International Journal of Advanced Science and Research
ISSN: 2455-4227
Impact Factor: RJIF 5.12
www.allsciencejournal.com
Volume 3; Issue 2; March 2018; Page No. 110-115

Cost-based multi-QoS job scheduling algorithm using genetic approach in cloud computing environment

Ratin Gautam[1], Shakti Arora[2]
[1] Student, Computer Science and Engineering, Panipat Institute of Engineering and Technology, Panipat, Haryana, India
[2] Computer Science and Engineering, Panipat Institute of Engineering and Technology Panipat, Haryana, India

Abstract
One of the best methods for cloud scheduling is the genetic algorithm (GA). The simple and parallel features of this algorithm make it applicable to several optimization problems. A GA searches the problem space globally and is unable to search locally. In the proposed model, the task scheduler calls the GA scheduling function every task scheduling [4] cycle. This function creates a set of task schedules and evaluates the quality of each task schedule with user satisfaction and virtual machine [6] availability. The function iterates genetic operations to make an optimal task schedule. In the presented work, task scheduling is done to reduce the total cost of task processing (on the processing units of the cloud) for the cloud provider by reducing the execution time and hence the delay cost or penalty cost.

Keywords: cloud computing, genetic algorithm, scheduling, task scheduling

1. Introduction
"Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." The beauty of cloud computing is that another company hosts your application. This means that they handle the costs of servers, they manage the software updates and depending on how you craft your contract (Service level agreement); you pay less for the service. There are six phases of computing paradigms, from dummy terminals/mainframes, to PCs, networking computing, to grid and cloud computing. In phase 1, many users shared powerful mainframes using dummy terminals. In phase 2, stand-alone PCs became powerful enough to meet the majority of users' needs. In phase 3, PCs, laptops, and servers were connected together through local networks to share resources and increase performance. In phase 4, local networks were connected to other local networks forming a global network such as the Internet to utilize remote applications and resources. In phase 5, grid computing provided shared computing power and storage through a distributed computing system. In phase 6, cloud computing further provides shared resources on the internet, in a scalable and simple way.

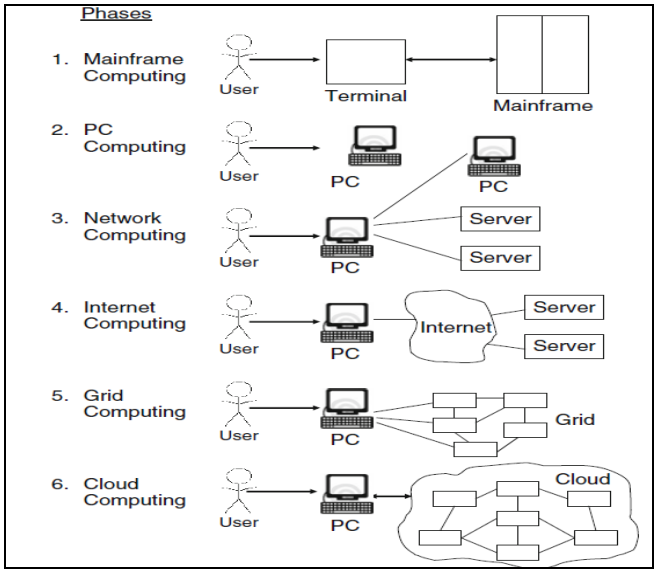

Fig 1: Six phases of computing paradigm

Cloud computing resources can be procured and disposed of by the consumer without human interaction with the cloud service provider. This automated process reduces the personnel overhead of the cloud provider, cutting costs and lowering the price at which the services can be offered. Cloud computing enables the measuring of used resources, as is the case in utility computing. The measurements can be used to provide resource efficiency information to the cloud provider, and can be used to provide the consumer a payment model based on "pay-per-use." For example, the consumer may be billed for the data transfer volumes, the number of hours a service is running, or the volume of the data stored per month.
Clouds can be classified in terms of who owns and manages the cloud; a common distinction is Public Clouds, Private Clouds, Hybrid Clouds and Community Clouds. A public cloud, or external cloud, is the most common form of cloud computing, in which services are made available to the general public in a pay-as-you-go manner. A Private Cloud, or internal cloud, is used when the cloud infrastructure, proprietary network or data centre, is operated solely for a business or organization, and serves customers within the business fire-wall. Most of the private clouds are large company or government departments who prefer to keep their data in a more controlled and secure environment. Here presents a comparison between public and private clouds.

