ISSN:0975-9646

S. Kowser Sultana et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (4) , 2016, 1847-1856

BALANCING LOAD USING GENETIC CRITERIA IN CLOUD COMPUTING

S. Kowser Sultana[1], G. DIVYA ZION[2], U. SUPRIYA[3]

[1]*M.Tech (CSE) Student, Dept of CSE, Ravindra Engineering College for Women, Kurnool, AP, India,*

[2]*Assistant Professor, Dept of CSE, Ravindra Engineering College for Women, Kurnool, AP, India.*

[3]*Assistant Professor,Dept of CSE, G.Pullaiah College of Engineering & Technology, Kurnool, AP, India.*

***Abstract*- Cloud computing uses the ideas of scheduling and load balancing to transfer tasks to underutilized VMs for effectively sharing the resources. IT based companies has already changed their way to buy and design hardware through this technology. To balance the load cloud environment can be used by splitting the amount of work by virtual machines in efficient way using existing algorithms. There are various classifications in load balancing algorithm. We present static or dynamic load balancing algorithms which are in comparison here. In this paper, we propose the dynamic algorithm i.e. genetic algorithm which is adaptable to balance the load.**

***Keywords*- Cloud Computing, Load balancing, Genetic Algorithm**

I. INTRODUCTION

Cloud Computing is believed to have been invented via "Joseph Carl Robnett licklider in the 1960's with his work scheduled "ARPANET"(The Advanced Research Projects Agency Networks) to connect people and data from anywhere at any time. J.C.R was one of the Americas leading computer scientists. The running generation of world, cloud computing is one of the most powerful, chief and also lightning technology. we can express that Cloud is relatively which is current at remote location. Cloud will offer services over network, i.e., on public sites or on private sites, i.e., WAN, LAN or VPN. Applications such as e-mail, conferencing meetings, customer relationship management (CRM), all run in cloud. Cloud is simply a metaphor for the internet. Cloud computing also known as on-demand processing. This is a kind of

Internet-based computing to provide shared processing resources and data to computer systems and other devices on demand. The cloud computing provides on demand personal services that means customers can request and control their own resources. The National Institute of criteria and Technology's definition of cloud computing identifies "five essential characteristics "are On Demand Self-Service, Broad network access, Resource pooling, Swift elasticity, Measured service.

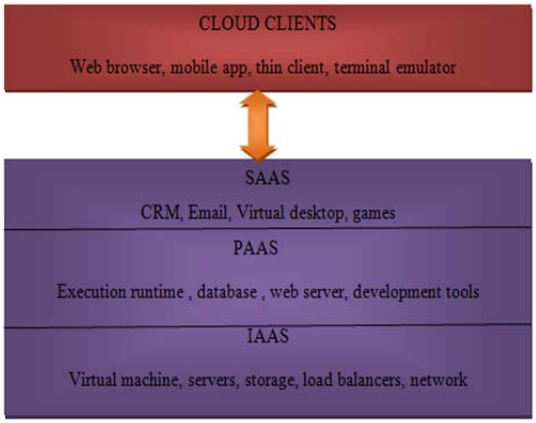


Fig1: Cloud architecture

Cloud computing primarily consists of three service models that are supplied to the end users those are as below:

A. Service Models

1. ***Infrastructure as a Service (IaaS):***
   IaaS supplies access to basic resources such as physical machines, virtual

machines, virtual storage, and so on. Apart from these sources, the IaaS also delivers Virtual machine disk storage, Virtual local area network (VLANs), Load balancers, IP addresses, Application bundles.

2. ***Platform as a Service (PaaS):***
   PaaS delivers the runtime surroundings for applications. It also gives improvement & deployment equipment, essential to build applications. PaaS has a attribute of point-and-click tools that permits non-developers to develop web applications.
3. ***Software as a Service (SaaS)***
   Software as a Service (SaaS) model enables providing software program application as a support to the end users. It refers to a software program that is deployed on a hosted service and is accessible via Internet.

.

B. Deployment Models

1. ***Public cloud model:*** The Public Cloud enables techniques and solution to be simply accessible to general public, e.g., Google, Amazon, Microsoft offers cloud services by means of internet.
2. ***Private cloud model:*** The Private Cloud permits techniques and solutions to be accessible within an organization. The Private Cloud is operated only within a single organization. Nevertheless, It might be managed internally or by third-party.
3. ***Hybrid cloud model:*** The Hybrid Cloud is a mixture of public and private cloud. Non-critical activities are carried using public cloud while the critical activities are carried out using private cloud.
4. ***Community cloud model:*** The Community Cloud permits techniques and solutions to be available by group of organizations. It shares the infrastructure among several organizations from a certain community. It might be managed internally or by the third-party.Without balancing load it is too much hard to manage this cloud computing. As a soft computing approach Genetic Algorithm has been used in this paper. Cloud Analyst - A Cloud Sims built Visual Modeler has been used for simulation and analysis of the algorithm.

