Volume: 6 Issue: VIII Month of publication: August 2018

DOI:

DE-GA based Model for Load Balancing in Cloud Computing

Yashika Sharma[1], Anshu Sharma[2]
[1]Research Scholar, [2]Assistant Professor, RPIIT Karnal   Kurtukshetra University.

Abstract: In this research, we have considered the above challenges and summarized the goals of our work for the thesis To study various existing load balancing techniques used for cloud computing environment. To develop a heuristic algorithm to improve performance for load balancing in cloud computing. To propose a new adaptive and efficient method for load balancing using some Meta Heuristic approach in cloud. To validate and verify the performance of above mentioned algorithms. Traditional way for task scheduling in cloud computing tended to use the direct tasks of users as the overhead application base. There are some problems with it. These problems leads to over-costed and over-priced in some high volume simple task while under-costed and under-priced in low-volume complex ones.

Index Terms: Cloud Sim, EBS, Load, GFS

I. INTRODUCTION

Cloud Computing is a catchword of 2010 and numerous specialist's distress on its precise explanation. But the best used one and corresponded one comprises the idea of web-based facilities which are obtainable on request from and improved and extremely ascendable facility supplier. Meanwhile such a difference on the description, one will be providing to improved comprehend of the idea. The cloud is IT as a provision, brought by IT capitals that are self-governing of position. It is a flair of computing in which energetically ascendable and frequently virtualized resources are delivered as a facility concluded the Internet where end-consumers devise no information of, proficiency in, or regulator above the expertise infrastructure (the cloud) that provisions them [1]. Cloud Computing mentions to the transfer of resources concluded the Internet. In its place of consuming your own hard drive or apprising requests for your requirements, you customs service-over-a network, at another position, to accumulate your info or custom its requests [2]. Doing so provides growth to positive confidentiality associations. Once you stock your photographs virtual, in its place of your computer, or usage of web mail or a social networking establish, you are expending "Cloud Computing" service. Cloud Computing is nothing but a fusion of virtualization and distributed computing; where immensely scalable IT related capabilities are provided to the multiple external customers "as a service" using internet technologies[11].

Cloud computing resulted from the convergence of Grid computing technology18]. In the middle of 1990's, to satisfy the rising demand such as bandwidth, storage, speed, capacity, and resources cloud computing is used. The greatest challenge in cloud environment is resource management and optimal utilization and also uses different management policies as well[10]. The development of cloud computing is closely related to the progress of the Internet. In early 2003, Google released GFS, BigTable, MapReduce and other technologies. A number of companies and research institutions expressed great interests in this framework. Then, Apache Foundation launched an open source distributed computing project: Hadoop. It is a framework that enables the distributed processing of large data sets across computer clusters. It achieves a similar function as GFS, Map/Reduce, BigTable and Chubby. In March 2006, Amazon published the Simple Storage Service (Simple Storage Services, S3) which is a service using the SOAP protocol for users to store and access their own data objects. For the service providers, they offered a reliable and extensible storage service to users who use it as an online storage solution case. In July 2007, Amazon launched a Simple Queue Service that allows the virtual hosts to send messages to each other. It also support data transfer between distributed applications with no need to consider the message loss problem. Recently, Amazon starts to provide EBS (Elastic Block Store) service to users with block-level storage interfaces.[4]

II. SCHEDULING

There has been various types of scheduling algorithm exist in distributed computing system. Most of them can be applied in the cloud environment with suitable verifications. The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments.

According to a simple classification, job scheduling algorithms in cloud computing can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, Jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time. The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min–Min algorithm and Max–Min algorithm. By On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system.

Since the cloud environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment. Most fit task scheduling algorithm (MFTF) is suitable example of On-line mode heuristic scheduling algorithm.

A. First Come First Serve Algorithm

Job in the queue which come first is served. This algorithm is simple and fast.

B. Min–Min Algorithm

This algorithm chooses small tasks to be executed firstly, which in turn large task delays for long time.

C. Max – Min Algorithm

This algorithm chooses large tasks to be executed firstly, which in turn small task delays for long time.

