP data traffic. To support such data demand cloud computing because of its rapid scalability, omnipresent network access, on-demand self-service and other features, seems to be the right choice. The mobile devices are constrained by their processing power, storage and some other features. However, cloud computing addresses this issue and provides them an illusion of infinite computing resources. The cloud computing comes at this point and improved the storage capacity and processing power. Cloud Computing provides them an illusion of infinite computing resources [6].

Mobile cloud computing combining the mobile devices and cloud computing is a new platform to create a new infrastructure, whereby cloud performs the profound lifting of computing-intensive tasks and storing enormous amounts of data. In this new architecture of Mobile Cloud Computing, data processing and data storage happen external of the mobile devices. MCC provides various advantages such as improvement in data storage capacity and processing power, improved synchronization of data because they are gathered and stored in one place, having policy to access from anywhere, improved reliability and ease of integration which could be possible due to the underlying enabling technologies HTML5, CSS3, broker or hypervisor for mobile devices and cloudlets that force adoption of mobile cloud computing. Mobile Cloud Computing as the combination of cloud computing and mobile networks is used to bring benefits for mobile users, network operators, as well as cloud computing providers that leverages varied cloud resources and network technologies towards unhampered functionality, storage, and mobility to provide a multitude of applications on mobile devices anywhere, on the pay-as-you-use principle anytime through the channel of internet. The ultimate goal of Mobile Cloud Computing is to enable execution of rich mobile applications on a plethora of mobile devices, with an emphasis on rich user experience.

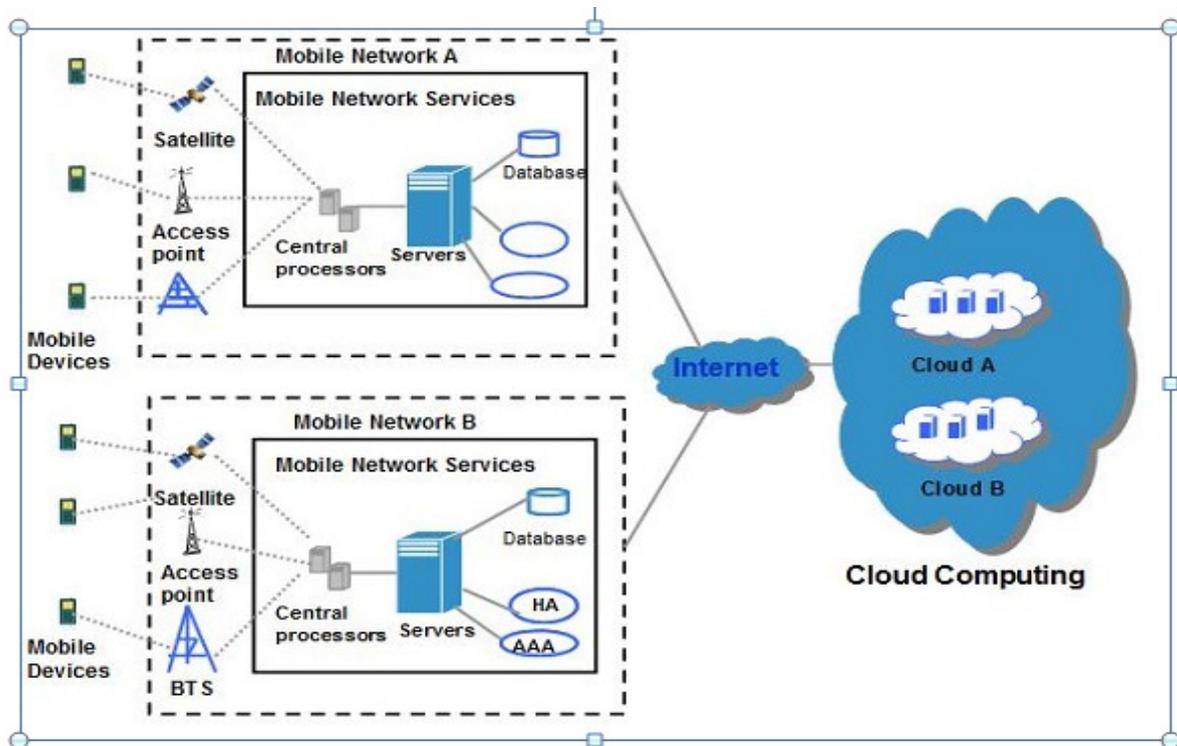


Fig 1: Architecture of Mobile Cloud Computing [15]