Objective of the Project:

The major problem which we present in this paper is that cloud computing distributes workload dynamically across multiple nodes where no single resource is underutilized and is known as optimization problem. To overcome this problem the load balancing strategy we use Genetic Algorithm (GA).

II.RELATED WORK

D. E. Goldberg,[5] considered Genetic Algorithm as one of the most widely used artificial intelligent techniques used primarily for effective search and optimization. It is a stochastic searching algorithm based on the mechanisms of natural selection and genetics. GAs has been proven to be very efficient and stable in searching out global optimum solutions, specially in complex and/ or vast search space.

T. R. Armstrong, D. Hensgen, described Load balancing mechanisms can be broadly categorized as centralized or decentralized, dynamic or static, and periodic or non-periodic. There has been few research on load balancing techniques in cloud computing environment. Armstrong et. al.[6] uses Minimum Execution Time (MET) to assign order to each job in arbitrary manner to the nodes on which it is expected to be executed fastest, regardless of the current load on that node. Use of some existing scheduling techniques like Min-Min, Round Robin and FCFS for load balancing also exist in literature.

Holland, J. H., Srinivas, M. and Patnaik, L. M., described as an adaptive search technique, the genetic algorithm (GA) provides an alternative to traditional optimization techniques by using directed random searches to locate optimal or near optimal solutions of complex problems and is rooted in the mechanisms of evolution and natural genetics [7-8].

S.H.Bokhari, S.Salleh and A.Y.Zomaya described Load Balancing algorithms are designed essentially to equally spread the load on processors and maximize their utilization while minimizing the total task execution time [9], [10], [11]. In order to achieve these goals, the load balancing mechanism should be ªfairº in distributing the load across the processors.

C.A. Gonzalez Pico and R.L. Wainwright [12] ,described the objective is to determine task schedules ªon the flyº as the tasks arrive, the strings in the population are used to represent the list of tasks known to the system at the time. However, as there may be too many tasks waiting to be assigned at a time, the sliding-window technique is used so that only tasks that are within the window are considered for execution each time.

III. EXISTING SYSTEM

LOAD BALANCING: Load balancing is dividing the amount of work that a computer has to do between two or more computers. So that more work will get completed in the same amount of time and all end users get served more rapidly .

Load balancing can be implemented with hardware, software, or a mixture of each. A load balancing algorithm which is dynamic in nature does not contemplate the prior state or behaviour of the technique, that is, it depends on the existing behaviour of the technique. The objectives of load balancing are:

- To enhance functionality considerably
- To have a backup strategy in situation the technique fails even partially
- To keep the technique stability
- To accommodate future modification in the technique

Based on the existing state of the technique, load balancing algorithms can be divided in to two categories:

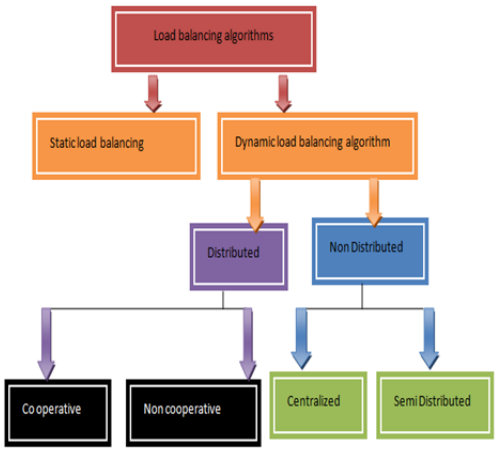


Fig 2: Types of Load balancing algorithms

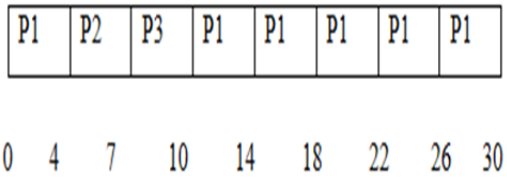
A. Static Load Balancing Algorithm

It doesn't rely on the existing state of the program . Prior information of the program is needed .In static environment the cloud provider installs homogeneous resources. Also the resources in the cloud are not versatile when environment is produce static. In this scenario, the cloud demands prior information of nodes capability, processing power , memory, performance and statistics of consumer demands. These consumer demands are not subjected to any alter at run-time. Algorithms proposed to attain load balancing in static environment can't adapt to the run time alterations in load. Even though static environment is simpler to simulate but is not effectively suited for heterogeneous cloud environment. Static load balancing algorithms are :

