

# Using Genetic Algorithm for Load Balancing in Cloud Computing

Hussain A Makasarwala  
 M.E CSE  
 PIET Vadodara

Prasun Hazari  
 Asst Prof CSE Dept.  
 PIET Vadodara.

**Abstract**— Cloud Computing is set of resources available on pay per use model. The User requests on cloud for the services which is increasing day by day and for the better performance it should be balanced such that all the requests should be answered in minimum time. Therefore, Load Balancing is required and it is one of the major Issues in Cloud Computing. Moreover, it is termed as Load Balancing is NP-Complete Problem because as the number of request increases, balancing the load becomes tougher. This paper gives a genetic algorithm (GA) based approach for load balancing in cloud. For population initialization, priority of request is considered based on their time. The idea behind the considering the priority is to get real world visualization. In Real World Scenario requests have some priorities which we can use for our Algorithm. Simulation of the proposed method is done using Cloud Analyst. Simulation Results shows that proposed method performs well then existing once with some real world picture.

**Keywords**—Genetic Algorithm, GA, Cloud Computing, Load Balancing.

## I. INTRODUCTION

Cloud Computing is emerging field for research and study. It is a pool of multiple configurable computing resources available on demand to user. The field is evolved from past technologies like web services, hardware virtualization, grid and utility computing, system management.

Generally, Cloud provides mainly three types of services which are as follows:

- **SAAS (Software as a Service).**  
The software's or applications which are preinstalled on cloud and can be used by End users
- **PAAS (Platform as a service).**  
It provides a services of providing a platform for developing or testing the application made by programmer on cloud.
- **IAAS (Infrastructure as a Service).**  
It mainly deals which the services required for providing Hardware or networking.

The cloud computing mainly gets deployed on four deployment servers by which the users can access the services provided by clouds. They are as follows:

- **Public Cloud**  
A cloud which can be access by anyone freely.
- **Private Cloud**  
This kind of clouds can be deployed for a particular institution or organisation which can be accessed by the user of that organisation only or authenticated users
- **Community Cloud**  
A cloud service which is shared amongst one or more organisation or can be said as community but not available publically.
- **Hybrid Cloud**  
This type of cloud can be obtained by combining the two types of cloud services

Moreover, Concepts of virtualization is used in cloud which leads to have a load balancing in the cloud. Virtualization means giving a logical name to physical resources and whenever this name is referred it will point towards corresponding physical resource. Multiple users will access cloud at same time and it is very necessary to serve them all with minimum response time and better service. For this reason load balancing is taken in to effect to balance the request of multiple users on virtual machines evenly. It is said that Load balancing is a NP-Complete Problem method because as the size of the problem increases the size of solution will increase too. That means the more the request comes to cloud it will get tougher to do balancing amongst the Different Virtual Machines.

Genetic Algorithm (GA) is much popular for solving NP-Complete problems. GA is one of technique which belongs to the class of evolutionary algorithms which generates solutions inspired by natural evolution. The Simple GA Concepts are as follows:

- **Population-** it is a set of possible solutions for proposed problem.
- **Chromosome-** the individuals in the population

- **Gene-** a variable in a chromosome

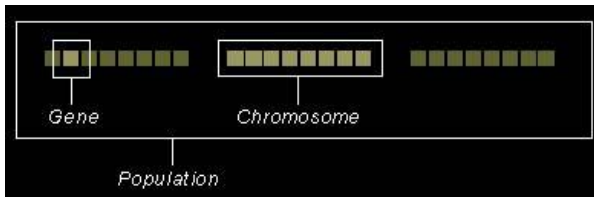


Fig 1: GA Population [13]

- **Fitness Function-** a type of an objective function used to figure out how close the solution is achieving the set aim
- The GA evolves through three operators:-
  - **Selection-** solutions with best fittest are selected
  - **Crossover-** for generation of Child, more than one parent is selected.
  - **Mutation-** altering the gene value in chromosome.

The Next section gives the idea about the different techniques of Genetic Algorithm for Load Balancing in Cloud Computing is been Discussed.

## II. RELATED WORK

There are many different techniques given by different author for load balancing in cloud Computing, from which some of them are discussed as below

Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam [1] has worked with the simple GA concepts. The Eye catching idea of algorithm is introducing the concept of Job Unit Vector (JUV) and Processing Unit Vector (PUV) which was the base for making fitness function. Binary encoding is used by authors for chromosome representation. Single point crossover with 0.5 Mutation rate was the author's algorithm GA parameters. Cloud Analyst is used as a stimulation tool and author has took different results with different number of data centers which was proved as better then Round Robin or Shortest job First methods. Moreover Author has considered same priority for all the requests and concluded that Ga provide Good QoS.