Table 1: Differences between public cloud and private cloud

	Public cloud	Private cloud
Infrastructure Owner	Third party (Cloud provider)	Enterprise
Scalability	Unlimited and On-Demand	Limited to the installed Infrastructure
Control and Management	Only manipulate the virtual machines, resulting in less management burden	High level of control over the resources, and need more expertise to mange them.
Cost	Lower cost	High cost including: space, cooling, energy consumption and hardware cost
Performance	Unpredictable multi-tenant environment makes it hard to achieve guaranteed performance	Guaranteed performance
Security	Concerns regarding data privacy	Highly secure

2. Proposed Definition

Over the preceding chapters the notion of task scheduling within the cloud is discussed. In cloud computing the number of user requests/tasks with different QoS requirements submitted dynamically. Now to make a most optimal schedule of these tasks in the cloud environment where the no. of processing units with different MIPS, different load and capacity is a challenge and actually a NP- complete problem. And to conclude which factors should be considered for calculating the optimal schedule and how much these factors affect the processing time, hence the total cost is another challenge.

a. Motivation

- The main reason behind task scheduling is to reduce the total cost of task processing (on the processing units of the cloud) for the cloud provider by reducing the execution time and hence the delay cost or penalty cost. By reducing the total execution time, task will be completed before deadline as mention in service-level agreement (SLA), results in less SLA violations and more profit.
- One more reason behind this issue is to reduce the no. of migrations between the processing units as these migrations will increase the execution time and bandwidth latency due to more traffic intensity in the network. So a better schedule will decrease the bandwidth latency that is another big issue in the cloud where every task is online and need high bandwidth.

b. Objectives

The presented work will cover the following research objectives

1. Design the cloud environment with some pre-assumed parameters
2. Define the factors deciding how optimal a schedule is.
3. Design and Implementation of new parametric task scheduling using genetic algorithm meta-heuristic
4. Comparative analysis of approach
5. Highly authenticated and secure Cloud environment
6. Request scheduling of the users accessing the request on the Cloud
7. Throughput time should be less
8. Analysis of result

3. The proposed algorithm

Cost based Multi-QoS Job Scheduling [2] using Genetic-Algorithm Approach in Cloud Computing Environment.
Job scheduling is an NP-complete problem, and the genetic algorithm based on evolutionary theory is very suitable for complex and volatile optimization problems [1]. The proposed algorithm has the following steps:

- Create the cloud environment.
- Define the fitness function.
- Define the encoding scheme of chromosomes (schedule).
- Calculate the cost of each task execution on each processing unit (many to many relationship) and randomly generate the initial population.
- Define the selection parameter for reproduction of next generation.
- Define the crossover parameters
- Define the mutation parameters
- And at last we define replacement and termination policy

a. Cloud Environment

When designing a solution to make a schedule of tasks, deciding which task execute on which processing unit, one need to take into account the following salient points.

- In this work, we create a cloud consisting of fixed no. of processing units. Data enter manager manages the each processing unit by creating a PUAV (processing unit attribute vector) defining MIPS (Millions of Instructions per Second), no of virtual machines (VM) created.
- User submits its task with predefined type and quality of service (QoS) requirements in SLA.
- A pre-processing unit creates the UJAV (user job attribute vector) for each job/task where a UJAV includes arrival time, expected instruction count, deadline, type of service (IaaS, PaaS, SaaS). And classifies the submitted tasks based on the UJAV. Classified tasks are submitted into

different queues for scheduling.
- Now the job/task scheduler does the optimal mapping of jobs on processing units with the help of genetic [3] algorithm that guarantees the optimal solution but not the best solution. It takes the input from the pre-processing unit (classified tasks) and datacenter manager (PUAV vector).

b. Fitness Function

In this algorithm, fitness function (f) is used to calculate how good a schedule is on the basis of following parameters:
- **Processing time(PT):** It is approximate time a job required to complete its work on a processing unit and calculated as
  PT = MIPS of processing unit on which job will be executed * Expected Instruction Count of job submitted
- **Arrival time (AT):** It is the time when job is submitted by the user in the cloud.
- **Deadline (DL):** It is the time when job is expected to be completed without any penalty cost or delay cost.
- **Input/Output requirements (IO):** It is calculated by the preprocessing unit on the basis of type of job defined by user in SLA.
- **Memory requirements (M/M):** It is also calculated by the preprocessing unit on the basis of type of job defined by user in SLA.
- **Load (L):** It is calculated as the no. of vms already allocated on a processing unit
- **Flexibility Factor (FF):** It is the factor that decides how much more time cloud provider can take to start the job (after submission in to cloud) so that it finishes before deadline.