1. ***Round Robin Algorithm:*** Round Robin algorithm [1] provides load balancing in static environment. Round Robin is one of the efficient algorithms for CPU scheduling. The simple approach in RR is it gives equal CPU time to each process. This CPU time is known as a quantum. The functionality of RR is basically depends upon the quantum value [2,3]. There is no distinct approach is accessible to select the quantum. If quantum is too large it maximizes the waiting time whereas if it is too small it increases the context switching. The quantum can be fixed throughout all iterations or alter dynamically primarily based on some optimization technique. This method continues until all processes complete their execution on CPU. The drawback of RR is that the largest job takes enough time for completion.

EXAMPLE OF RR WITH TIME QUANTUM = 4

Process	Burst Time
P_1	24
P_2	3
P_3	3



Waiting Time:

P1: (10-4) = 6

P2: (4-0) = 4

P3: (7-0) = 7

Completion Time:

P1: 30

P2: 7

P3: 10

Average Waiting Time: (6 + 4 + 7)/3= 5.67

Average Completion Time: (30+7+10)/3=15.67

2. First Come First Serve (FCFS):

The best CPU scheduling algorithms is First Come First Serve (FCFS) which selects the first arrived process for execution from the ready queue [4]. Regarding to priority, such that the process having top priority will execute first. The disadvantages of FCFS is that it is non pre-emptive. The least tasks which are at the back of the queue have to wait around for the long process at the front to complete

.Its turnaround and response is quite low. Hence, FCFS always select first process.

B. Dynamic Load balancing algorithm

Decisions on load balancing are primarily based on existing state of the program. No prior information is required. So it is better than static approach. Right here we will examine on various dynamic load balancing algorithms for the clouds of different sizes. These algorithms are more resilient than static algorithms, may easily adapt to alteration and provide better results in heterogeneous and dynamic environments. In dynamic environment the cloud service provider installs heterogeneous resources. The resources are flexible in dynamic environment. In this scenario cloud cannot rely on the prior knowledge whereas it takes into account run-time statistics. The requirements of the users are granted over all flexibility (i.e. they may change at run-time). Algorithm proposed to attain load balancing in dynamic environment can certainly adapt to run time changes in load. Dynamic environment is difficult to be simulated but is highly versatile with cloud computing environment. Dynamic load balancing algorithms have two types. That is thought as following:

1. ***Distributed approach:***

In this approach the dynamic load balancing algorithm is executed by all nodes present in the system and the task of load balancing is shared among them. The interaction among nodes to achieve load balancing can take two forms:

i. ***Cooperative***: In this, the nodes work side-by-side to achieve a common objective, for example, to improve the overall response time.

ii. ***Non-cooperative***: In this, each node works independently toward a goal local to it, for example, to improve the response time of a local task.

2. ***Non Distributed approach:***

In this approach either one node or a group of nodes do the task of load balancing. Non-distributed dynamic load balancing algorithms can take two forms:

i. ***Centralized***: In this, the load balancing algorithm is executed only by a single node in the whole system: the central node. This node is solely responsible for load balancing of the whole system. The other nodes interact only with the central node.

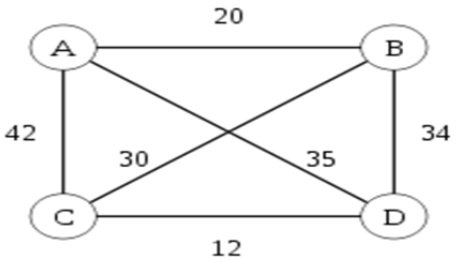
ii. ***Semi-distributed:*** In this, nodes of the system are partitioned into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in each cluster by appropriate election technique which takes care of load balancing within that cluster.

C. Policies or Strategies in dynamic load balancing

1. ***Transfer Policy:*** The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote node is referred to as Transfer policy or Transfer strategy.
2. ***Selection Policy:*** It specifies the processors involved in the load exchange (processor matching)
3. ***Location Policy:*** The part of the load balancing algorithm which selects a destination node for a transferred task is referred to as location policy or Location strategy.
4. ***Information Policy:*** The part of the dynamic load balancing algorithm responsible for collecting information about the nodes in the system is referred to as Information policy or Information strategy.