D. Most fit Task Scheduling Algorithm

In this algorithm task which fit best in queue are executed first. This algorithm has high failure ratio.

E. Priority Scheduling Algorithm

The basic idea is straightforward: each process is assigned a priority, and priority is allowed to run. Equal-Priority processes are scheduled in FCFS order. The shortest-Job-First (SJF) algorithm is a special case of general priority scheduling algorithm. An SJF algorithm is simply a priority algorithm where the priority is the inverse of the (predicted) next CPU burst. That is, the longer the CPU burst, the lower the priority and vice versa. Priority can be defined either internally or externally. Internally defined priorities use some measurable quantities or qualities to compute priority of a process.

1) *Round Robin Algorithm[7]* In the round robin scheduling, processes are dispatched in a FIFO manner but aregiven a limited amount of CPU time called a time-slice or a quantum. If a process does not complete before its CPU-time expires, the CPU is preempted and given to the next process waiting in a queue. The preempted process is then placed at the back of the ready list. Though the algorithm is very simple, there is an additional load on the scheduler to decide the size of quantum and it has longer average waiting time and low throughput. The Round Robin algorithm mainly focuses on distributing the load equally to all the resource. Using this algorithm, the scheduler allocates one VM to a node in a cyclic manner. The round robin scheduling in the cloud is very similar to process scheduling. Then the scheduler starts with a task and moves on to the next task, after a VM is assigned to that task. Restate until all the nodes have been allocated at least one VM and then the scheduler returns to the first task again. Hence, in this case, the scheduler does not wait for the exhaustion of the resources of a node before moving on to the next task. As an example, if there are three job and three VMs are to be scheduled, each task would be allotted one VM, provided all the nodes have enough available resources to run the VMs.

Index-> (index+1) mod NoVM (Eq. 1)

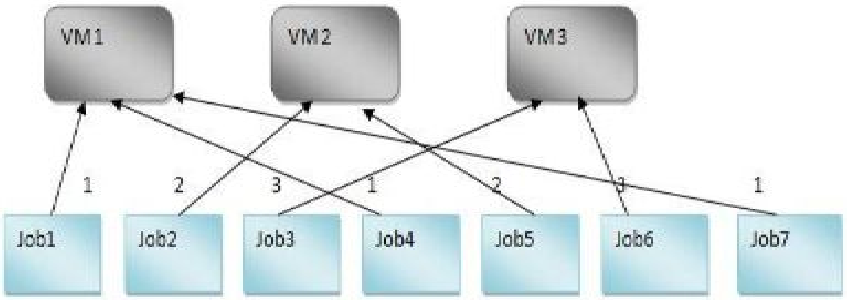


Figure 2.1 : Round Robin Algorithm

F. Opportunistic Load Balancing Algorithm[47]

In this algorithm attempts to dispatch the selected job to the available VMs which has the lowest load compared to the other VMs. The idea is to scale the current loads for each VM before sending the job. Then, the VM that has the minimum load is selected to run the job. Assigns a task to the machine that becomes available next, without considering that the execution time of the task on that machine. when multiple machines become available at the same time, then one is arbitrarily selected.

Index ->Min{v.getready()|v VML (eq. 4)

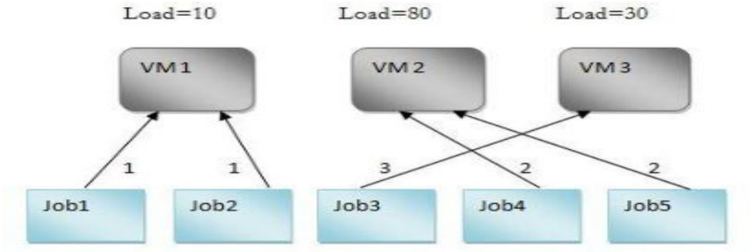


Figure 2.2 .Oppurtunistic load balancing algorithm

III. RELATED WORK

In [2]we deal with the various methodologies adopted to handle all the processes and jobs concurrently executing and waiting into the web application and web server

