I.J. Information Technology and Computer Science, **2012, 10, 74-79**
Published Online September 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.10.09

An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment

Shaminder Kaur
Department of IT, UIET, Panjab University, Chandigarh, India
sohal.shamin18@gmail.com

Amandeep Verma
Department of IT, UIET, Panjab University, Chandigarh, India
verma_aman81@yahoo.com

Abstract— Cloud computing is recently a booming area and has been emerging as a commercial reality in the information technology domain. Cloud computing represents supplement, consumption and delivery model for IT services that are based on internet on pay as per usage basis. The scheduling of the cloud services to the consumers by service providers influences the cost benefit of this computing paradigm. In such a scenario, Tasks should be scheduled efficiently such that the execution cost and time can be reduced. In this paper, we proposed a meta-heuristic based scheduling, which minimizes execution time and execution cost as well. An improved genetic algorithm is developed by merging two existing scheduling algorithms for scheduling tasks taking into consideration their computational complexity and computing capacity of processing elements. Experimental results show that, under the heavy loads, the proposed algorithm exhibits a good performance.

Index Terms— Cloudlets, Cloud Computing, Genetic Algorithm, Makespan, Task-Scheduling

I. Introduction

A cloud is a type of parallel and distributed system a collection of interconnected and virtualized computer that are dynamically provisioned and presented as one or more unified computing resources based on service level agreements established through negotiation between the service providers and consumers. In this information technology oriented growing market of businesses and organizations, cloud computing is an emerging and attractive alternative to satisfy their day by day increasing needs. It provides virtual resources that are dynamically scalable. It describes virtualized resources, software, platforms, applications, computations and storage to be scalable and provided to users instantly on payment for only what they use [1].

Cloud ecosystem comprises of three main entities: Cloud consumers, cloud service providers, and cloud services. Cloud consumers consume cloud services provided by the cloud service provider. These services may be hosted on the service provider's own infrastructure or on the third party cloud infrastructure providers [2].

1.1 Cloud Computing Service Models

It provides three service models which are- *Cloud Infrastructure as a Service (IaaS), Cloud Platform as Service (PaaS)* and *Cloud Software as a Service (SaaS)*. Cloud IaaS provides consumer the processing, storage, networks and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating system and applications. On the basis of consumers specified hardware (number of CPU cores, physical memory size etc) and software stack (operating system, middleware and application software) is immediately made available by clouds IaaS providers. Cloud PaaS service facilitates developers with providers. Cloud PaaS service facilitate developers with provider specific programming language and tools to develop the applications. Cloud SaaS provides the capability to users to use the provider's application running on cloud infrastructures. The applications are accessible from various client devices through a thin client interface such as web browsers [3]. Sales force Relationships Management (CRM) [4] system and Google Apps [5] are two examples of SaaS. All organizations and business enterprises are taking the full benefits of Cloud computing to reduce the cost and to grow their business quickly. Hence efficient task scheduling is must to influence the decision of service provider for cost benefit of this computing paradigm.