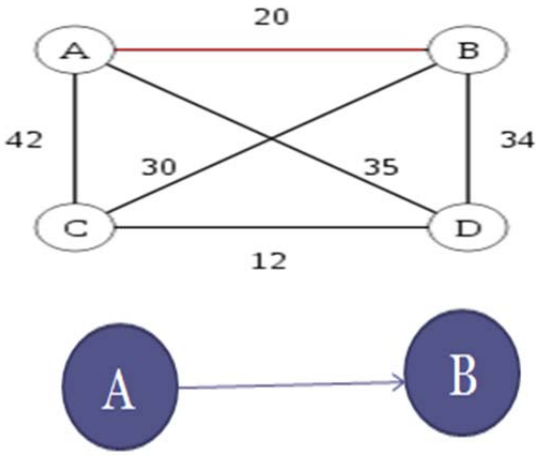
Ant colony optimization: The objective of this algorithm [13] is trying to minimize the makespan and maintain the load across all the nodes with give the better results than first come and first serve. The ant colony optimization load balancing technique ability to find the optimal path to find the solution for optimization problems. The ants search for new food that all are work together by using the existing food as a source to come back to the nest. In [14] modify the existing ant colony technique and that has been applied from the perspective of grid network systems or cloud with the aim of load balancing. The task depends on the type of first node that was encountered whether it was under loaded or overloaded. The best example of ACO is TSP problem.

- The Problem is how to travel from city A and visit all city on the map, then back to city A again.

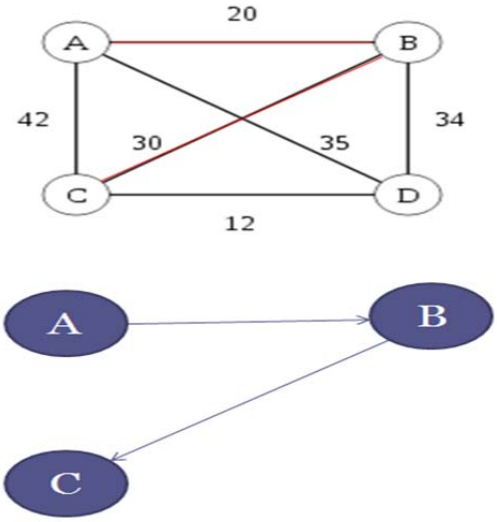


The rules: you only visit each city once and you can't pass through any traversed path.

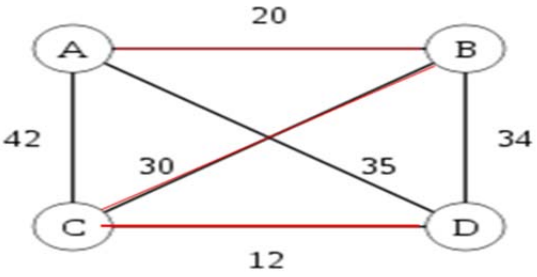
Solution: Find the shortest path from city A (start) to any other city.

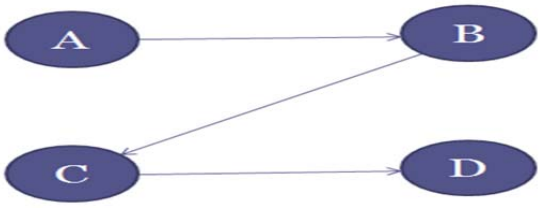


Because the nearest city is B, so we go to B. From B, we find any other city but A(because A has been visited) that has nearest path. So we choose C:

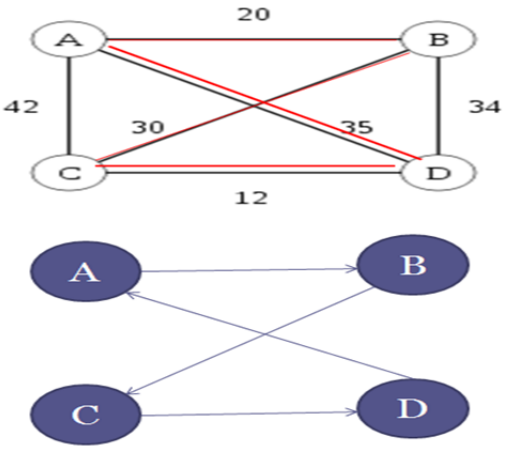


From C, we look to nearest city again, but don't look for A and B, because both has been visited. So we choose D.





At this node (D), we can't go to any city, because all neighbor of D has been visited. We go back to first city(A).



We only visit each city once and we can't pass through any traversed path.

