*2015 The 5th International Workshop on Computer Science and Engineering*
doi: 10.18178/wcse.2015.04.009

# Task Scheduling in Cloud Environment Using Multi-objective Genetic Algorithm

R K Jena

Institute of Management Technology, Nagpur, India

rk_jena2@rediffmail.com

Keywords: Task Scheduling, Cloud Computing, Multi-Objective Genetic Algorithm, CloudSim, PISA.

**Abstract.** Cloud computing is an emerging computing paradigm with a large collection of heterogeneous autonomous systems with flexible computational architecture. Task scheduling is an important step to improve the overall performance of the cloud computing. Task scheduling is also essential to reduce power consumption and improve the profit of service providers by reducing processing time. This paper focuses on task scheduling using a multi-objective genetic algorithm (TSGA) to optimize energy and processing time. The result obtained by TSGA was simulated by an open source cloud platform (CloudSim). Finally, the results were compared to existing scheduling algorithms and found that the proposed algorithm (TSGA) provide an optimal balance results for multiple objectives.

## Introduction

Cloud computing is the next generation computational paradigm. It is an emerging computing technology that is rapidly consolidating itself as the future of distributed on-demand computing [1, 2]. Cloud Computing is emerging as vital backbone for the varieties of internet businesses using the principle of virtualization. Many computing frameworks are proposed for the huge data storage and highly parallel computing needs of cloud computing [2]. On the other hand, Internet enabled business (e-Business) is becoming one of best business model in present era. To fulfill the need of internet enabled business, computing is being transformed to a model consisting of services that are commoditized and delivered in a manner similar to traditional utilities such as water, electricity, gas etc. Users can access services based on their requirements without regard to where the services are hosted or how they are delivered. Several computing paradigms have promised to deliver this utility computing [3]. Cloud computing is one such reliable computing paradigm. Cloud computing architecture typically consists of a front end and a back end connected by Internet or Intranet [4]. The front end comprises of client devices like thin client, fat client or mobile devices etc. The clients need some interface and applications for accessing the cloud computing system. The back end consists of the various servers and data storage systems. A central server is used for administering the cloud system. The central server monitors the overall traffic and fulfilling the client demands in real time. The main objective of cloud computing environment is to optimally use the available computing resources. Scheduling algorithms play an important role in optimization process.

This paper presents an optimization algorithm for user job scheduling to achieve optimization of energy consumption and overall computation time. The rest of the paper is organized as, section 2 contains a literature survey about scheduling in cloud computing, section 3 describes about the model development. Section Section 4 outlines the proposed task scheduling model based on Multi-Objective Genetic algorithm. Section 5 discusses details about experimental setup and experimental results of the proposed model and the paper concludes with conclusion in Section 6.

## Literature Review

In cloud computing environment, user services always demand heterogeneous resources (e,g CPU, I/O, Memory etc.). Cloud resources need to be allocated not only to satisfy Quality of Service (QoS) requirements specified by users via Service Level Agreements (SLAs), but also to reduce energy usage and time to execute the user job. Therefore scheduling and load balancing techniques are very crucial to increase the efficiency of cloud setup using limited resources. Task scheduling in Cloud computing has been addressed by many researchers in the past [5-7]. In 2011, Hsu *et al*. [6] focused on energy efficiency in datacenter by using efficient task scheduling to physical servers. Heuristic based techniques have also been used in task scheduling in cloud environment. Mondal *et al*. [8] used Stochastic Hill Climbing algorithm to solve load balance in Cloud computing. Hu *et al*. [9] introduced the scheduling strategy on load balancing of PE resource in Cloud computing environment by using Genetic algorithm. It considered previous data and the current state of work in advance to the performance behavior of the system which can solve the problem of load imbalance in Cloud computing. In 2012, Wei *et al*. [10] presented Genetic algorithm for scheduling in Cloud computing to increase the system performance. Li *et al*. [11] proposed a Load Balancing Ant Colony Optimization (LBACO) Algorithm to reduce makespan in Cloud. Karaboga *et al*., [12] presented ABC algorithm to solve the problem and find the most appropriate parameters in changing environment. Bitam *et al*. [13] proposed Bee Life algorithm for scheduling in Cloud.

## Model Development

To solve the problem of resource optimization using Genetic algorithm within the cloud framework, a typical cloud computing model is proposed as shown in fig. 1. The cloud system consists of many data center that are distributed geographically all over globe and are accessible using internet. Each data center consists of many computing and saving elements and other resources. Processing Elements (PEs) in each data center are connected by a high bandwidth communication network. Therefore negligible communication delay is considered in this model. In the proposed model, user can access the cloud resources using user interface. The proposed task scheduling module in the framework is responsible for efficient allocation of user tasks into different available PE with an objective to optimize energy consumption and time. In fig. 1, 'DC' indicates the Data Center and 'PE' indicates the sets of Processing Elements.