Therefore,

FF = DL-AT-PT;

Fitness Function (f): (L * a + FF * b + IO * c + M/M * d)
where
a, b, c and d are the variables used to give the weightage to these factors such that a+b+c+d =1.

4. Encoding scheme and initial population generation

In genetic [5], we encode each possible solution (called chromosomes) in a possible format either in binary, decimal, hexa etc. In this scheduling algorithm, we have to map number of jobs with number of machines. That's why
- **Step 1:** We create a linear array A [1: n], where n denote the no. of jobs. Therefore job is the index of array
- **Step 2:** And assign each index a value which denotes the processing unit on which that job will be executed

For example: let we have 9 processing units and 10 jobs are in the input queue of the scheduler then let one of the possible solution is

3 6 9 1 2 4 3 5 7 8

It means 1st job is scheduled on processing unit 3, 2nd job is scheduled on processing unit and so on.
Now we generate all possible chromosomes by using this encoding scheme to create initial population P0.

5. Selection Procedure

Before applying crossover and mutation operator, we have to select the parent chromosomes from the current generation. In this algorithm we use elitist method of selection. According to this, we select the best two chromosomes for crossover (p1, p2) and a random chromosome for mutation operator, denoted as p3.

6. Crossover Operator

Crossover operator is used for reproduction procedure. In this algorithm we use cyclic crossover method for generating children (c1, c2) from parent chromosomes (p1, p2).
- **Step 1:** find the cycle starting from 1st gene of p1 then the first gene of p2 then corresponding element in p1 and so on upto we get the same element in p2 as 1st gene of p1
- **Step 2:** copy the elements in the cycle to the child c1 (c2) with the corresponding position of parent p1 (p2).
- **Step 3:** Delete the elements of cycle from p1(p2) for determining the remaining elements
- **Step 4:** fill the child c1 (c2) with the remaining element from p2 (p1), at corresponding positions.

7. Mutation Operator

Mutation operator is also a reproduction operator, used for generating more diversity in next generation. Here we use swap mutation on a chromosome, let it be p3
- **Step 1:** take two random mutation point n and m.
- **Step 2:** swap p3[n] and p3[m] to produce c3

8. Simulation and Results

In simulation we evaluated the performance of improved algorithm in terms of arrival time, deadline, processing time, number of input-output requests, number of memory requests and load in comparison with existing algorithm.

a. Simulation Tool

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and FORTRAN. You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas. MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms.

b. Matlab Features

High-level language for technical computing
- Development environment for managing code, files, and data

- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating Matlab based algorithms with external applications and Languages, such as C, C++, Fortran, Java™, COM, and Microsoft® Excel®

Matlab is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.
Matlab is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fortran. The name Matlab stands for matrix laboratory. Matlab was originally written to provide easy access to matrix software developed by the Linpack and Eispack projects, which together represent the state-of-the-art in software for matrix computation.
Matlab has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, Matlab is the tool of choice for high-productivity research, development, and analysis. Matlab is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fortran.
Matlab features a family of application-specific solutions called toolboxes. Very important to most users of Matlab, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of Matlab functions (M-files) that extend the Matlab environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.
Matlab has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, Matlab is the tool of choice for high-productivity research, development, and analysis. Matlab is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fortran.
Matlab features a family of application-specific solutions called toolboxes. Very important to most users of Matlab, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of Matlab functions (M-files) that extend the Matlab environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

c. Simulation

Task/Job Scheduling related to the reducing total cost, total execution time, bandwidth latency and SLA violations. The existing algorithm takes only the processing time as the only factor affecting the total cost. The proposed algorithm works on all these factors simultaneously. The modified algorithm with the following improvements:

- It gives a schedule that considers the load on a processing unit as an important criterion so that no. of migrations due to load balancing reduced.
- It gives a schedule that not consider the processing time but flexibility factor (arrival time – deadline – processing time), defines for how much time the delay of task can be afforded by cloud provider because after that delay cost or penalty cost is also added with the total cost of processing a task on the processing unit.