Shortest path is: A-B-C-D-A

= 20+30+12+35

= 97

Algorithm	Static Environment	Dynamic Environment	Centralized Balancing	Distributed Balancing	Hierarchical Balancing
Round Robin	Yes	No	Yes	No	No
FCFS	Yes	No	Yes	No	No
Ant Colony	No	Yes	No	Yes	No
Genetic Algorithm	No	Yes	Yes	No	No

Table1: Comparison of Load Balancing Algorithms in cloud Environment

IV. PROPOSED SYSTEM

Best Example of dynamic load balancing algorithm is Genetic Algorithm. Genetic Algorithm is dynamic environment and also a centralized environment. Genetic Algorithm is used to find the best solution to a given problem. Genetic Algorithms (GAs) are search based algorithms based on the concepts of natural selection and genetics. some basic terminology which will be used are:

Population – It is a subset of all the possible (encoded) solutions to the given problem. The population for a GA is analogous to the population for human beings except that instead of human beings, we have **Candidate Solutions** representing human beings.

Chromosomes – A chromosome is one such solution to the given problem.

Gene – A gene is one element position of a chromosome.

Allele – It is the value a gene takes for a particular chromosome.

There are two basic genetic algorithms operators which are crossover and mutation. These two operators are work together to explore and exploit the search space by creating new variants in the chromosomes. There are many empirical studies on a comparison between crossover and mutation. It is confirmed that mutation operator play the same important role as that of the crossover.

A. Genetic Algorithms Operators

1. ***Selection:*** It is usually the first operator applied on population. Chromosomes are selected from the population of parents to cross over and produce offspring. It is based on Darwin's evolution theory of "Survival of the fittest". There is a method to select the best chromosomes is Roulette Wheel Selection.The better the chromosomes are, the more chances to be selected they have.

Roulette Wheel Selection A string is selected from the mating pool with a probability proportional to the fitness.Probability of the ith selected string: Chance to be selected is exactly proportional to fitness. Chromosome with bigger fitness will be selected more times.

2. ***Cross Over:*** After selection phase, population is enriched with better individuals. It makes clones of good strings but does not create new ones. Cross over operator is applied to the mating pool with a hope that it would create better strings.

3. ***Mutation:*** After cross over, the strings are subjected to mutation. Mutation of a bit involves flipping it, changing 0 to 1 and vice-versa.

Simple Genetic Algorithm

Step 1: Encoding of the problem in a binary string

Step 2: Random generation of a population

Step 3: Calculate fitness of each solution

Step 4: Select pairs of parent strings based on fitness

Step 5: Generate new string with crossover and mutation until a new population has been produced

Repeat step 2 to 5 until satisfying solution is obtained

A simple example:

Traveling Salesman Problem (TSP) by using Genetic Algorithm

- Assume number of cities N = 10
- **Representation:** it is an ordered list of city

1) London 2)Venice 3)Dunedin

4) Singapore 5) Beijing 6) Phoenix

7) Tokyo 8) Victoria

CityList1 (3 5 7 2 1 6 4 8)

CityList2 (2 5 7 6 8 1 3 4)

- **Crossover:**

Parent1 (3 5 | 7 2 1 6 | 4 8)

Parent2 (2 5 | 7 6 8 1 | 3 4)

Child (5 8 | 7 2 1 6 | 3 4)

- **Mutation:**

Before: (5 8 7 2 1 6 3 4)

After: (5 8 6 2 1 7 3 4)

V. SYSTEM DESIGN

To balance load by using Genetic Algorithm

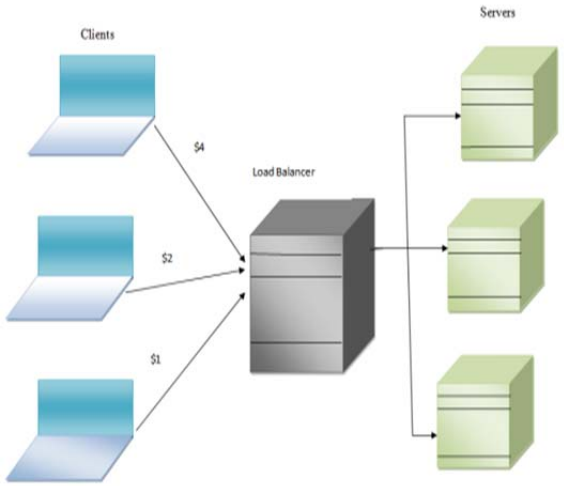


Fig 4: Organizing server to balance load

The proposed balancing technique is given below.
Step 1: At the starting point, randomly initialize a population of processing unit after coding them into binary strings.

Step 2: Calculate the main value of each population for the fitness part.

