# Load Balancing Task Scheduling Based On Variance of Genetic Algorithm

Ayushi Harkawat, Shilpa Kumari, Poorva Pharkya, Deepak Garg
Masters of Computer Application ( M.C.A), National Institute Of Technology Kurukshetra, Kurukshetra, India.

**Abstract – In the intellection of cloud the foremost task is to balance the load and wield the task and it's scheduling with competent resource utilization. Since, users are increasing exponentially and on the other hand there's scarcity of resources, thus to suffice the requirements concept of cloud is used. By using the defined algorithm MPGA (Multi Population Genetic Algorithm) the above mentioned problems are solved up to some extent. But there are two more concepts that cannot be ignored while working with cloud and those are cost and time and it's concealed to some expanse with the help of algorithm MPGA as stated above. So, this paper holds HMPGA (Hybridization of MPGA) that avail the approach of algorithm of min-min for initializing the population and along with it also uses the intellection of hill climbing so that most fitted value can be found. HMPGA covers the need of least cost, balancing the node and also to minimize the makespan of nodes.**

**Index Terms – Genetic algorithm, Load balancing, Cloud computing, Task scheduling, Hill Climbing, HMPGA.**

## 1. INTRODUCTION

### 1.1 Cloud Computing

In layman terms cloud can be defined as the data that is accessed, delivered and stored on internet instead of hard drive of a computer. Cloud computing provides flexibility to use data and services from any part of world and without digging into the internal details of the processes or its methods. There are several companies that provide the facility of cloud to the users and which in drastically help users to save their data in encrypted form. Services provided by cloud computing are IaaS (Infrastructure as a Service), SaaS (Software as a Service), PaaS(Platform as a Service) and these services helps users to work efficiently on the services provided by defined companies. Types of cloud computing are categorized under three forms as private cloud, public cloud and hybrid cloud in which there are different level of rules or conditions applied before giving it to users. Paying little amount of money and having gain without any additional hardware appliances is the major advantage why companies prefer cloud.

### 1.2 Task Scheduling

Task scheduling is the most important function in the area of cloud computing. There are several algorithms to manage task that can be categorized as round robin algorithm and minimum link algorithm, both of these algorithm lack in some or the other way i.e. both are not preferred reason being the performance. Users working on cloud are in large number and so scheduling of task with proper utilization of resources is difficult and it should fulfill users need, so task scheduling algorithm should be efficient and reliable[1]

### 1.3 Load Balancing

Load balancing deals with dividing the task into two or more than two nodes so as to serve all the users on required time and users are accessed faster. Similarly, in cloud environment task and workloads are divided to bring poise between nodes. With the succor of balancing of load in cloud there can be good utilization of resources and reduction in cost can be done. In comparison with DNS (Domain Name System) load balancing and in case of balancing the load in cloud numerous services are provided by sundry companies based on computer networking. Load balancing methods can be categorized as static load balancing in this type the information of nodes performance is not used that is it will balance load amongst the given nodes and that to without calculating performance that is relative of the defined nodes, example of this kind can be algorithm of Round Robin and on the other hand dynamic load balancing technique the information of nodes performance is used as unlike the static one and also uses the concept of virtual servers, one of its example can be Least Connection.[3]

### 1.4 Genetic Algorithm

It is the algorithm that posse's heuristic method and is adaptive in nature along with this it incorporates the idea which is based on genetics plus the natural selection. Problem of optimization is resolved by exploiting it using the search method which is random. The most fitted value is selected from the individuals that are in sequent generation. Initial population is selected that best describes the problem and after selection of population it follows selection of thefittest survival with crossover that demands individuals mating and finally the mutation process of random changes.

## 2. RELATED WORK

### 2.1 JLGA

Genetic Algorithm is primarily applicable for finding the most fitted value in an optimize way and for this process different types of biological expansion are used. The term JLGA refers to the algorithm in which populations are initialized and

divided into different jobs or tasks in the anatomy of matrix where columns act as a numbers of distinctive task and the different forms of nodes acts in place of rows and for deputize these operation a algorithm is used which is called Greedy Algorithm. JLGA is used for finding the most optimal fitted value with approbation of balancing the Loads and Time.

Algorithm of JLGA

Input:

Maximum value: maximum iterations

S: Scale of the population; N: Different no of nodes

C1, C2: Job's average spanning C1 + C2 = 1

λ1, λ2 : Probability of fitnessvalue1 and fitnessvalue2, λ1 + λ2 = 1.