d. Simulation Scenario

Table 1: Simulation Parameters

S. No	Parameter	Values
1.	No of CPUs	10
2.	No of users	Varies dynamically
3.	MIPS	Fixed and Assigned to each PU already
4.	Arrival time	Randomly generated
5.	Deadline	Randomly generated
6.	Size(expected IC)	Randomly generated
7.	IP request	Randomly generated
8.	Mem request	Randomly generated
9.	Type of process	1. IaaS 2. PaaS 3. SaaS
10.	vms(for each PU)	Fixed
11.	Load (for each PU)	Vary dynamically from 0 to no of VMS
12.	Generations(no of iterations)	100

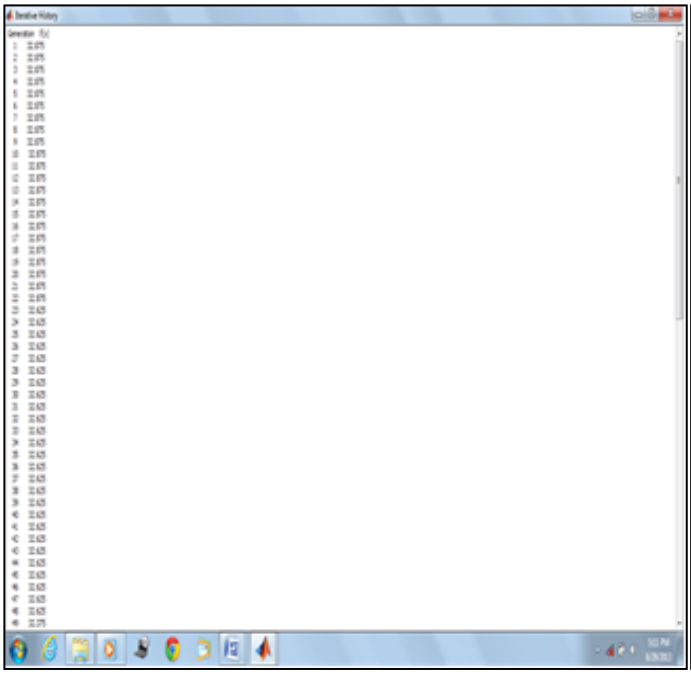

Fig 2: Best schedule cost in first 50 generations

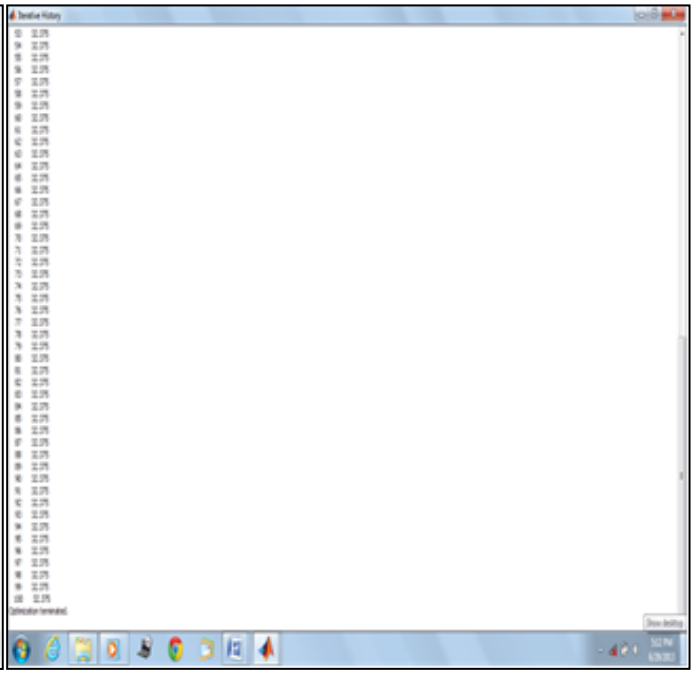

Fig 3: Best schedule in next 50 generations

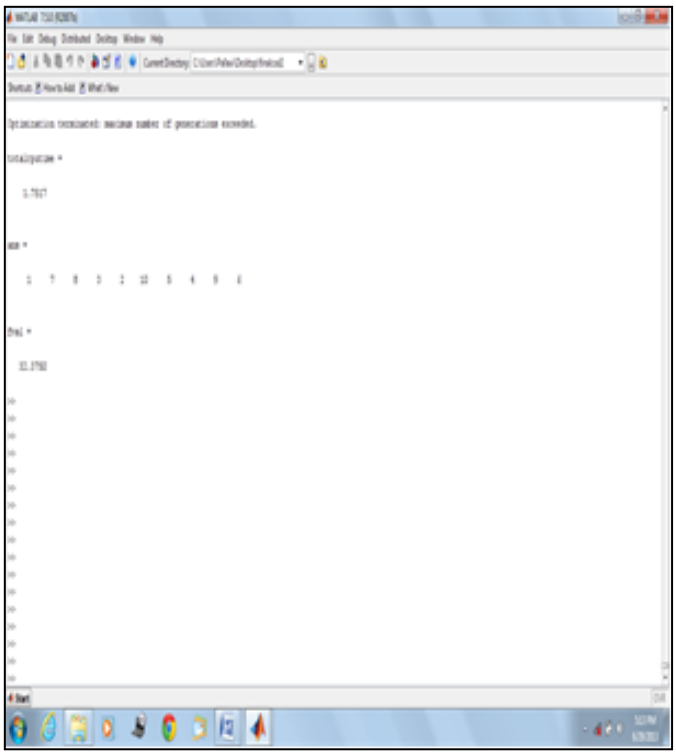