Step 3: If the maximum number of iteration or optimum solution is found then do:
Step 3(a): Consider chromosome with lowest fitness twice and delete the chromosome with maximum fitness value to build the mating pool.

Step 3(b): Use single point crossover by randomly picking the crossover point to make new offspring.

Step 3(c): With a mutation probability of (0.05), mutate new offspring.

Step 3(d): As new population place new offspring and use this population for following round of iteration. This is called the accepting part.

Step 3(e): Finally test for the end condition.

Step 4: End.

PROCESSORS	CURRENT TASK	PROCESSOR QUEUE	TOTAL LENGTH (TIME UNITS)
1	5(4)	3(1) 2(2)	7
2	1(20)		20
3	7(18)		18
4	10(9)	9(5)	14
5	8(10)	6(15) 4(3)	28
Total load in the system			87

Table 1: State of the System at the Outset

Step 1: Current State of the System:

Table 1 holds the information at time t. Since processor 3 is idle, a new set of tasks has to be allocated.

Step 2: New Tasks to be scheduled:

The new set of tasks within the sliding-window is shown in Table 2.

11(8)	12(7)	13(5)	14(11)	15(16)	16(6)	17(5)	18(13)	19(10)	20(12)

Table 2: New Tasks inserted into the Sliding Window

Total time units=8+7+5+11+16+6+5+13+10+12 = 93

Step 3: Fittest Task Mapping Generated by the GA:

Based on the set of tasks in Step 2, the fittest task mapping generated using the GA

According to table -2

P1: 13(5) 15(16)

P2: 11(18) 14(11)

P3: 12(7) 17(15)

P4: 18(13) 20(12)

P5: 16(6) 19(10)

Load (time units) added to processors:

P1:5+16= 21

P2: 8+11= 19

P3: 7+15= 22 =>**93**

P4: 13+12= 25

P5: 6+10= 16

Step 4: Determine the New Thresholds:

The next step is to determine new thresholds based on the current system load (given in step 1) and new tasks to be scheduled (Step 2):

L_{ave} = (87+93)/5

L_{ave} = 36

T_H = 1.2 * 36 = 43.2

T_L = 0.8 * 36 = 28.8

Hence, the new heavy threshold is 43 while the light threshold is 29.

Step 5: Find Acceptable Load Sizes for each Processor:

The acceptable load that each processor can accept in this round can be calculated by subtracting the current processor queues (Step 1) from the heavy threshold determined in Step 4.

Acceptable load ≤ heavy threshold − total queue length of processor

P1: ≤ 43-7 = 36

P2: ≤ 43-20 = 23

P3: ≤ 43-18= 25

P4: ≤ 43-14 = 29

P5: ≤ 43-28 = 15

Step 6: Task Assignment Decision.

Hence, by inspecting the task assignment in Step 3, we can see that the new set of tasks for processor 1 can be assigned since its load is less than the acceptable load. The same applies to the tasks on processors 2-4. However, the new tasks for processor 5 will not be assigned since its load of 16 exceeds the acceptable load of 15. Therefore, only tasks for processors 1-4 in the generated task mapping will be assigned to the respective processors.If processor queues that exceed the acceptable load sizes are still assigned, this will cause a more severe load imbalance situation.

Step 7. New State of the System:

After the assignment of tasks, the load information table at time t+1 is as shown in Table-3.From the above table-1 processor 1 is detected to be idle. Therefore, another list of tasks has to be assigned. In order to do this, the sliding-window must be filled up with new tasks again.

PROCESSORS	CURRENT TASK	PROCESSOR QUEUE	TOTAL QUEUE LENGTH(TIME UNITS)
1	5(3)	3(1) 2(2) 13(5) 15(16)	27
2	1(19)	11(8) 14(11)	38
3	7(17)	12(7) 17(15)	39
4	10(8)	9(5) 18(13) 20(12)	38
5	8(9)	6(15) 4(3)	26
Total load in the system			168

Table 3: After the assignment of tasks, the load information table at time t+1

Step 8: Updating the Sliding Window

Previously (in Step 2), the sliding-window consisted of the tasks in Table 4.

11(8)	12(7)	13(5)	14(11)	15(16)	16(6)	17(5)	18(13)	19(10)	20(12)

TABLE - 4 Contents of Old Sliding Window

- As tasks 11, 12, 13,14, 15, 17, 18 and 20 were assigned to the processors in step 7, these tasks in the sliding-window have to be replaced with eight new tasks from the system task queue.
- Hence, the new sliding window will have the tasks shown in Table 5.

