Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 9, September 2015, pg.199 – 207

RESEARCH ARTICLE

Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm (FUZZY LOGIC)

[1]Sunny Dahiya, [2]Preety
[1,2] Department of Computer Science
[1,2]International Institute of Technology and Business
[1,2]Jhundpur, Sonipat, Haryana, 131023, India
[1] dahiya.sny@gmail.com, [2] preeti.gulia11@gmail.com

Abstract: Cloud environment provides the distributed services available to multiple clients. These client access services are required to share in a resource effective way. In this work, a improved genetic based scheduling model is presented for cloud system scheduling. The presented work will be divided in two main stages. In first stage, the ordering of virtual machines will be done. This scheduling will be applied using fuzzy logic applied on the strengthen parameters. These parameters include memory, performance and the resource availability. . Once the requirements are given the cloud interface, the selection of the most effective VMs will be identified under the defined vectors. To perform the service to VM matching, weighted cost evaluation approach is suggested in this work. This cost evaluation will be done in two layers. In first layer, the evaluation cost will depend on the priority sequence decided by the user to the main decision vectors. After defining the first level allocation, the second stage is applied on client side. This stage is applied using fuzzy integrated genetic approach. The work will be implemented in matlab environment. The work is about to reduce the wait time and the allocation cost of different services

Keywords: Cloud computing, Task Scheduling, Genetic Algorithm (GA), Modified Genetic Algorithm (MGA), Fuzzy Logic.

1. INTRODUCTION

Vaquero et al. [1] have stated "clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited

by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized Service Level Agreements."

There are three types of cloud computing services : Platform as a Service (PaaS), Infrastructure as a service (IaaS) and Software as a Service (SaaS)

Scheduling is considered as the arrangement of the requests performed by different users in an organized and defined form. This form is defined under the real time scenario and able to improve the system capability. The cloud system defined under various process and memory based constraints is able to provide the generalized system allocation to the environment. There are number of scheduling and queuing approaches so that the multiprogramming based shared system is provided[2][3]

A number of metaheuristic scheduling algorithms is used for task scheduling in cloud environment like genetic algorithm (GA), simulated annealing (SA), ant colony optimization (ACO) and particle swarm optimization (PSO)[4], out of which genetic algorithm (GA) has been proved quite effective. These all algorithms are derived from natural phenomena of biology. Genetic algorithm is applicable in different domains like in robotics, machine learning, circuit designing, time-series forecasting, auto sensing, signal processing, game playing etc [5, 6]. The main advantage of GA is that it has the great ability of handling multi objective problems and can be easily interfaced with existing simulations and models. It can explore big search space so it does not impose any fixed length of the solution for a particular problem.

2. LITERATURE REVIEW

There are number of scheduling and resource allocation approaches used by different cloud servers. Some of the scheduling scheme were based on the cost analysis and some gives the concept of job forwarding and job migration. In this chapter some of the work done by earlier researchers is discussed.

VC-MIGRATION: actually defines the formation of clusters in cloud server. These kind of server system also includes the communication migration along with the virtual cluster specification. The framework is defined to achieve the control mechanism for virtual cluster. The experimentation is here provided to achieve the cluster formation, cluster scheduling, mutation processing, concurrent processing, process migration, virtual system integration, virtual cluster effective scheduled process integration[7].

Another work on application system migration for cloud servers is provided as the middle layer environment. Author has defined the reconfiguration capabilities to improve the scalability and elasticity so that the system work load will be distributed and the reconfiguration will be achieved for the system. Such system also provides the ideal reconfiguration so that the VM migration and application level processing will be done. Author defined the ideal reconfiguration processing will be obtained. Author defined the performance integration and reconfiguration to improve the system accuracy and processing [8].