Tingting Wang, Zhaobin Liu, Yi Chen, Yujie Xu, Xiaoming Dai [2] has provided the idea of using two fitness function such that first fitness function calculates the time performance of VM and Second fitness calculates the intensity of different node's Load using Variance. Numbers are used for chromosome representation which can be called as permutation encoding. A single point crossover and Mutation is performed after selecting two best fitness chromosomes. Algorithm end when terminating condition is reached. Authors had compared there proposed method with

original GA and shown the results. Also all the requests are of same priority was the consideration done by authors.

Saeed Javanmardi, Mohammed Shojaraf, Danilo Amendola, Nicola Cordeschi, Hongu Liu and Ajith Abraham [3] has proposed a method in which they used fuzzy theory with genetic algorithm. The parent chromosomes are selected based on the fuzzy inference system output. Also crossover is performed using fuzzy theory. The requests are having same priority is the consideration done by authors. The paper has no Pseudo Code or any result comparisons with other Available Algorithms, but the conclusion of the authors are there proposed algorithm improves the performance of system in terms of execution time.

Santanu Dam, Gopal Mandal, Kousik Dasgupta and Paramartha Dutta [4] has proposed a method which is a combination of two methods one is GA and other is Gravitational Emulation Local Search (GELS). GELS is used for initiating Population for GA. Initial population is generated based on velocity calculation of chromosome done by GELS. Then, the two chromosomes are selected based on Fitness and two point crossover and Mutation is applied. Again velocity is calculated for chromosome and fitness chromosomes are added to new population. Then the algorithm ends. By using Cloud analyst stimulation of the proposed algorithm is carried out and satisfactory results are obtained by authors. No priority is considered for the request by the authors which cannot be the real scenario.

Jinhua GU, Jinhua Hu, Tianhai Zhao, Guofei Sun [5] has given an approach in which Tree Structure are used for chromosome representation. Initialization of population is done by using concept of spanning tree in which the root node is predefined server node, first level is Physical Machines and Second level is Virtual Machines. The algorithm proceeds through selection of fittest chromosome. Crossover operator is applied on selected chromosomes such that the chromosomes selected are having same leaf nodes. Mutation is done by changing leaf nodes of the parent node randomly.

Neha Gupta, Parminder Singh [6] has proposed a method in which the tasks are divided in to sub tasks. The reason behind the division is the parallel execution of jobs on multiple machine. The GA works for distribution of sub tasks to the multiple processors such that execution of one full task will be faster compared to working on one system. Initial population consist of two array consists of VMs and request respectively. Author did not provide any knowledge about use of Crossover. Mutation is applied to fittest chromosome to find the best VM for the job. The stimulation of proposed system is done in Cloud Sim 2.0

Mayur S. Pilavare, Amish Desai [7] has proposed the method in which the VMs are prioritized by using Logarithmic Lease Mean Square Matrix. This Gives priority to the VMs and initial population is made such that VMs having high priority gets the Jobs first. After these steps, GA operators are applied

to obtain optimum output. Here there is no priority given to the jobs.

On basis of this survey, one common point which some of the papers are having is that there is no priority is given to any of the requests which cannot be the real scenario

### III. PROPOSED METHOD

In our proposed method, Genetic Algorithm is a key role for load balancing. Population Initialization is done based on priority. The priority of request is being calculated based on the time. It is calculated from the length of the Job. More the length of the job, it will be more time consuming. The request require less time is more prior. The GA parameters are considered as follows

#### A. Chromosome Representation

In the proposed method, permutation encoding is used as chromosome representation. That is decimal numbers are used for genes representation. The example of chromosome representation used in the method is as follows:

$$A = \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\}$$

In the above shown example, the digit represents the Virtual Machine Identifier (VM\_ID) and place at which it is placed is Cloudlet ID (C\_ID).

#### B. Fitness Function

Fitness function is the base part for evolution of algorithm. This function is a decision making in the algorithm for selecting the algorithm. The chromosomes having good fitness is selected for generating child.

The Fitness function used in our proposed method is based on the Cloudlet Length (N), Processor Speed (MIPS), Current Load (u), and Capacity of VM (CP).

$$F = (N/MIPS) + (u/CP)$$

The more the value of fitness function the less the chromosome is fit.