Cloud Computing though widely been adopted by the IT industry, yet there are many existing issues like Load Balancing, Server Consolidation, Energy Management, Security etc that are not fully addressed. Fundamental to these issues is the issue of load balancing that is a method to uniformly distribute user requests to the virtual machines in the cloud to achieve high resource utilization ratio and user satisfaction that in turn led to the benefits to both the cloud providers and users. The similar challenge is addressed by Mobile Cloud Computing also where several and variable computing requirement comes on datacenters. Typical applications of Mobile Cloud Computing include the Google's Gmail and Google Voice for smart phone's that are two well known mobile cloud apps. In Mobile cloud computing infrastructure both the data storage and data processing takes place outside of the mobile device. A cloud based mobile application is comparable to an application purchased or downloaded from a mobile application store where the processing power is driven, from the cloud, not from the handheld device. But it is important to ensure in a web based application that the backend web server will not breakdown due to overloading. At this position the selection of load balancing strategies is essential. This also relates to the issue of VM migration.

Our objective is to address this issue. But in our approach we do not consider the suspension of state of task and execute all the user requests at a same time by dividing the computing capacity of the datacenter into number of virtual machines[1] preventing the server from downtime thereby balancing the load and that too with reduced processing time to achieve better performance.

The rest of the paper is organized as follows: Section II includes the Related Work. Section III, describes the proposed algorithm. Section IV shows the experimental setup of the experiment and says about the working environment. Section V shows the result and comparison analysis.

II. RELATED WORK

Cloud Computing is fast emerging as an advanced technology in IT industries providing the users for accessing the shared pool of distributed resources .When the demand on the resources increases the level, the problem of load imbalance arises leading towards opening of several research avenues in the domain.

Xiaocheng Lui et al[1] had implemented Conservative Migration and Consolidation supported Backfilling CMCBF to increase the utilization of resources in the data-centers when two or more jobs are running in parallel. Meanwhile the algorithm tries to use the computing capacity of the data center nodes running parallel processes with low resource utilization to improve the performance of the job scheduling. The algorithm uses virtualization technologies to divide the computing capacity of each node into two levels i.e foreground virtual machine and the background virtual machine. It ensures that a job is dispatched to run in foreground VMs whenever the number of required VMs is free or occupied by jobs arriving later than it satisfies its node requirement or they can also be dispatched to background VMs to run if the VM requirement is satisfied or else their state is suspended to run later when the required number of VMs are free. It also allows jobs to run in background VMs simultaneously with those foreground VMs to improve node utilization. The paper had also shown the comparison of results of CMCBF with some other algorithms and shows that the algorithm outperformed the others.

Mudassar Ahmad et al [2] had implemented a prognostic load balancing strategy named Amplified ESBWLC for computational latency reduction in Mobile Cloud Computing based on Simple Exponential Smoothing Forecast Technique. In this simple exponential smoothing method a “smoothing parameter” or “smoothing constant” is used to determine the weights assigned to the observations. This experimental results shows that the selection of data centers leads to minimization of load on data centers and reduction in latency felt by users and helps in minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks.

Rashmi.K.S et al.[3] had implemented an Enhanced Load Balancing Approach to avoid deadlocks among the Virtual Machines (VMs) in the cloud while processing the requests received from the users by VM migration. The algorithm avoids the deadlock by providing the resources on demand resulting in increased number of job executions. The algorithm by avoiding the deadlock enhances the number of jobs to be serviced by cloud service provider and thereby improving the working performance of the cloud service provider by reducing the rejection in the number of jobs submitted. The algorithm is implemented using the hop time and wait time. Hop time is the duration involved in migration of the job from the overloaded VM to the underutilized VM for providing the service. Wait time is the time after which the VMs become available to service the request.