Christina Delimitrou (2013)defined a scheduler to achieve the heterogeneous communication and interference so that the adaptive cloud environment will be generated. Author defined a paragon under the analytical study over the system so that information distribution and scheduling will be performed effectively. Author defined a collaborative filtration approach for accurate classification of data and to reduce the workload over the system so that heterogeneity to the system will be achieved. Author defined the interference analysis over the scheduled system (Delimitrou, 2013)

3. PROPOSED WORK

Cloud environment provides the distributed services available to multiple clients. These client access services are required to share in an resource effective way. In this work, a improved genetic based scheduling model is presented for cloud system scheduling. The presented work will be divided in two main stages. In first stage, the ordering of virtual machines will be done. This scheduling will be applied using fuzzy logic applied on the strengthen parameters. These parameters includes memory, performance and the resource availability. . Once the requirements are given the cloud interface, the selection of the most effective VMs will be identified under the defined vectors. To perform the service to VM matching, weighted cost evaluation approach is suggested in this work. This cost evaluation will be done in two layers. In first layer, the evaluation cost will depend on the priority sequence decided by the user to the main decision vectors. After defining the first level allocation, the second stage is applied on client side. This stage is applied using fuzzy integrated genetic approach. The work will be implemented in matlab

environment. The work is about to reduce the wait time and the allocation cost of different services. Figure 1 shows the different stages of the work.

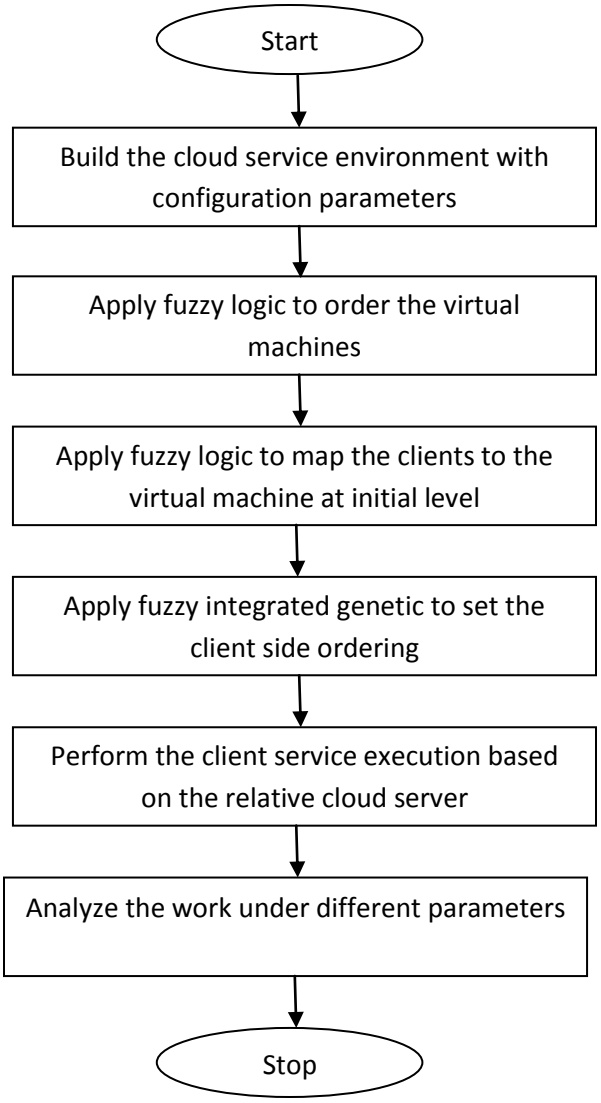

Figure 1. Flow of Standard Genetic Algorithm

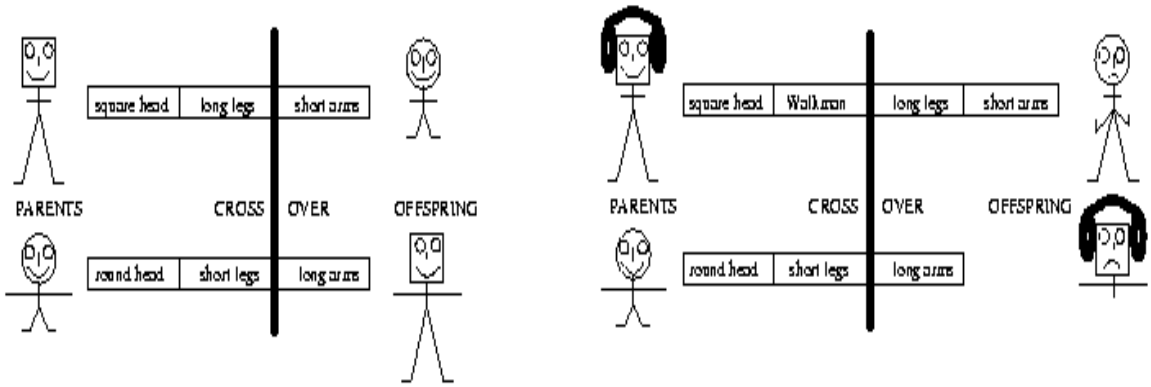

Figure 3.2 - The Crossover Operator

a) GENETICS

i) Initial population:

The major inclusion in this work model is of genetic process. The genetic process requires the population set to generate the optimized process sequence. To apply the genetic at first possible sequences are generated for N number of user requests. These sequences collectively formed the population set. The sequence is here formed in the form of matrix. The scheduling sequences permutations are formed to generate the genetic population set.

ii) Fitness Rule:

The next step associated with the work is to specification of fitness rule. The fitness rule basically decides the constraints respective to which the genetic process is formed. The fitness rule is applied on the selected population sequence and to identify the valid sequence. The valid sequence observation is obtained under the cost vector. If the cost of the sequence is acceptable, it is considered as the fitness rule. The cost vector is here dependent on the time of process execution. No two processes should be executed at same time. The process must be executed before the dead line. The minimization process is applied for cost optimized sequence selection.

iii) Ranked Selection Function:

In this work, a ranked method is applied to generate the optimized sequence. The sequence ordering is here applied on individuals under the fitness rule so that the population element will be considered for the sequence set. The probability driven method is here applied to generate the optimized sequence. The population set specification is here given by the ranked method. This method is based on the best and worst case analysis. This probabilistic estimation is given as

$pi = 1/N *\{pworst + (pbest – pworst) * [(I-1)/(N-1)]\}$

Here N => Size of Population vector

p_{worst} => Worst probability of sequence selection

p_{best} => Best probability of sequence selection

This selection process is here applied in ranked form so that the optimized solution will be derived.

iv) Crossover:

The next process applied here is to apply crossover operation on selection parents. Here figure 3.2 is showing the process applied by crossover operation. According to this process a new child is generated.

In this work, a distance preserving cross over is applied. The distance is here estimated from the cost vector and the minimum cost sequence combination is considered as next child.

v) Mutation:

Unlike normal GAs where recombination is the primary operator, SAGA uses mutation on a converged population. This is in effect hill-crawling rather than hill-climbing. The crossover operator is used as a protection against Muller's Ratchet – the effect where a mutation rate set too high causes loss of the current optimal value. Recombination means that the current optimum will not be lost because the genome can be 'repaired' by crossing two mutated genomes.

b) ALGORITHM

Crossover(P1,p2)

/*P1 and P2 are two Parent Sequence of client request on which the crossover procedure is implemented*/

{

1. N=Length(P1)

   [Find Length of client request sequence]

2. For i=1 to N

   [Process the client request Sequence to generate child Sequence]

   {

3. If(P1(i).Equals(P2(i))

   {

   Set Child(i)=P1(i)

   }

   [Copy the parent data elements to child]

4. For I=1 to Length(P1)

   {

   For j=1 to Length(P2)

   {

5. If(P1(i).Equals(P2(j)))

   {

   Child(k)=P1(i)

   }

   }

6. For i=1 to N

   [Process Remaining Cities of P1 & P2]

   {

7. Analyze the P1 (i) & P2(i) under Fuzzy Parameters

8. If(High(P1(i))

   {

   Child(j)=P1(i)

   }

   Else

   {

Child(j)=P2(i)

}

}

9. Else If(Medium (P1(i))

{

Child(j)=P1(i)

}

Else

{

Child(j)=P2(i)

}

}

10. Return Child;

}

4. EXPERIMENTAL RESULTS

4.1 Tool

4.1.1 Matlab

Matlab is used here as the simulation language that provide the interactive environment to provide quick and accurate results. Matlab is a product of Mathworks that includes the scientific software package integrated to provide the mathematical computation and provide graphical visualization so that result representation can be improved. For processing, Maltab is having vast range of function to provide the effective information processing. Matlab provides the integration in the form

4.2 Genetic Toolbox

Genetic is the integrated tool available in matlab environment. This tool can be applied for any application in two ways

It can be run from command line using ga command

It can be applied as the algorithmic tool in graphical environment.