housed into the same system or different systems. Various issues like virtual resources, queuing strategies, resource managers etc. we has been discussed here apart from the main coverage points. Web application remains in contact with the user to work inreal-time with the user's requests to avoid any kind of delay. ame system or different ones the fact that the level of performance and efficienc level differs by a value, not so large, but, significant enough. Each microsecond of resource' usage counts a lot in terms of CPU performance and efficiency. h the application and they demand for the subsequent number of resources as needed by them and make sure that they are used in the most appropriate manner. Web application remains in contact with the user to work in real-time with the user's requests to avoid any kind of delay. An application can have any number of processes [4] under it which also depends on the ongoing processes. Different processes are kept in different process groups [4] to a certain order of uniformity in the scheduling [4] process and therefore, are allotted their respective Process Group ID. Every process can contribute to any number of jobs depending upon the type and number of operations to be performed by that particular instance of process. Jobs are assigned a particular ID. a detailed working of different job scheduling methods behind the web application on a web server on the same system or on different systems in a virtual cloud computing environment [1]. The various terminologies and concepts were thoroughly looked and thus explained. more future findings regarding the scheduling techniques in a cloud environment. More efficient and faster ways to schedule jobs and increase CPU throughput [2, 4] needs to be discovered. Also, this will fuel the greater knowledge and popularity of cloud environment.

In [4]presents a novel heuristic scheduling algorithm, called hyper-heuristic scheduling algorithm (HHSA), to find better scheduling solutions for cloud computing systems. The diversity detection and improvement detection operators are employed by the proposed algorithm to dynamically determine which low-level heuristic is to be used in finding better candidate solutions. To evaluate the performance of the proposed method, this study compares the proposed method with several state-of-the-art scheduling algorithms, by having all of them implemented on CloudSim (a simulator) and Hadoop (a real system). The results show that HHSA can significantly reduce the makes pan of task scheduling compared with the other scheduling algorithms evaluated in this paper, on both CloudSim and Hadoop.

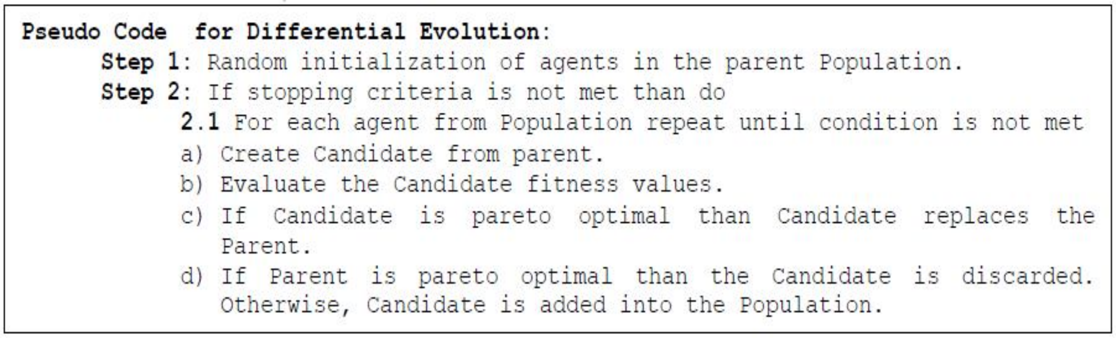
In [6] proposal the use of two basic swarm intelligent metaheuristic algorithms, including ant colony optimization and priority based Artificial Bee Colony (ABC). The main objective was to combine some metaheuristic properties of ants and honey bees to prepare a hybrid algorithm for effective scheduling and uniform distribution of workload among the cloud virtual resources. The proposed algorithm was decentralized to avoid the single point of failure. The parameters used were number of processing cores, MIPS searching, memory size of virtual machine and virtual machine bandwidth. In the implemented algorithm, ants were considered as cloudlets that are composed of userbases and food as virtual machine. Cloud Analyst toolkit was used for simulation and performance analysis. The results were derived in the form of statistical metrics. The simulation results demonstrated that the proposed approach outperforms the previous approaches like Round-Robin, Equally Spread Current Execution Load (ESCE) and metaheuristic ACO.