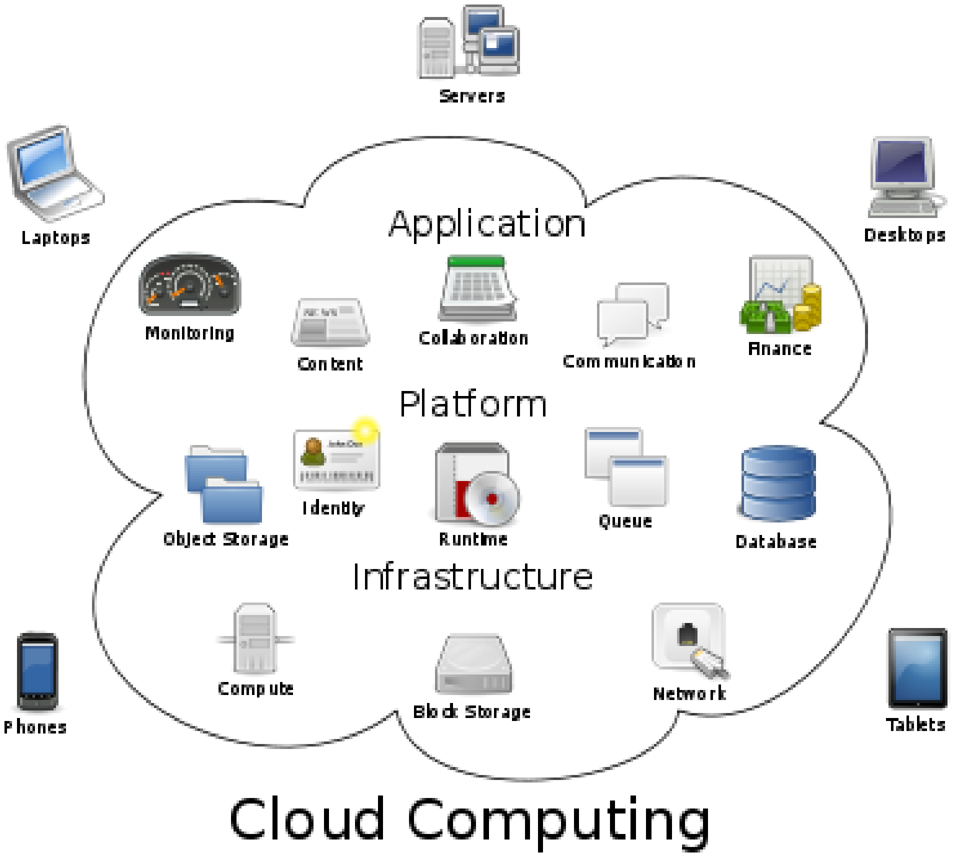

Fig. 1: Service Models

1.2 Problem issues in Clouds

Cloud computing is recently a booming area and has been emerging as an commercial reality in the information technology domain. However the technology is still not fully developed. There are still some areas that are needed to be focused on.

- Resource management
- Task scheduling

Task scheduling and provision of resources are main problem areas in both Grid as well as in cloud computing. Cloud computing is emerging technology in IT domain. The scheduling of the cloud services to the consumers by service providers influences the cost benefit of these computing paradigms. However, there are so many algorithms are given by various researchers for task scheduling, which are discussed in related work.

The remainder of this paper is organized as follows: Section 2 gives related work regarding task scheduling problem in cloud computing technology. Section 3 describes the problem still existing with the in scheduling the tasks in cloud environment. Section 4 presents the implementation of problem. Section 5 presents results and discussions. Conclusion and future work are given in the final section.

II. Related Work

In 2008, A heuristic method to schedule bag-of-tasks (tasks with short execution time and no dependencies) in a cloud is presented in [6] so that the number of virtual machines to execute all the tasks within the budget, is minimum and the same time speedup. In 2009, Marios D. Dikaiakos and George Pallis realized the concept of organization of Distributed Internet Computing as Public Utility and addressed the several significant problems and unexploited opportunities concerning the deployment, efficient operations and use of cloud computing infrastructures [7]. In 2009, Dr. Sudha and Dr. Jayarani proposed the efficient Two-level scheduler (user centric meta-scheduler for selection of resources and system centric VM schedular for dispatching jobs) in cloud computing environment based on QoS[8]. In 2010, Yujia Ge and Guiyi Wei proposed a new scheduler which makes the scheduling decision by evaluating the entire group of tasks in a job queue. A genetic algorithm is designed as the optimization method for a new scheduler who provides better makespan and better balanced load across all nodes than FIFO and delay scheduling [9]. In 2010, An optimal scheduling policy based on linear programming, to outsource deadline constraint workloads in a hybrid cloud scenario is proposed in [10]. In 2011, Sandeep Tayal proposed an algorithm based on Fuzzy-GA optimization which evaluates the entire group of tasks in a job queue on basis of prediction of execution time of tasks assigned to certain processors and makes the scheduling decision [11]. In 2011, Laiping Zhao, Yizhi Ren & Kouichi Sakurai proposed a DRR (Deadline, Reliability, Resource-aware) scheduling algorithm, which schedules the tasks such that all the jobs can be completed before the deadline, ensuring the Realiability and minimization of resources [12].In 2011, S. Sindhu & Saswati Mukherjee proposed two algorithms for cloud computing environment and compared it with default policy of cloudsim toolkit while considering computational complexity of jobs. This paper provided us a framework for our investigation [13].

III. Problem formulation

Task scheduling and provision of resources are main problem areas in both Grid as well as in cloud computing. From the study of related work, we concluded that the existing scheduling strategies in clouds are based on the approaches developed in related areas such as distributed systems and Grids. Scheduling in these areas is mainly tailored toward ensuring single application Service Level Agreement (SLA) objectives. In cloud environment on the other hand require guarantying numerous SLA objectives and quality of service. There are many algorithms like Min-Min, Max-Min, Suffrage, Shortest Cloudlet to Fastest Processor (SCFP), Longest Cloudlet to Fastest Processor (LCFP) and some meta-heuristics like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant-Colony Optimization (ACO) and Simulated Annealing (SA) already existing for task scheduling. However, there are so many algorithms are given by various researchers for clouds, But, none of the above existing algorithms have considered the

- *computational complexity(job length, processing power)*
- *computing cost(processor cost)*

These are required by tasks for scheduling. The two Task Scheduling algorithms of Cloud LCFP & SCFP provide a framework for our investigation which takes the computational complexity and processing power of resource into consideration.

Our proposed work focuses on optimizing the task scheduling algorithms with meta-heuristic algorithms that is Genetic Algorithm (GA) in a private cloud environment. With the combination of SCFP, LCFP and a meta-heuristic GA as an optimization method, we propose to developed a new approach Modified Genetic Algorithm (MGA) for task scheduling. MGA is developed by modifying the initial population with LCFP, SCFP and by controlling the stochastic operators of standard genetic algorithm which lead to achieve a very good results and better efficiency of the algorithm than the standard genetic algorithm. This is for single user jobs in which the fitness will be developed to encourage the formation of solutions to achieve time minimization.