4.3 Architecture Results

To apply the cloud scheduling, the first work is to setup the server and client side parameters. The cloud environment configuration is required on server side with relative resource constraint specification. The parameters considered in the experimentation is given here under

Table 4.1 : Experimentation Parameters

Parameter	Values
Number of Clouds	5
Number of Virtual Machines	10
Load on Machine	5
IO Limit	5
Memory Limit	64M
Processing Limit	1000ms
Simulation Time	100 sec

These constraints are setup at code level. As the process is formed, the algorithmic framework is formulated. At first, the cloud capabilities analysis is performed and based on these observations, the virtual machine ordering is done. In second stage, the clients are processed to setup the execution order for clients.
The scheduled cost results are shown in figure 4.1

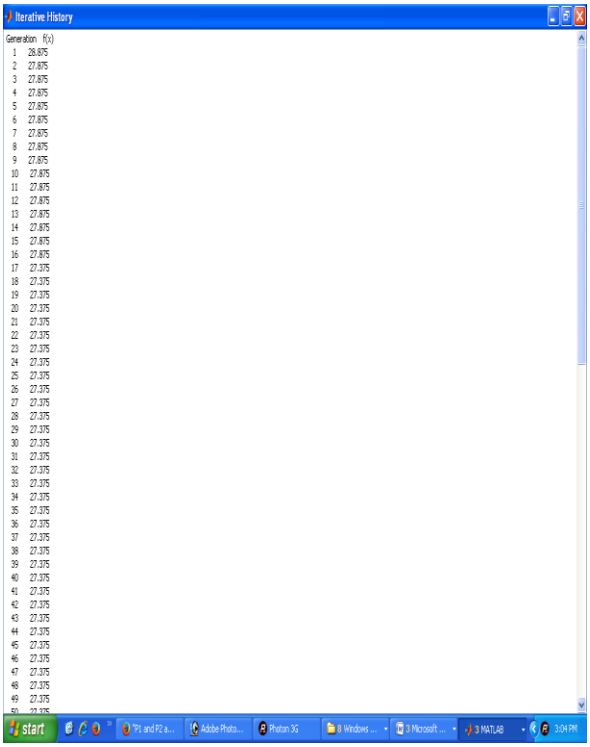

Figure 4.1 : Scheduled Cost

Here figure 4.1 is showing the scheduled cost obtained after each iteration of genetic process. This scheduling is applied to obtain the optimized cost effective sequence. This first time allocation process is shown in table 4.2

Cloud	VM	Request
2	0	10
2	0	22
2	0	25
1	2	19
1	2	29
3	4	16
3	4	23

Table 4.2 : First Level Allocation

After this allocation, the genetic adaptive scheduling process is applied. For this the scheduling sequence pool is formed. The cost adaptive results are obtained from the allocation process shown in figure 4.2

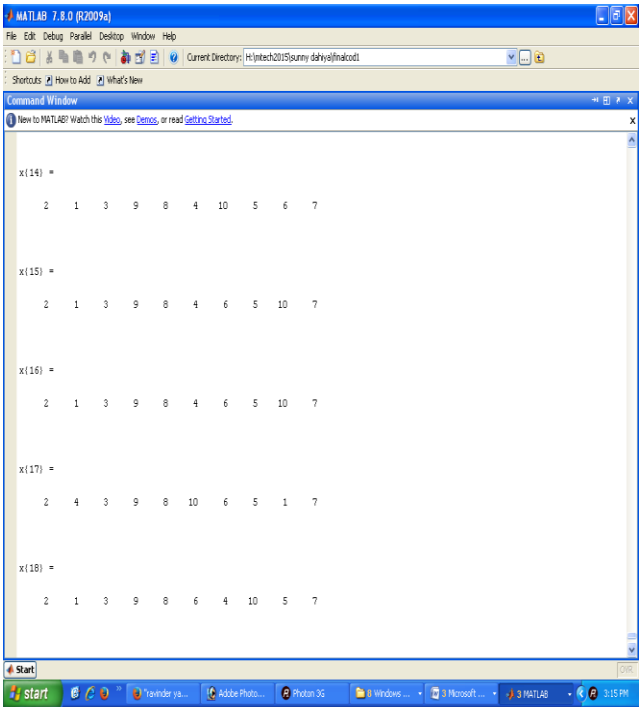

Figure 4.2 : Scheduling Sequence Results

Here figure 4.2 is showing the results obtained for the scheduling sequence. As the genetic process is applied, the ranked selection is applied on the pool and the generation of the next child is done using cross over operator. The optimized generated sequence is showing after each iteration is shown in fig.