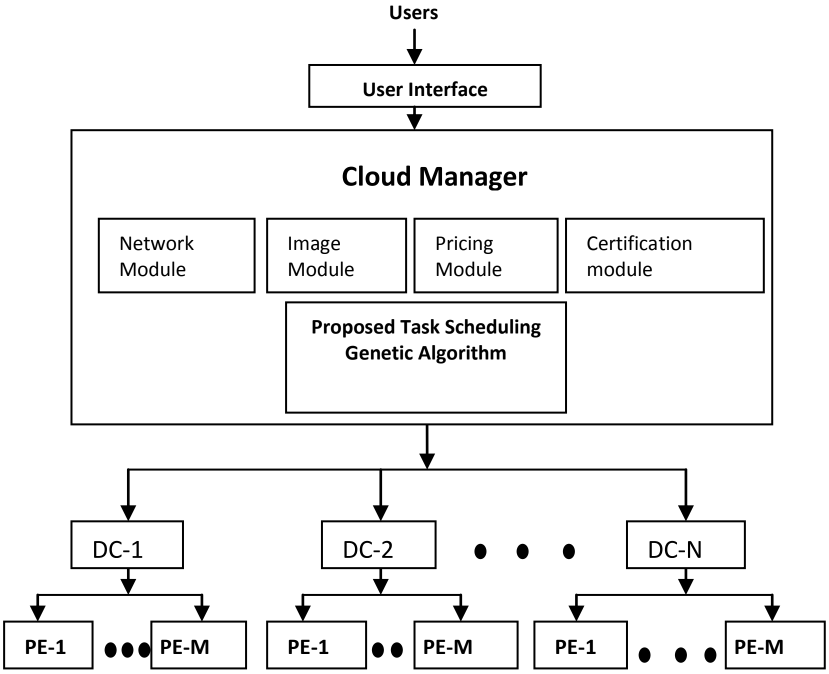

Fig 1: Cloud Scheduling Environment

### 1.1 Problem Formulation

In the proposed model, a cloud application is considered as a collection of user task that carry out a complex computing task using cloud resources. During the scheduling process, the user tasks are assigned to the available data centers (DC's) ( $D_1, D_2, D_3 \ldots. D_M$ ) . Each data center is associated with $< m >$ . m is the number of available Processing Elements (PEs) to execute user tasks. Each data center has set of Processing Elements $\{P_1, P_2 \ldots P_m\}$ to compute user's task. Each Processing Elements is associated with a duplet < s, p > . 's' and 'p' denotes the execution speed and power consumption of each Processing Elements respectively. Each User Job is represented as a as a Directed Acyclic Graph (DAG), denoted as $G(V, E)$. The set of nodes $V = \{ T_1 \ldots, T_n \}$ represents the tasks in user job, the set of arcs denotes precedence constraints and the control/data dependencies between tasks. An arc is in the form of $< T_i, T_j > \in E$, where $T_i$ is called the parent task and $T_j$ is the child task. The data produced by $T_i$ is consumed by $T_j$ . It is assumed that a child task cannot be executed until all of its parent tasks have been completed. In a given task graph, a task with no parent is referred as an *entry task*, and one without any child is called an *exit task*. In this model only one entry and one exit tasks node is considered. Therefore two dummy tasks $T_{entry}$ and $T_{exit}$ is added in the beginning and at the end of the DAG having zero execution time respectively.

Each vertex E in the DAG is associated with a value $< l >$ , '$l$ ' represents the length of the task in Million Instruction (MI). The problem of this model is how to optimally schedule user jobs to the Processing Elements available in the cloud under different data center. All the PEs is considered homogeneous, unrelated and parallel. Scheduling is considered as non preemptive, which means that the processing of any task can't be interrupted.

## 1.2 Objective Function

Suppose user job $U_i$ is assigned to Data center $D_j$ and $T_j$ ( $a$ set of tasks of user job ( $U_i$ )) is assigned to a Processing Element ( $P_j$ ). If the time require executing $T_j$ using $P_j$ is denoted by $\Gamma_j$. The finishing time of $T_j$ can be expressed as:

$$Finish(T_j) = \text{start}\ (T_j) + \Gamma_j \qquad (1)$$

So, the total time spend to complete the user job by $D_j$ ($\text{Makespan}_j$) can be defined as:

$$\text{Makespan}_j = \max\{Finish\ (T_j)\} \qquad (2)$$

Where $T_{j=1 \ldots n}$ the tasks are assign to $D_j$

The Energy consumption to compute the user job ($U_i$ ) by Datacenter $D_j$ is calculated as follows:

$$E_j = \sum_{k=1}^{N} (\Gamma_k \times p_k) \qquad (3)$$

The objective functions of this proposed model can be expresses as:

**Minimize** $\text{Makespan}_j \quad \text{j} = 1..\text{M}$ (4)

**Minimize** $E_j \quad \text{j} = 1..\text{M}$ (5)

Subject to:

1. The user job must finish before deadline $(d_i)$
2. Each user job can be allocated to only one Data center.

## Task Scheduling using Multi-objective GA

In order to deal with the multi-objective nature of cloud task scheduling problem , a multi-objective genetic algorithm is proposed in this research. The algorithm starts with a set of randomly generated solutions (population). The population's size remains constant throughout the algorithm. In a multi-objective optimization, the best solutions encountered over generations are fled into a secondary population called the "Pareto Archive". In the selection process, solutions can be selected from this "Pareto Archive"(elitism). A part of the offspring solutions replace their parents according to the replacement strategy. In this study, an elitist non-dominated sorting genetic algorithm NSGA-II [14] is used. User submits their jobs into the system trough a system interface as shown in the fig.1. Each user job is assigned a timestamp indicating the time of arrival time of the job and added to the arrival queue of the system. Based on the number of available Data Center, the required number job is selected for execution using FCFS principle. A multi-objective Genetic algorithm based method ( TSGA) is used to optimally schedule the tasks of the user jobs to Processing Elements of the respective data center.

```
Algorithm TSGA ( )
{

Input: Number of Data Center ( DC)
          Number of available Processing Element (PE) in each DC
          Number of User job enrolled for scheduling
          Number of Tasks for each User Job
          Power Consumption metric for PE
          Execution Speed metric for PE
          Iteration (I)  := 1
             While  (I <N)
        {
           Assign randomly user jobs to Data Centers
                          {
                NSGA-II
           Store the best solution in a Solution Stack (SS (I))
                }
           I=I+1
          }
     }
```

In the above algorithm, user job is randomly distributed among available datacenter and tasks of each user job are optimally assigned to the PEs of each allocated datacenter by using multi-objective genetic algorithm (NSGA-II). End of each iteration the optimum result is store in a Solution Stack(SS). After end of user defined 'N' iteration, the SS contain N best solutions. The results of TSGA algorithm are a set of Pareto solutions, providing a wide range of options to choose the best solution based on users degree of preference for a particular objective dynamically.

## Implementation and Result

The multi-objective genetic algorithm (NSGA-II) is implemented using PISA tool [15]. Uniform cross over and one point mutation is considered for this experiment. Other GA related parameters used in shown in table-2. CloudSim-3.0.1 is used to evaluate the scheduling of TSGA. The experiments consist of 20 datacenters and 180-360 tasks under the simulation platform. The parameters setting on the proposed algorithm is shown in Table -1 and Table-2.

Table -1: Workload Parameters

| Type | Parameters | Values |
|---|---|---|
| Datacenter | Number of Datacenter | 20 |
| | Number of PE per Datacenter | 10-20 |
| Processing Elements(PEs) | Speed of PE | 1000-200000MIPS |
| | Power Consumption | 0.28-3.45kW |
| Task | Total Number of Tasks | 180-360 |
| | Length of Tasks | 5000-15000 Million |

| | | |
|---|---|---|
| | | Instruction |

Table-2: GA Parameters

| Parameters | Values |
|---|---|
| Number of Generation | 300 |
| Population Size | 100 |
| Crossover  probability | 0.98 |
| Mutation Probability | 0.01 |

Several experiments and with different parameter setting are performed to evaluate the efficiency and efficacy of TSGA algorithm.  Comparison between proposed algorithm (TSGA) with Maximum Applications Scheduling Algorithm(MASA) and Random Scheduling Algorithm(RSA) are given below.  The MASA aims to maximize the number of scheduled applications, while the RSA randomly assigns the applications to the cloud.

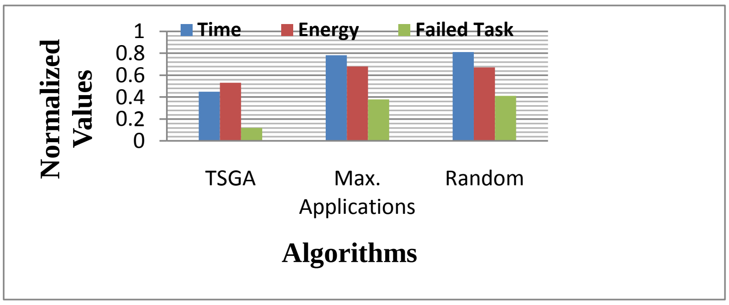

Fig. 2 : Comparisons of different approaches

Fig 2 shows a comparison of results between TSGA and Maximum Scheduling Algorithm(MSA) and Random Scheduling Algorithm(RSA). The proposed algorithm (TSGA) reduced 30% of energy consumption and 25% of time (Makespan) in compare to other scheduling algorithm. The figure (fig. 2) also shows that TSGA drastically reduced the number of failed tasks, which generally increase the profitability of the cloud environment.  The fig 3 shows the effect of optimal solutions with respect to increased number of iteration.  The increased number of iteration improves the quality of solution up to a certain limit. The solution doesn't change much after that. Result shows (fig. 3) after 200 iterations the quality of solution doesn't improve significantly. Again the number of maximum iteration depends on the complexity of the scheduling i.e the number of user job, number of data center etc.