Output:

Elitevalue ,N : Ideal solution for the nodes

1: iterationvalue, λ ← 0; fitness1val←Ǿ

2: Elitevalue ,N , temporary1,N, point1, point2, Ps,N , Fitness1val1,S ← Ǿ

3: p ← Value of the Greedy Initialization

4:while iterationvalue<Maximumvalue do

5: λ=random(0,1)

6: if λ < λ2 then

7: fitness1val = Fitnessval2

8: else

9: fitness1val = Fitness1val1

10: end if

11: for i = 1 to S do

12: Fitnessvali ← fitnessval(i)

13: end for

14: Elitevalue1,N ← Ideal fitness find the value one by one

15: *p*1, *p*2 ← Selection of the Roulletewheels(*Ps,N)*

16: calculate pc, pm

17: if random(0, 1) < pc then

18: temporary1,N ← cossover(p1, p2)

19: end if

20: if random(0, 1) < pm then

21: P1,N ← mutation(temporary2)

22: end if

23: iterationvalue ← iterationvalue + 1

24:end while

2.1. b Fitness Function

The algorithm we study JLGA which is divide into two category fit1 and the next one is fit2.The first one is for balancing the inter-nodes and the second one is used for minimizing the time as much as possible and the outcomes is improved fitness. Then necessity of fitness function is for finding the discrete population in genetic algorithm. Fitness function plays an important role for determining the best fitted value in JLGA.

2.2 MPGA

MPGA perdure the procedure of global inquest mechanism ,the major benefits of using MPGA is to put together both the communication cost and also the loading time and for using these benefits in MPGA different types of algorithms are used min-min algorithm is one of the best algorithm for finding the best fitted value.[4]

2.2.a Algorithm of MPGA[4]

Input:

MaximumValue : maximum no of iteration

Scale: size of populations (scale); Node: numbers of Working node; Job: numbers of different jobs

*K11, K22*: Total and average job spanning *K11* + *K*2 2= 1

*P11*, *P22*: probability of fitness11 and fitness22

*P*11+ *P*22 = 1.

*iterate:iterators*

***C:cost11***

***T:time11***

Output:

*elite*11,*Node*: optimal result

1:*iterate11, P* ← 0; *fitnessfunction*← Ǿ

2:*elite*11,*Node* , *temp*22,*N11, I*11, *I*22, *Pointscale,N* , *Fitness*11,*Scales* ← Ǿ

3:***I11 ← initialization of min min for population***

4:while*iterate* <*MaximumIteration*do

5: *P11*=random(0,1)

6: if *P11 < P22* then

7: *fitnessfunction= Fitness22*

8: else

9: *fitnessfunction= Fitness*11

10: end if

11: for *jt*= 1 to *Scale* do

12: *Fitjt*← *fitnessfunction*(*jt*)

13: end for

14: *elite*11,*Node*← *fitnesstest individuals among all*

15: *I*11, *I*22 ← selection of roullete wheel(*Ps,N)*

16: *calculate pcrosses, pmutation*

17: if *random*(0, 1) <*pcrossover*then

18: *temp*22,*Node*← *crossover*(*I1, I2*)

19: end if

20: if *random*(0, 1) <*pmutation*then

21: *I*1,*Node*← *mutation*(*temp11*)

22: end if

23: **calculation are done *according to total time and the fitness value***

24: *iterate* ← *iterate* + 1

25:end while

## 3. OUR PROPOSED HMPGA

This paper presents, Hybridization of Multipopulation of Genetic Algorithm (HMPGA) for balancing the load as well as the communication cost in the parallel manner. This algorithm combines the admirable attributes of JLGA and MPGA forfinding the most optimal values for the nodes. Theexpeditiousgrowth of the computers boosts the interest and the importance ofresearches in this era. In this field balancing the loads and scheduling the tasks in the utmost component for providing the better fulfillment. There is a crucial algorithm can be used to solve the different problem which is occur on the nodes.

For more appropriate performance our efforts is to acquirement of the most optimal values, for that we work on the new approach which is based on hill climbing method for finding the peak value with the minimum time ,minimal cost and superlatively way for load balancing.

For Enhancement of the performance our accomplishment is make equal proportion on time, cost and the third factor is load. We also take some good feature of min-min algorithm and try to find the best fitted value for the optimization process.

Input:

MaximumIteration: max iterations

S: population of scale; N: Nodes numbers; J: Different Jobs

*C*1, *C*2: Mass of different averages of spanning for jobs and total planning of distinctive jobs, *C*1 + *C*2 = 1

$\lambda$1, $\lambda$2 : Probability of *fitnessvalue*1 *and fitnessvauel*2

$\lambda$1 + $\lambda$2 = 1.