#### C. Crossover Operator

Partial Mapping Crossover (PMX) is used as crossover operator in our method. PMX is a special type of crossover operator which is used for ordered chromosomes.

In PMX, A two cut points are taken and mapping is done between this two cut points.

For E.g.

Parent1 - 1 2 3 | 6 5 | 4  
 Parent2 - 6 2 1 | 3 4 | 5  
 Child1 - 1 2 6 | 3 4 | 5  
 Child2 - 3 2 1 | 6 5 | 4

Here, two cut point is taken. Block is interchange between two parents and duplicates are sorted using mapping between block i.e. in above example 6 is replaced by 3 and 5 with 4 and vice versa.

#### D. Mutation Operator

Mutation is done for abrupt change in population. In our Proposed Method Swapping of Digits is done.

#### E. Flow Chart of Proposed Method

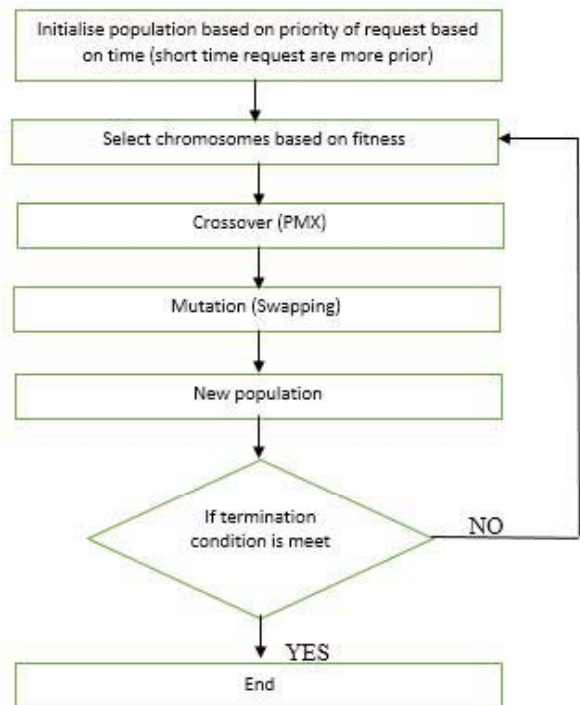


Fig 2: Flow Chart of Proposed Method

#### F. Algorithm of Proposed Method

- Step 1: Calculate the priority of incoming request on based of their time.
- Step 2: Initialize the population on based on above priority of jobs.
- Step 3: Select the best chromosome based on fitness
- Step 4: Perform Partially Mapping Crossover.
- Step 5: Perform Mutation by using Swapping technique.
- Step 6: Add the chromosome to new population
- Step 7: Check for termination condition
- Step 8: End

IV. RESULTS

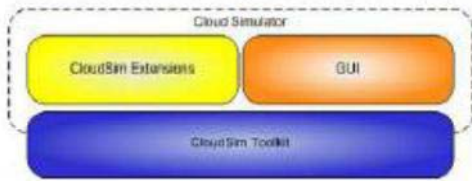
A. Simulation Tool.

For Simulation of Proposed method, Cloud Analyst has been used in this paper. Cloud Analyst developed on CloudSim is a GUI based simulation tool. The Cloud Analyst also enables a modeler to repeatedly execute simulations and to conduct a series of simulation experiments with slight parameters variations in a quick and easy manner [14]. The Main Features of Cloud Analyst are [14]:

- Graphical User Interface.
- Repeatability of Experiments.
- High Degree of Configurability & Flexibility.
- Graphical Output.



(a)



(b)

Fig 3: (a) GUI of Cloud Analyst [1] (b) Architecture of Cloud Analyst build on CloudSim [1]

B. Simulation Setup

Several scenario is being considered for simulation of proposed Method starting with 1 Data Center 25 VMs, 50VM, 2 Data Center with 25VMs, 50 VMs and with varying numbers of Avg. Peak Users.

Based on the above considerations, calculation of overall response time is done. The simulation is done for other algorithms like Round Robin, Throttled etc. for comparing the results with proposed method.