Entezari-Maleki et al [4] had implemented a task scheduling algorithm RASA. It is composed of two traditional scheduling algorithms; Max-min and Min-min. The algorithm builds a matrix of completion time of the task on the resource. If the number of available resources is, odd, the Min-min strategy is applied to assign the task, otherwise the Max-min strategy is applied. The remaining tasks are assigned to their appropriate resources by one of the two strategies alternatively. If the number of available resources is odd it is preferred to apply the Min-min strategy first in the first round otherwise is better to apply the max-min strategy the first. Alternative exchange of the two strategies results in consecutive execution of a small and a large task on different resources and hereby, the waiting time of the small tasks in Max-min algorithm and the waiting time of the large tasks in Min-min algorithm are ignored.

III. PROPOSED ALGORITHM

In the proposed framework we had used genetic based techniques to balance the load of the system and to schedule the request to different virtual machines by finding the right mapping solution and that also with reduced response time of execution of requests. A genetic algorithm is a type of searching algorithm. It searches a solution space for an optimal solution to a problem. The key characteristic of the genetic algorithm is how the searching is done. The algorithm creates a “population” of possible solutions to the problem and lets them “evolve” over multiple generations to find better and better solutions. The algorithm is composed of four functions:

1. Population Size: In this first of all two array lists are created one is for containing the list of virtual machines over which tasks are to be scheduled to execute and other is for containing the list of requests to be processed.
2. Objective Function: This function will fetch all the possible virtual machine over which task is to be executed.
3. Mutation Function: This function will schedule the task to the resource queue according to the calculation required by the tasks and the ability of resources to process the task or we can say it can find the best virtual machine for task to be executed.
4. Fitness Function: This function will call the virtual machine that satisfies the condition by performing the above function when the task is to done.

Methodology of the work:

- Begin
- Resources of virtual resource pool are pretreated.
- The reached tasks are put into buffer forming a set.
- Initialize workload by random solutions
- Evaluate each requests in terms of workload and cost
- Do
- Recombine pairs of virtual machines and requests
- N tasks are divided into M classes, the special kind of tasks are put together and make up a set, the tasks of large amount of calculation are put together and make up a set and the tasks of small amount of calculation are put together and makeup a set, it has M sets.
- Choose a task in each queue head; there are M tasks in total.
- Mutate the resulting offspring and add new vm to free pool if required.
- M tasks are scheduled to virtual machines at the same time, the tasks of large amount of calculation are scheduled to resources queue whose calculation ability are strong., the tasks of small amount of calculation are scheduled to resources queue whose calculation ability are weak.
- Select individuals for next generation

IV. EXPERIMENT AND RESULT

The working environment for cloud computing where the proposed algorithm was implemented was done using cloud sim-2.0 as a framework in the simulator environment. The simulation was performed on core 3 processor with 1GB Ram and disk space performed on OS 7 platform. The whole experiment was written in java language on NetBeans IDE. This work considered various Datacenter, Virtual Machines (VM), host and cloudlet components from cloudsim for execution analysis of the algorithms. This research work considered different datacenters located at different regions comprises of one host per datacenter. The simulation also comprised of different requests by the user to be processed. There are in total 18 cloudlets or requests in total which are ranging from 300 MB to 23000 MB. Table 1 shows the configuration of data centers which consist of virtual machines with their configurations as:

Table 1: Virtual Machine Configuration

VM (with ID: 0)	VM 2(with ID:1)
Mips = 1000 Size = 10000 MB Ram = 512 MB (vm memory) Bandwidth = 1000 kbps Pes Number = 1(number of cpu's)	Mips = 5000 Size = 10000 MB Ram = 512 MB (vm memory) Bandwidth=1000kbps Pes Number = 1(number of cpu's)

The algorithm was implemented using cloudsim simulation. The simulation runs and results were compared with the CMCBF as shown in figure 1 as per the average response time of the requests. As shown in figure our algorithm significantly outperforms CMBF on response time. In CMCBF if the idle nodes were not available for the incoming job, then it saves the state of the job and restores it later when the nodes required by it becomes free. But this is not the case in genetic algorithm and it saves the time that was utilized during the suspension and restoration process.