Fig 4: Optimal schedule and its cost

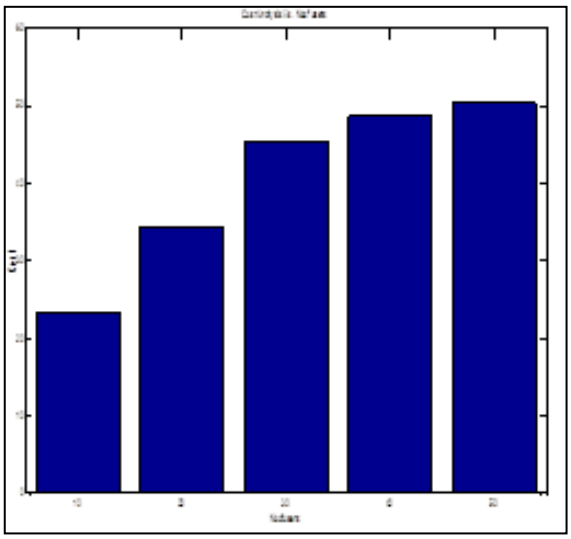

Fig 5: Cost analysis with no. of users

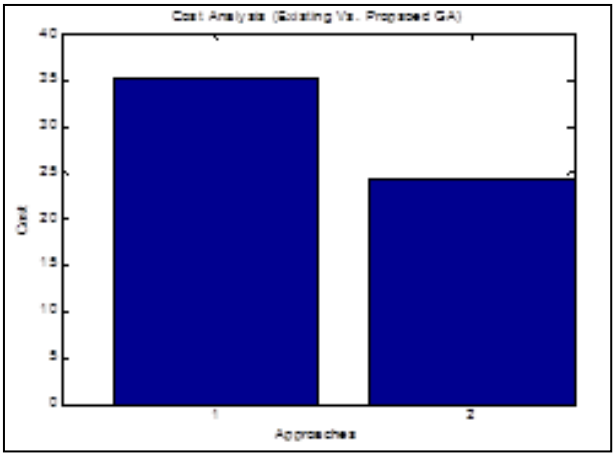

Fig 6: Comparative analysis on the basis of cost

Here in figure 2 and 3 we display the best schedule in each of the 100 generations. We can see that after 49th generation we get the same resulted cost in the consecutive generations. Now we calculate how the cost of execution increases as the number of users increases with the help of bar graph in the figure 5. In figure 6, we do the comparative analysis of existing and proposed algorithm on the basis of cost. In the figure 7, we do the comparative analysis of existing and proposed algorithm on the basis of no of iterations.