21(4)	22(7)	23(8)	24(10)	25(2)	16(6)	26(15)	27(12)	19(10)	28(9)

TABLE - 5 Contents of New Sliding Window

Total time units = 4+7+8+10+2+6+15+12+10+9= 83

With this new set of tasks to assign, all the above steps will be repeated again.

VI. SIMULATION RESULTS AND ANALYSIS

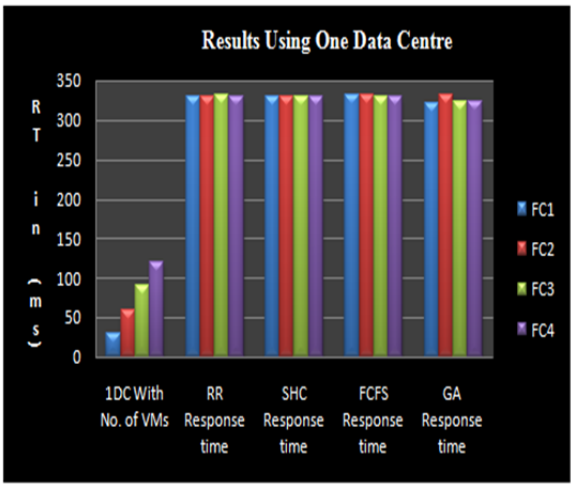


Fig1: Performance analysis of proposed GA with SHC, FCFS and RR by using one data centre.

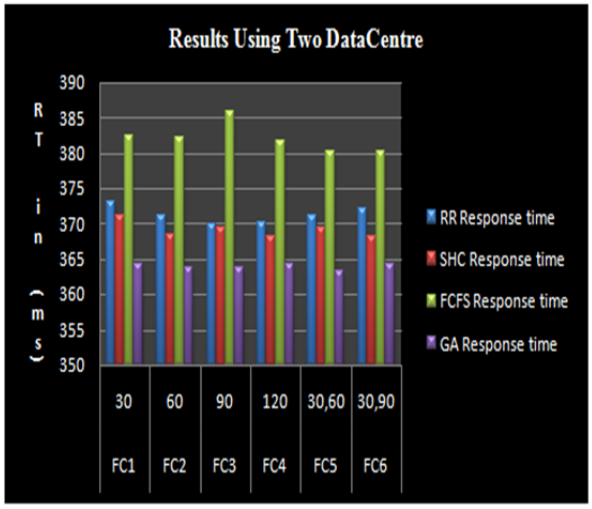


Fig2: Performance analysis of proposed GA with SHC, FCFS and RR by using two data centres.

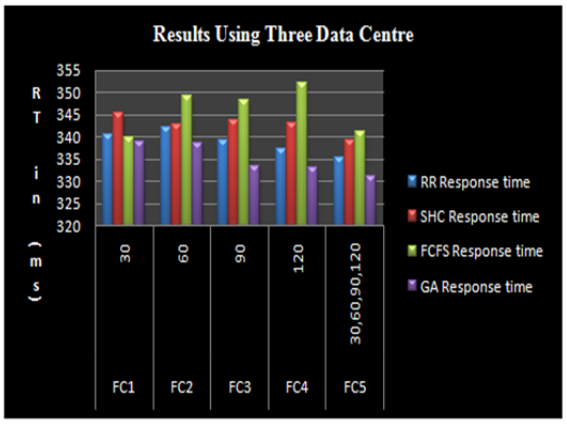


Fig3: Performance analysis of proposed GA with SHC, FCFS and RR by using three data centre.

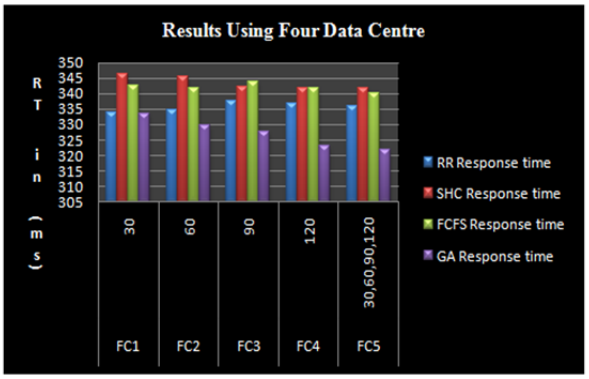


Fig4: Performance analysis of proposed GA with SHC, FCFS and RR by using four data centre.