[7]performed the comparative analysis of four load balancing policies/strategies including RR, Throttled algorithms were implemented using ESCE and FCES. All four algorithms were implemented using Cloud analyst simulation tool. The parameters used for comparative analysis were: average response time, data center request servicing time and total cost of data center. The performance of Round Robin algorithm was found to be the best among all.

IV. PROPOSED MODEL

A. Differential Evolution

One inconceivably fit count from formative figuring due to its awesome connection properties and few control parameters is Differential Evolution (DE). Differential Evolution (DE) is a capable change approach. Storn and Prince in 1995 is the pioneer of Differential development check. Differential Evolution is a key yet astoundingly persuading streamlining approach in understanding a blueprint of errand booking issues with some veritable applications. Differential development joined with progress approachs (ES"s) and transformative programming (EP) can together be made into a class out of people – based, with Derivative free systems known as Evolutionary count. Every single one of these frameworks copy Darwinian development and build up a masses of individuals beginning with one age then onto the join age by like transformative operational parts, for instance, change, cross breed and decision.

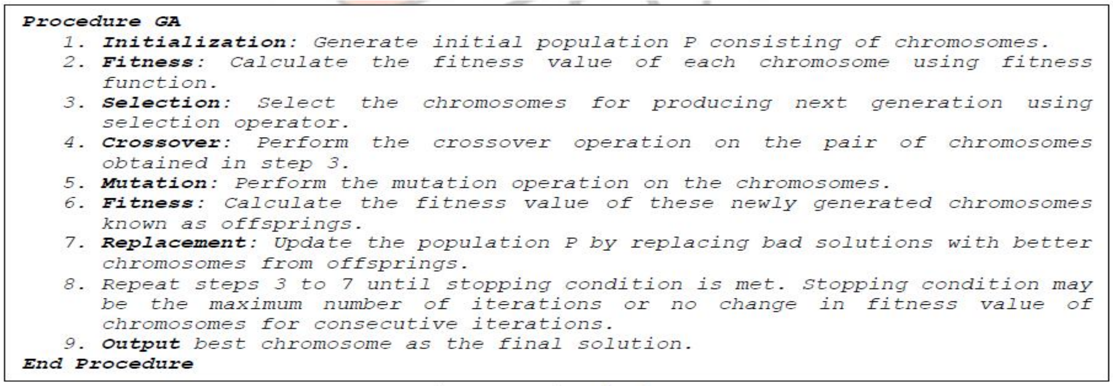


```
Pseudo Code  for Differential Evolution:
      Step 1: Random initialization of agents in the parent Population.
      Step 2: If stopping criteria is not met than do
            2.1 For each agent from Population repeat until condition is not met
            a) Create Candidate from parent.
            b) Evaluate the Candidate fitness values.
            c) If Candidate is pareto optimal than Candidate replaces the
               Parent.
            d) If Parent is pareto optimal than the Candidate is discarded.
               Otherwise, Candidate is added into the Population.
```

Fig.4.1 Pseudo code of DE approach

B. Genetic Algorithm

GA was first exhibited by Holland in 1975 and addresses a people make streamlining structure arranged in light of a moral story of the movement procedure found in nature. In GA, each chromosome (individual in the all-inclusive community) addresses a possible response for an issue and is made out of a development of qualities. The significant masses is taken subjectively to fill in as the starting stage for the check. A prosperity work is depicted to check the respectability of the chromosome for the earth. In light of success regard, chromosomes are picked and mix and change hones are performed on them to pass on offsprings for the new masses. The prosperity work surveys the likelihood of each tyke. The framework is underscored until the minute that the moment that worthy family are made. Pseudo code of GA mean update of booking issue in cloud is showed up in Fig.4 2



```
Procedure GA
   1. Initialization: Generate initial population P consisting of chromosomes.
   2. Fitness: Calculate the fitness value of each chromosome using fitness
      function.
   3. Selection: Select the chromosomes for producing next generation using
      selection operator.
   4. Crossover: Perform the crossover operation on the pair of chromosomes
      obtained in step 3.
   5. Mutation: Perform the mutation operation on the chromosomes.
   6. Fitness: Calculate the fitness value of these newly generated chromosomes
      known as offsprings.
   7. Replacement: Update the population P by replacing bad solutions with better
      chromosomes from offsprings.
   8. Repeat steps 3 to 7 until stopping condition is met. Stopping condition may
      be the maximum number of iterations or no change in fitness value of
      chromosomes for consecutive iterations.
   9. Output best chromosome as the final solution.
End Procedure
```