C. Results

Overall Response time in (ms) is being calculated and results are obtained for different scenarios as follows:

TABLE 1: 1 Data Center 25 VMs- users in 100s

| Proposed GA | Basic GA | Round Robin | Throttled | ESCEL  |
|-------------|----------|-------------|-----------|--------|
| 248.94      | 249.19   | 249.35      | 249.21    | 249.24 |

TABLE 2: 1 Data Center 50 VMs. - users in 1000s.

| Proposed GA | Basic GA | Round Robin | Throttled | ESCEL  |
|-------------|----------|-------------|-----------|--------|
| 254.99      | 255.04   | 255.32      | 255.16    | 255.15 |

TABLE 3: 2 Data Center 25 VMs. - users in 1000s.

| Proposed GA | Basic GA | Round Robin | Throttled | ESCEL  |
|-------------|----------|-------------|-----------|--------|
| 255.92      | 256.00   | 257.08      | 256.03    | 256.70 |

TABLE 4: 2 Data Center 50 VMs. - users in 100s.

| Proposed GA | Basic GA | Round Robin | Throttled | ESCEL  |
|-------------|----------|-------------|-----------|--------|
| 249.35      | 249.63   | 250.04      | 249.65    | 249.70 |

TABLE 5: 2 Data Center 50 VMs. - users in 1000s.

| Proposed GA | Basic GA | Round Robin | Throttled | ESCEL  |
|-------------|----------|-------------|-----------|--------|
| 256.24      | 256.27   | 257.45      | 256.28    | 256.94 |

Based on above results, Performance analysis of proposed method is generated as follows:

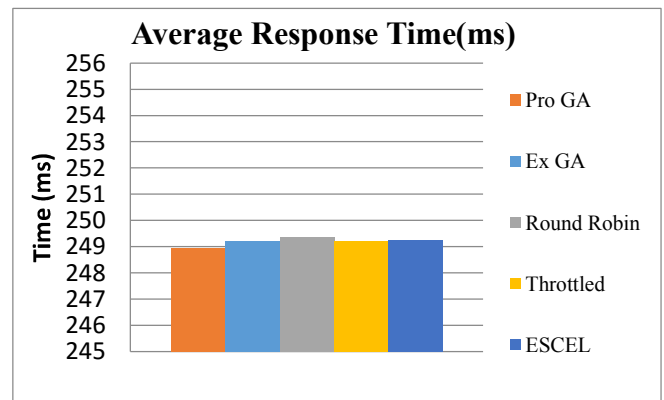


Fig 4: Performance Analysis of proposed method based on TABLE 1



V. CONCLUSION

From the above results and performance analysis, we can conclude that the proposed method gives better Average Response time compared to previous available Methods. The concept of using priority of request can be the deciding factor for faster results. In Our proposed method, we have considered time but there can be many other parameter which can be considered for calculating priority of request. Moreover, using permutation encoding increase the range of request ID's (Cloudlets). The proposed method idea can be used for visualization of working of GA in real world scenario which was our one of the objective.

VI. REFERENCES

- [1] Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", in International Conference on Computational Intelligence: Modelling Techniques and Applications (CIMTA) 2013, Procedia Technology 10 – 2013, Elsevier Ltd. 2013.
- [2] Tingting Wang, Zhaobin Liu, Yi Chen, Yujie Xu, Xiaoming Dai, "Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing", in IEEE 12<sup>th</sup> International Conference on Dependable, Autonomic and Secure Computing 2014.
- [3] Saeed Javanmardi, Mohammed Shojafar, Danilo Amendola, Nicola Cordeschi, Hongbo Liu and Ajith Abraham, "Hybrid Job Scheduling Algorithm for Cloud Computing Environment", in Proceedings of the 5th International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014, Springer 2014.
- [4] Santanu Dam, Gopa Mandal, Kousik Dasgupta and Paramartha Dutta, "Genetic Algorithm and Gravitational Emulation Based Hybrid Load Balancing Strategy in Cloud Computing", in Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference 2015, IEEE 2015.
- [5] Jinhua GU, Jinhua Hu, Tianhai Zhao, Guofei Sun "A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment" in Journals of Computer, Academy Publishers 2012.
- [6] Neha Gupta, Parminder Singh "Load Balancing Using Genetic Algorithm in Mobile Cloud Computing" in International Journal of Innovation in Engineering & Technology (IJIET) 2014.
- [7] Mayur S. Pilavare, Amish Desai "A Novel Approach towards Improving Performance of Load Balancing Using Genetic Algorithm in cloud Computing" in IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIECS. IEEE 2015
- [8] Zhao Li, Dong Yu-Min, Huang Chen-Yang, "A Study of Link Load Balancing based on Improved Genetic Algorithm", in 2013 Sixth International Symposium on Computational Intelligence & Design, IEEE 2013.
- [9] Chandrashekar S. Pawar, Rajnikant B Wagh, "Priority Based Dynamic Resource Allocation in Cloud Computing with Modified Waiting Queue", in International Conference on Intelligent Systems and Signal Processing, IEEE 2013.
- [10] K C Gouda, Radhika T V, Akshatha M, "Priority based Resource Allocation model for Cloud Computing", in International Journal of Science, Engineering and Technology Research, Vol. 2 issue 1, January 2013.
- [11] Mithra P B, P Mohammed Shameem, "A Novel Load Balancing Model for Overloaded Cloud Partition", in International Journal of Research in Engineering and Technology, Vol 3 Special issue 7 May-2014.
- [12] Melanie Mitchell, "An Introduction to Genetic Algorithm".
- [13] [doc.ic.ac.uk/~nd/surprise\\_96/journal/vol1/hmw/article1.html](http://doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html)
- [14] [buyya.com/papers/CloudAnalyst-AINA2010.pdf](http://buyya.com/papers/CloudAnalyst-AINA2010.pdf)