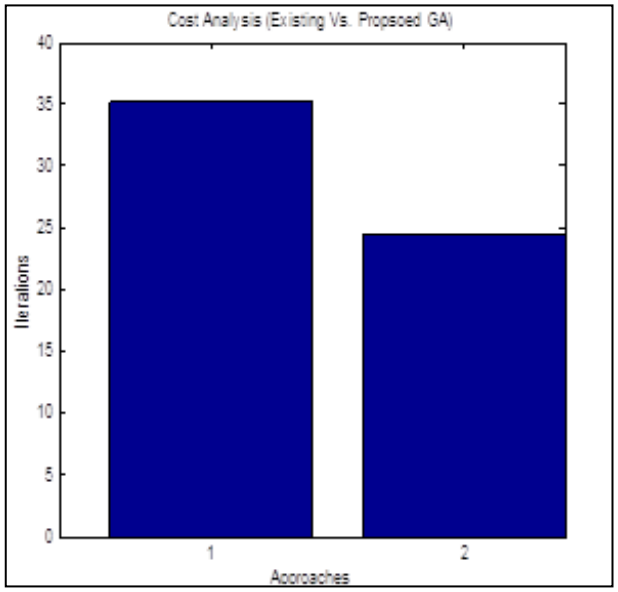

Fig 7: Comparative analysis on the basis of no. of iterations

9. Conclusion and Future work

Assigning the resources in the cloud is the most critical issue as more effectively we assign the resources, more we get the throughput. That's why efficient scheduling of tasks is required so that resources are utilized properly. In this work, we try to make a schedule of task – processing unit mapping such that penalty cost of execution after deadline can be reduced and load on the processing units get balanced and hence reduces the number of migrations & there cost. As the tasks are submitted in the cloud, they are pre-processed to classify them according to the type of service and then scheduled with the help of genetic algorithm which assures the optimal schedule but not the best schedule.

In the present work, we try to keep all the factors affecting the scheduling of tasks that is less processing time, less bandwidth latency, reduced total cost and less SLA violations. More work done in future on it is:

As genetic algorithm is used to optimize the schedule, it is slow in execution, so we have to improve the time wasted in the execution of genetic algorithm in future, as time is an important criterion in performance analysis of a cloud.

In this, we consider jobs are non-divisible, independent and non-pre-emptive but in future we can do the work on workflows and batch jobs.

10. References

1. Xiaoli Wang, Yuping Wang, Hai Zhu. Energy-Efficient Multi-Job Scheduling Model for Cloud Computing and Its Genetic Algorithm, @ Hindawi Publishing Corporation, 2012.
2. Liu CL, Layland JW. Scheduling algorithms for multiprogramming in a hardreal- time environment, @ Journal of the Association for Computing Machinery. 1973; 20(1):46-61.
3. Davis L. Handbook of Genetic Algorithms, @ New York: Van Nostrand Reinhold, 1991.
4. Sourav Banerjee, Mainak Adhikari, Utpal Biswas, Advanced Task Scheduling for Cloud Service Provider Using Genetic Algorithm, @2012, IOSEJEN.
5. Dutta D, Joshi RC. A Genetic –Algorithm Approach to Cost-Based Multi-QoS Job Scheduling in Cloud Computing Environment, @2011, ICWET.
6. Rosenblum M, Garfinkel T. Virtual Machine Monitors: Current technology and future trends, @ IEEE Computer Society Press, 2005, pp. 39-47.
7. Sipser M. Introduction to the Theory of Computation, International Thomson Publishing, 1st edition, 1996.
8. Coffman EG, Computer and Job-Shop Scheduling Theory Edition: John Wiley and Sons, Inc., New York, NY, 1976.
9. Brucker P. Scheduling Algorithms, @ Springer, Berlin 3rd edition, 2001.
10. Zhangjun Wu12, Xiao Liu2, Zhiwei Ni1, Dong Yuan2, Yun Yang. A Market-Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems in JSC, 2010.
11. Dar-Tzen Peng, Kang Shin G, Tarek Abdelzaher F. Assignment and Scheduling Communicating Periodic Tasks in Distributed Real-Time Systems, @ IEEE Transactions On Software Engineering. 1997; 23(12):745-758.
12. Tzu-Chiang Chiang, Po-Yin Chang, Yueh-Min Huang. Multi-Processor Tasks with Resource and Timing Constraints Using Particle Swarm Optimization, @ IJCSNS International Journal of Computer Science and Network Security. 2006; 6(4):71-77.
13. Chandrasekhara Rao B, Swathi L, Thirupathi Rao K, Krishna Reddy V, Reddy LSS. Genetic-Simulated Annealing Algorithm for Energy Efficient Placement of Virtual Machines in Cloud Computing Environment, @ Global Journal Engineering and Applied Sciences(GJEAS). 2011; 1(3):30-33.
14. Shoukat MM, Maheswaran M, Ali S, Siegel HJ, Hensgen D, Freund RF. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems, @ Journal of Parallel and Distributed Computing. 1999; 59:107-131.