Fig. 4.2 GA procedure

V. PERFORMANCE MATRICES

A. Makespan

Makespan is a measure of the throughput of the heterogeneous figuring systems, for instance, cloud. It can be figured as the going with association

$$makespan = max(CTi)\,,$$
$$\text{where ti belongs to MT}$$

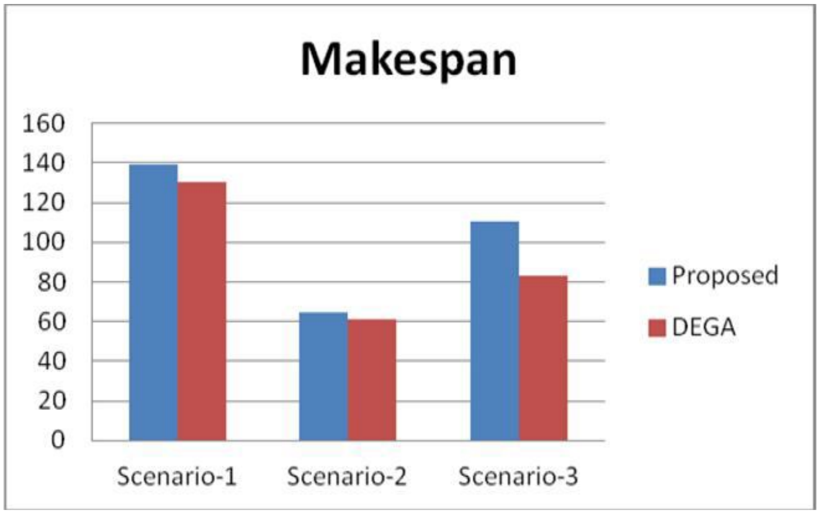


Fig. 5.1 comparison of makespan of both algorithms

The minimum time incurred in makespan the better algorithms will be considered.

B. Ordinary Resource Utilize rate

It is one of the estimations that is used as a piece of [14]. Typical resource use of each benefit can be registered.

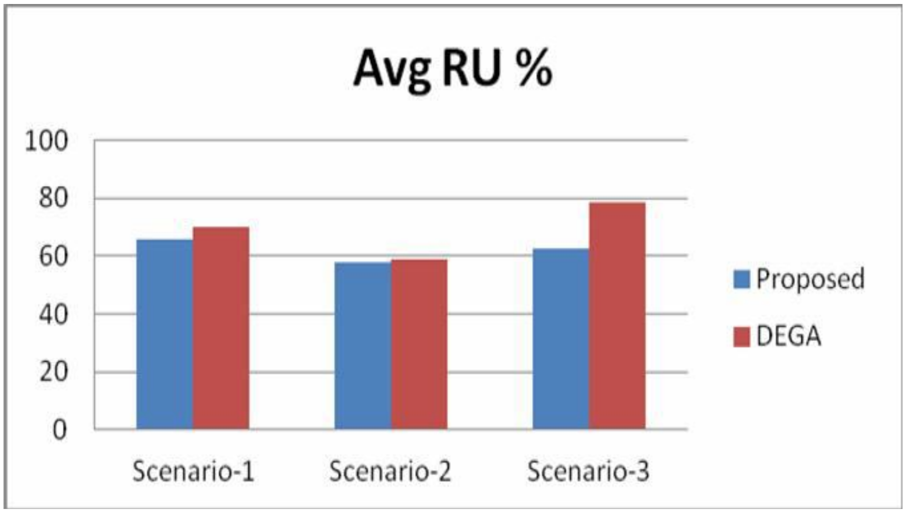


Fig. 5.2 Comparison of average resource utilization of algorithms.