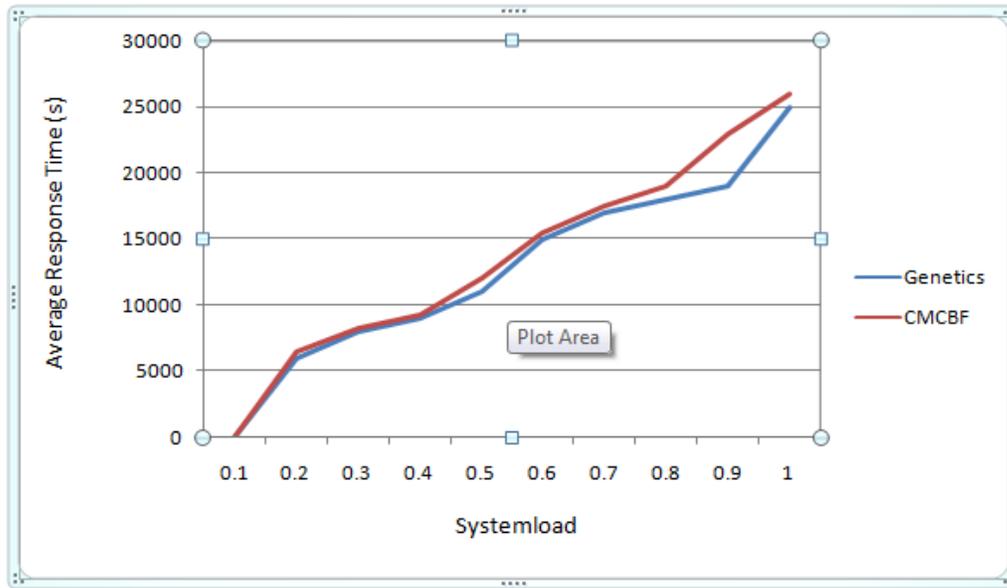


Figure 2: Average response time Genetic based vs CMCBF

The average number of job migrations in Genetic based algorithm are more as compared to that in CMCBF. This is due to the availability of large computing capacity of the VMs that allows various requests to be processed without incurring number of job migrations. In the CMCBF large number of jobs triggers many backfilling operations, thus increases the chance of job migrations.

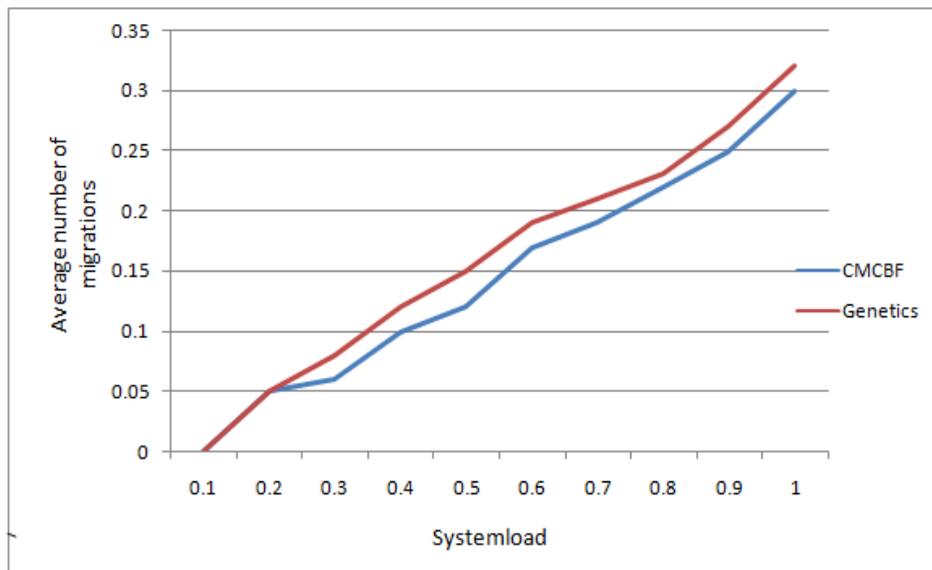


Figure 3: Average number of migrations in Genetic based vs CMCBF

V.CONCLUSION

Mobile cloud computing by combining the mobile devices and cloud computing is a new platform to provide a new infrastructure, whereby cloud operates the lifting of computing-intensive jobs and placing enormous amounts of data. All the data processing and data storage is done in the cloud external of the mobile devices. As an increasing number of mobile applications leverage the computing power of the cloud, it becomes important to efficiently manage computing resources for these applications for improving the performance. Performance can be improved

by increasing the responsiveness of the jobs. So scheduling jobs for both efficient resource use and job responsiveness is challenging.