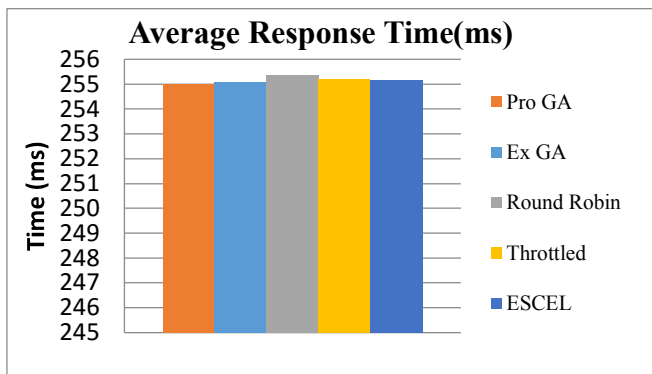


Fig 5: Performance Analysis of proposed method based on TABLE 2

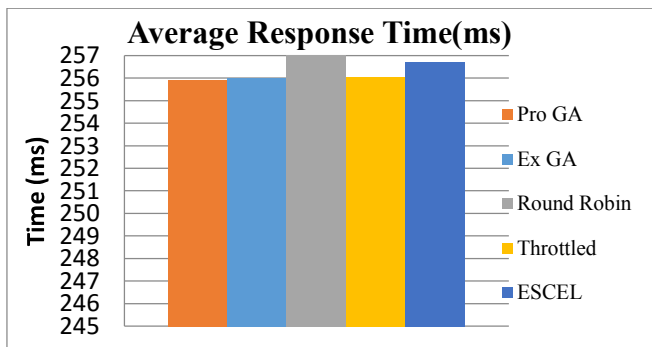


Fig 6: Performance Analysis of proposed method based on TABLE 3

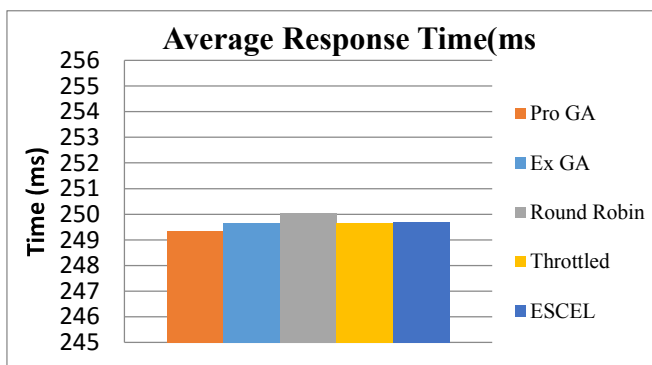


Fig 7: Performance Analysis of proposed method based on TABLE 4

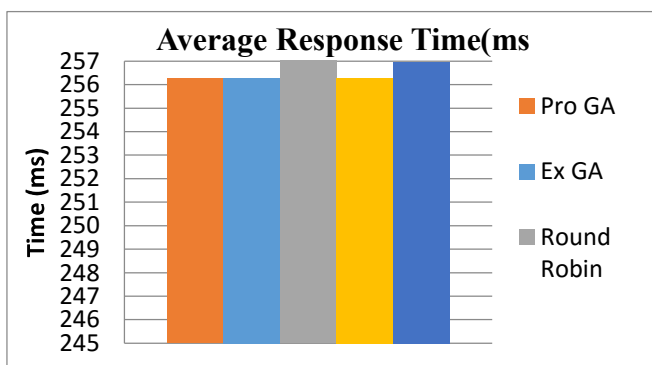


Fig 8: Performance Analysis of proposed method based on TABLE 5