C. Load Altering Level

The mean square deviation of ru is described as: The best and most beneficial load changing level is expert if d meets zero and β approaches 1. Thusly, arranging figuring will have better execution if d is close to 0 and β is almost 1. It is the other metric that is used as a piece.

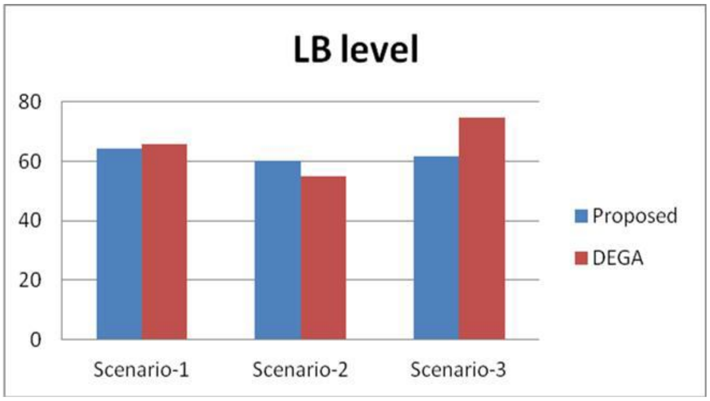


Fig. 5.3 LBL for both algorithms.

VI. CONCLUSION AND FUTURE SCOPE

In this paper , Cloud Computing is rising as the next generation platform, we have the need for management and scheduling system that allows us to access the cloud services for deployment and execution application. I have intended to accomplish this requirement by building a system, which will be easy to use, lightweight and scalable and allows the user to deploy his applications on cloud services and schedule efficiently.

REFERENCES

[1] Cloud computing concept (2016https://en.wikipedia.org/wiki/Cloud_computing april) wikipedia. [Online].

[2] Broberg J., Goscinskki A. Buyya r., Introduction t Pvt. 2016 to Cloud Computing. New Delhi: Wiley ludia_ Ltd.

[3] Shubham Mittal and Avita Katal, An Optimized Task Scheduling Algorithm in Cloud," IEEE 6th International Advanced Computing Conference, pp. 197-202, 2016.

[4] Santhosh B and Manj.aiah D H, "A Hybrid Av Task-Min and Max-Min Algorithm For Scheduling Tasks In Cloud Com utigg," International Conference on Control,lnstrumentation, Communieation and Computational Teehnologies (1CCICCT), IEEE, pp. 325-328, 2015.

[5] Rajkumar Buyya, James Broberg, and Andrej Goscinskki, Intoduction to Cloud Computing. New Delhi, India: Wiley India Pvt.Ltd.

[6] History of cloud computing. (2017, May) [Online]. https://www.google.co.in/search?q=history+of+cloud+computing+diagram&source—Inms&tbm=isch&sa=X&ved=0ahUKEwjE6OWn47zUAhWLr48KHUWQAYIQ_AUICigB&biw=1366&bih=662#imgrc=BGr2JhgRmGRzOM.

[7] cloud computing examples applications.(2016,November)[Online]. https://www.unc.edu/courses/2010spring/law/357c/001/cloudcomputing/examples.html

[8] Nidhi Bansal, Amit Awasthi, and Shruti Bansal, "Task Scheduling Algorithms with Multiple Factor in Cloud Computing Environment," Springer India, pp. 619-627, 2016.

[9] Layered Architecture of Cloud Computing. (2017, May) [Online]. https://www.google.co.in/search?q=layered+model+of+cloud+computing+architecture&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiKysHb5LzUAhUJuo8K1-IUVxBgsQ_AUICig B&biw=1366&bih=662#imgrc=uF3J_AwevK3AEM:

[10] Mohsin Nazir, "Cloud Computing: Overview and current research Challenges," pp. 14-22, 2012.

[11] Seema Sharma and Jyoti Godara, "Review paper on Cloud Computing," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 4, no. 6, Pp. 58-61, November - December Volume 4, Issue 6, 2015.

[12] Pirtpal Singh and Navpreet Kaur Walia, "A Review: Cloud Computing using Various Task Scheduling Algorithms," International Journal of Computer Applications(IJCA), vol. 142, no. 7, pp. 30-32, May 2016.