We have designed and tested an algorithm which is made by Genetic Algorithm. The main goal of it is to schedule multiple jobs on multiple machines in an efficient manner such that the jobs take the minimum time for the completion and increasing the responsiveness of jobs. The tasks of large amount of calculation are scheduled to resources queue whose calculation ability are strong and the tasks of small amount of calculation are scheduled to resources queue whose calculation ability are weak. The performance analysis produced expected results and thus proved the proposed approach is efficient in optimizing schedule by balancing the loads.

REFERENCES

- [1] Amandeep, "Analysis of Load Balancing Techniques in Cloud Computing", International Journal of Computers & Technology, vol. 4, no.2, pp.2277-3061, March-April, 2013.
- [2] B.Subramani, "A New Approach For Load Balancing In Cloud Computing", IEEE ,vol.2, pp. 1636-16405, May 2013.
- [3] T.EpoMofolo, R.Suchithra, "Heuristic Based Resource Allocation Using Virtual Machine Migration: A Cloud Computing Perspective" International Refereed Journal of Engineering and Science (IRJES), vol. 2, pp.40-45, May 2013.
- [4] Weizhe Zhang, Hui He, Gui Chen and Jilong Sun, "Multiple Virtual Machines Resource Scheduling for Cloud Computing" Applied Mathematics and Information Sciences 7, pp.2089-2096, 2013.
- [5] Xiaocheng Lui, Cheng Wang, Juliang Chen and Albert Y.Zomaya, "Priority-Based Consolidation of Parallel Workloads in the Cloud", IEEE Transactions On Parallel And Distributed Systems, vol.24, no.9, pp.1874-1882, September, 2013.
- [6] Shalmali Ambike, Dipti Bhansali, Jae Kshirsagar, Juhi Bansiwai " An Optimistic Differentiated Job Scheduling System for Cloud Computing", International Journal of Engineering Research and Applications (IJERA) ,vol. 2, pp.1212-1214, Mar-Apr 2012.
- [7] Luiz F.Bittencourt, Edmundo R.M.Madeira, and Nelson L.S da Fonseca, "Scheduling in Hybrid Clouds", IEEE Communications Magazine, vol.4, pp.42-47, September, 2012.
- [8] Linan Zhu, Qingshui Li and Lingna He "Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm", International Journal of Computer Science Issues (IJCSI), vol. 9, no.2, pp.1694-0814, September 2012.
- [9] Luiz F.Bittencourt, Edmundo R.M.Madeira and Nelson L.S da Fonseca, "Scheduling in Hybrid Clouds" IEEE Communications Magazine, pp.42-47, September, 2012.
- [10] Linan Zhu, Qingshui Li and Lingna, "Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm" International Journal of Computer Science Issues (IJCSI), vol. 9, no.2, pp.1694-0814, September 2012.
- [11] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose and Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software-Practice and Experience, vol.41, pp.23-50, January 2010.
- [12] R.Buyya, C.Yeo, Venugopal, S.Broberg, I.Brandic, "Cloud computing and emerging Platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, vol.25, pp.599-616, June 2009.
- [13] Saeed Parsa and Reza Entezari-Maleki, "RASA: A New Task Scheduling Algorithm in Grid Environment" World Applied Sciences Journal, vol.2, no.7, pp.152-160, 2009.
- [14] Mladen A.Vouk, "Cloud computing-issues, research and implementations", Journal of Computing and Information Technology, vol. 8, no. 6, pp.235-246, April 2008.
- [15] Mudassar Ahmad, "Prognostic Load Balancing Strategy for Latency Reduction in Mobile Cloud Computing", Middle-East Journal of Scientific Research, vol.6, pp.805-813, 1990.