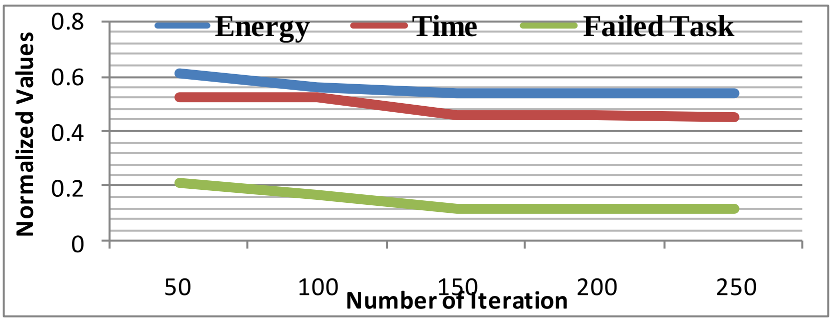

Fig.  3: Optimal values with respect to number of Iteration

## Conclusion

This paper presented multi-objective GA based optimization algorithm which can solve the task scheduling problem under the computing environment, where of the number of data center and user job changes dynamically. But, in changing environment, cloud computing resources needs to be operated in optimally manner. Therefore, multi-objective GA based algorithm is suitable for cloud computing environment because the algorithm is able to effectively utilize the system resources to reduce energy and makespan. For further studies, the optimization model should add more essential objectives ( bandwidth, load balancing, cost etc) and should focus more robust algorithm.

## Reference

[1] Message Passing Interface (MPI), www.mcs.anl.gov/mpi/,(accessed 10.01.2011)

[2] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters, Sixth Symposium on Operating System Design and Implementation (OSDI’04), pp. 1-13, Dec. 2004.

[3] Rajkumar Buyyaa, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic . Cloud Computing and Emerging IT platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Journal Future Generation Computer Systems*, 25( 6), 2009.

[4] Liu, K.. Scheduling Algorithms for Instance Intensive Cloud Workflows. *Ph.D. Thesis*, Swinburne University of Technology, Australia, 2009.

[5] S.Sindhu, and S. Mukherjee. Efficient task scheduling algorithms for cloud computing environment. In *High Performance Architecture and Grid Computing, Communications in Computer and Information Science*, 169, pp. 79-83, 2011.

[6] Y. C. Hsu, P. Liu, and J. J. Wu. Job sequence scheduling for cloud computing. In *Int. Conf. on Cloud and Service Computing (CSC 2011)*,, pp. 212-219 , 2011.

[7] Y. Fang, F. Wang, and J. Ge. A task scheduling algorithm based on load balancing in cloud computing. In *Web Information Systems and Mining, Lecture Notes in Computer Science* , 6318, pp. 271-277,2010.

[8] B. Mondal, K. Dasgupta, and P. Dutta. Load balancing in cloud computing using Stochastic Hill Climbimg-A soft computing approach. In *Procedia Tachnology*, 4, pp. 783-789, 2011.

[9] J. Hu, J. Gu, G. Sun, and T. Zhao,.A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In *3rd Int. Symp. on Parallel Architectures, Algorithms and Programming(PAAP)*, pp. 89-96,2010.

[10 ] Y. Wei, and L. Tian. Research on cloud design resources scheduling based on genetic algorithm, In *2012 Int. Conf. on Systems and Informatics (ICSAI 2012)*, pp. 2651-2656,2012.

[11] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang. Cloud task scheduling based on load balancing Ant Colony Optimization. In *6th Annual ChinaGrid Conf.*, pp. 3-9, 2011.

[12 ] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. In *Artificial Intelligence Review 2012, Springer Science Business Media B.V. 2012*, March 2012.

[13] S. Bitam. Bees life algorithm for job scheduling in cloud computing. In *Conf. on Computing and Information Technology (*ICCIT *2012)*, pp. 186-191,2012.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, 6(2), pp. 182–197, 2002.

[15 ] PISA , Institute TIK, ETH Zürich, http://www.tik.ee.ethz.ch/sop/pisa/