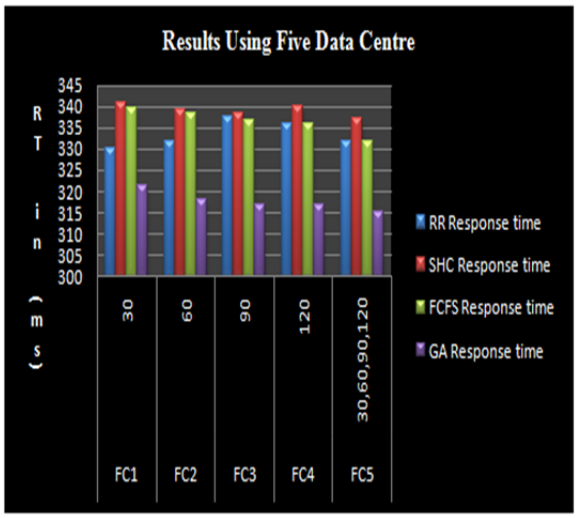


Fig5: Performance analysis of proposed GA with SHC, FCFS and RR by using five data centre.

From the graphs shown above it can be conclude that the proposed system is better for the latest technology. By comparing the results GA performs less response time. This paper can do better load balancing with the help of Genetic Algorithm in the cloud network. In this paper, a genetic algorithm based load balancing strategy for Cloud Computing has been developed to provide an efficient utilization of resource in cloud environment. Analysis of the results, indicates that the proposed strategy for load balancing not only outperforms a few existing techniques but also guarantees the QoS requirement of customer job.

VII. CONCLUSION & FUTURE WORK

Cloud computing is a really cheap way for companies to have all the resources they need in once place. It is a much better way to spread your resources, and it becomes easier to access things from longer distance. Cloud computing provides everything to the user as a service which includes platform as a service, application as a service, infrastructure as a service. One of the major issues

of cloud computing is load balancing because overloading of a system may lead to poor performance which can make the technology unsuccessful. So there is eternally a requirement of efficient load balancing algorithm for efficient utilization of resources. Our paper focuses on the different load balancing algorithms and their applicability in cloud computing environment.

In future work we will discuss Simulated Annealing for global search optimization in cloud load balancing and simulation. Some additional algorithms which can help in solving some sub-problems in load balancing which are applicable to cloud computing.

REFERENCES

[1] Sotomayor, B., Montero, R. S., Llorente, I. M. & Foster, I. (2009). Virtual infrastructure management in private and hybrid clouds. IEEE Internet Computing, 13(5), 14-22.

[2] Abraham S. & Peter B. Galvin & Greg Gagne," Operating System Concepts ", John Wiley & Sons,Inc. 2007.

[3] Silberschatzs, A., Galvin, P.B., Gagne, G. 2009, "Operating System Concepts. Massachusetts: Addison Wesley".

[4] Silberschatzs, A., Galvin, P.B., Gagne, G. 2009, "Operating System Concepts. Massachusetts: Addison Wesley".

[5] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning* Addison Wesley, 1989.

[6] T. R. Armstrong, D. Hensgen, "The relative performance of various mapping algorithms is independent of sizable variances in runtime predictions", in Proc. of *7th IEEE Heterogeneous Computing Workshop (HCW 98)*, pp. 79-87,1998.

[7] Holland, J. H., *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press, Ann Arbor, U.S.A. (1975).

[8] Srinivas, M. and Patnaik, L. M., "Genetic Algorithms: A Survey," *IEEE Computer*, Vol. 27, pp. 17-26 (1994).

[9] S.H. Bokhari, ªOn the Mapping Problem,º IEEE Trans. Computers, vol. 30, no. 3, pp. 550 557, Mar. 1981.

[10] S. Salleh and A.Y. Zomaya, Scheduling in Parallel Computing Systems: Fuzzy and Annealing Techniques. Kluwer Academic, 1999.

[11] A.Y. Zomaya, ªParallel and Distributed Computing: The Scene, the Props, the Players,º Parallel and Distributed Computing Handbook, A.Y. Zomaya, ed., pp. 5-23. New York: McGraw-Hill, 1996.

[12] C.A. Gonzalez Pico and R.L. Wainwright, ªDynamic Scheduling of Computer Tasks Using Genetic Algorithms,º Proc. First IEEE Conf. Evolutionary Computation, IEEE World Congress Computational Intelligence, vol. 2, pp. 829-833, 1994.

[13] Kun Li, et al., "*Cloud Task scheduling based on Load Balancing Ant Colony Optimization*" Journal of IEEE (2011).

[14] Kumar Nishant, et al. "*Load Balancing of Nodes in Cloud Using Ant Colony Optimization*" Journal of IEEE (2012).

[15] www.tutorialspoint.com