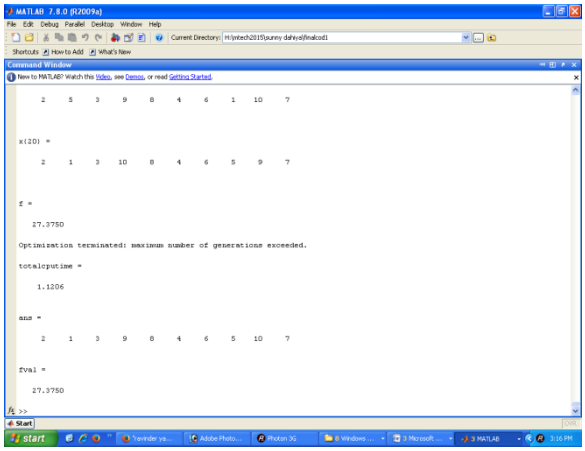

Figure 4.3 : Cost Effective Optimize Scheduled Sequence

Here figure 4.3 is showing the final outcome derived after fuzzy integrated genetic process. The figure here shows the generated optimized sequence along with cost and time vector taken to obtain the output.

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusion:

In first layer of this model, the cloud side processing is performed. On server side, the analysis is performed to analyze the cloud system capabilities and arrange the virtual machines in capability order. In second stage of this

model, the client requests are processed and arranged in order of their requirement. To perform scheduling, the first sub stage is to perform initial allocation of client requests to particular virtual machine. This allocation is done based on the requirement and availability mapping. After this allocation, the possible sequences are obtained and formed a sequence pool which is used by the genetic process model for scheduling.

In genetic process, the ranked sequence selection is performed under cost adaptive fitness rule. Later on the fuzzy integrated DPX method is applied to generate next possible sequence. This process is repeated till the cost effective sequence solution is not obtained. The presented work is implemented in matlab environment. The experimentation results shows that the work has provided the cost effective results.

5.2 Future Work:

In this work, a fuzzy integrated genetic model is presented for cloud request scheduling so that the maximum utilization of resources will be done and the cost effective solution will be formed. The work can be improved in future under following aspects.

1. The presented work is defined the overload conditions in terms of deadline as well as the memory limit of the Clouds. In future some other parameters can also be taken to decide the migration condition.
2. The presented work is defined for the public Cloud environment, but in future, the work can be extended to private and the hybrid Cloud environment.

References

[1] L.M. Vaquero, a Break in the Clouds: Towards a Cloud Definition, SIGCOMM Somputer Communication Review, 39:50-55, 2009.

[2] Xun Xu, from cloud computing to cloud manufacturing, Robo. Comput. Integr. Manuf. , 28 , pp. 75-86,2011

[3] Omer Khalid, Ivo Maljevic, Richard Anthony, Miltos Petridis, Kevin Parrott, Markus Schulz, Deadline Aware Virtual Machine Scheduler for Grid and Cloud Computing, International Conference on Advanced Information Networking and Applications Workshops, pp 85-90, 2010.

[4] W. Meihong, and Z. Wenhua, "A comparison of four popular heuristics for task scheduling problem in computational grid.", 6th International Conference on Wireless Communications Networking and Mobile Computing, IEEE, 2010.

[5] M. Dianati, I. Song, and M. Treiber. "An introduction to genetic algorithms and evolution strategies." Technical report, University of Waterloo, Ontario, N2L 3G1, Canada, 2002.

[6] K. D. Joshi, and A. A. Pandya. "Genetic algorithms and their applications–traveling sales person and antenna design." pp. 1-5, 2003.

[7] Kejiang Ye, Xiaohong Jiang, VC-Migration: Live Migration of Virtual Clusters in the Cloud, ACM/IEEE 13th International Conference on Grid Computing, pp 11-17, 2012

[8] Shigeru Imai, Thomas Chestna, Carlos A. Varela, Elastic Scalable Cloud Computing Using Application-Level Migration, IEEE/ACM Fifth International Conference on Utility and Cloud Computing, pp 91-98, 2012