***C:cost***

***T:time***

Output:

*elite*11,*N*: N is for Ideal solution

1: *iterations,* $\lambda$ ← 0; *fitnessvalue*← Ǿ

2:*elite*11, *N, temp*1, *N, p*1, *p*2, *Ps,N , Fitnessvalues*1,*S* ← Ǿ

3:***p*← *Min-Min initializations for all the population***

***Fitness2=*(Minimize completion time of jobs + Minimize all the loads per node)**

***Fitness1=(Minimizing the makespan + Minimizing the cost)***

4:while*iterations* <*Maximumvalue*do

5: $\lambda$=random(0,1)

6: if $\lambda$ < $\lambda$2 then

7:*fitnessvalue*= ***Fitnessvalue*2** (Minimizing Jobs Completion time + minimizing loading per nodes)

8: else

9:*fitnessvalue*= ***Fitnessvalue*1** (Minimizing Makespans + minimizing costs)

10: end if

11: for *i* = 1 to *S* do

12: *Fitnessvaluei*← *fitnessvalue*(*i*)

13: end for

14: *elite*11,*N*← *one by one finding fitnessvalue*

15: *p*11, *p*22 ← Roullete wheel selections(*Ps,N)*

***Optimalmate1=hill climbing method (p11)***

***Optimalmate2=hill climbing method (p22)***

16: *calculate prcrossover, prmutation*

17: if *random*(0, 1) <*prcrossover*then

18: *temp*orary1,*N*← *cossovers*(*optimalmate1*, *optimalmate*2)

19: end if

20: if *random*(0, 1) <*pmutation*then

21: *P*1, *N*← *mutations* (*temporary1*)

22: end if

23: *iterations* ← *iterations* + 1

24: end while

Result 1: This graph show the balancing of nodes

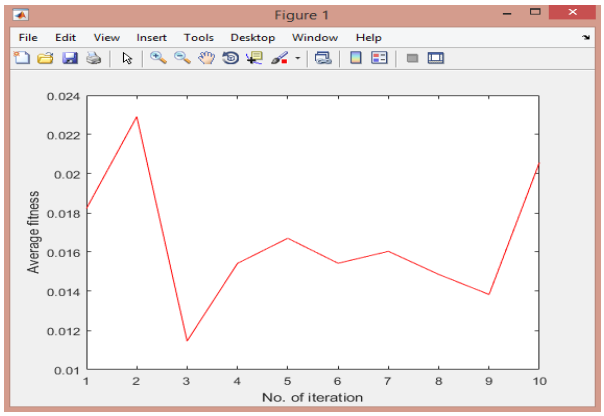


Fig 1

Result 2:This graph show the balancing of cost

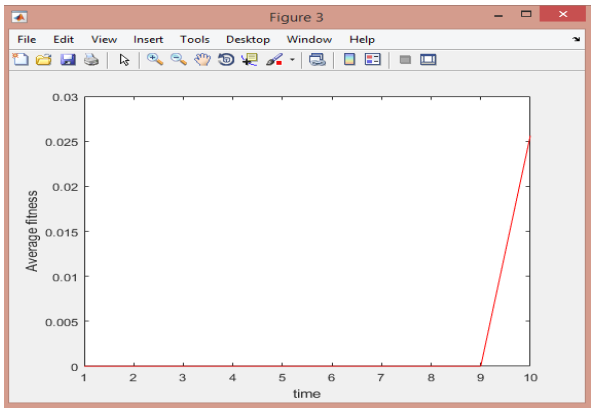


Fig 2

## 4. CONCLUSION

According to the different inspection, in this paper we evaluate distinctive solutions and then examine all of them in consideration of drawbacks. To develop more appropriate algorithm, researches are cause discrepant factors. We founded that in the superior study any algorithm are not work on the both communication cost and the load in well-balanced forms for jobs these drawbacks are considerable .For balancing the cost as well as load in together is the biggest and ultimate solution for the selection of execution of jobs.

## 5. FUTURE WORK

In future, we can work on priorities of jobs scheduling and also enhance the performance by dynamically load balancing of the nodes, for balancing the nodes dynamically virtual machines are required at the running time of the jobs. So that , jobs are automatically re submitted into the nodes.

## REFERENCES

[1] Wang T., Liu Z., Chen Y., Xu Y.: Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing. In: 12th International Conference on Dependable, Autonomic and Secure Computing, IEEE Computer Society, pp. 146--152 (2014).

[2] Kruekaew B. and Kimpan W.: Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony. In: Proceedings of the International Multi Conference of Engineers and Computer Scientists , vol. 1, pp. 185--189 (2014).

[3] Liu X., Li K., Min G., Xiao B. ,Shen Y. and Qu W.: Efficient unknown tag identification protocols in large-scale rfid systems, In: Parallel and Distributed Systems,IEEE Transaction ,vol. 85, pp. 190--197 (2014).

[4] Bei W., Jun L.:Load Balancing Task Scheduling based on Multi Population Genetic Algorithm in Cloud Computing. In: 35thChinese Control Conference, IEEE Xplore, pp. 5261--5266 (2016).

[5] Ramezani F., Lu J., and HussainF.:Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization. In: International Journal Parallel Programming, vol. 42, pp. 739--754 (2013).