IV. Algorithm Descritption

Our main purpose is to schedule tasks to the adaptable resources in accordance with adaptable time, which involves finding out a proper sequence in which all the tasks can be executed such that execution time and execution cost can be minimized. Cost is also an important parameter as the cloud computing services by service providers to service consumers are provided on internet on pay as per usage basis.

For time minimization, the Genetic Algorithm is a flexible approach enabling, for the same problem, different individual representations and algorithm implementations to select individuals and perform crossover and mutation. A **Genetic algorithm (GA)** is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. However, the appropriate representation of potential solutions is crucial to ensure that the mutation of any pair of individual (i.e. chromosome) will result in new valid and meaningful individual for the problem. An output schedule of tasks is an array list of population (called chromosomes or the genotype of the genome), which encode candidate solutions to an optimization problem, evolves toward better solutions. Time minimization will give profit to service provider and less maintenance cost to the resources. It will also provide benefit to cloud's service users as their application will be executed at reduced cost.

4.1 Existing Algorithm

Standard Genetic Algorithm (SGA)

- Produce an initial population by randomly generated individuals
- Evaluate the fitness of all individuals
- while termination condition not met do
    - select fitter individuals for reproduction
    - crossover between individuals
    - mutate individuals
    - evaluate the fitness of the modified individuals
    - Generate a new population
- End while

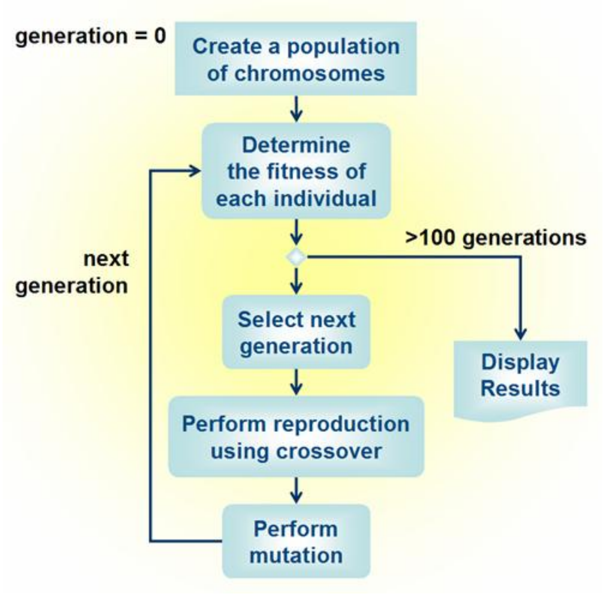
generation = 0
Create a population of chromosomes
Determine the fitness of each individual
next generation
>100 generations
Select next generation
Display Results
Perform reproduction using crossover
Perform mutation

Fig. 2: Flow Chart- Standard Genetic Algorithm (SGA)

4.2 Proposed Algorithm

Modified Genetic Algorithm (MGA)

- Generate an initial population of individuals with output schedules of algorithms Longest Cloudlet to Fastest Processor (LCFP), Smallest Cloudlet to Fastest Processor (SCFP) and 8 Random Schedules.
- Evaluate the fitness of all individuals
- while termination condition not met do
    - Select fitter individuals for reproduction with minimum execution time.
    - Crossover between individuals by two-point crossover.
    - Mutate individuals by simple swap operator.
    - Evaluate the fitness of the modified individuals having relevant fitness.
    - Generate a new population
- End while

Here, Cloudlets refer to user jobs in Cloud Computing.

4.2.1 Schedule Encodings

For schedule encoding we used Permutation-based representation. In this representation each processor must be present only once. This kind of representation is especially useful in sequence based problems, thus it is also interesting for scheduling problems. For the task scheduling problem this representation is obtained in two steps:

1. For each processor *Pi*, construct the sequence *Si* of task assigned to it.
2. Concatenate sequences *Si*. The resulting sequence is a permutation of tasks assigned to processors. This representation requires maintaining additional information on the number of tasks assigned to each processor.

4.2.2 Initial Population:

We have merged **LCFP, SCFP** and 8 Random Solutions to generate the initial population of meta-heuristic which encode candidate solutions to an optimization problem, evolves toward better solutions.

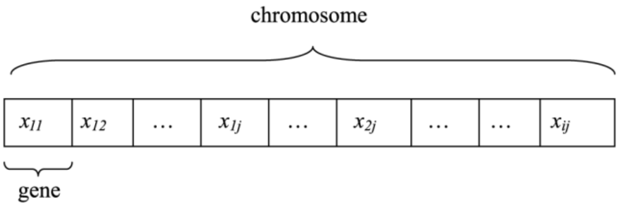

Fig. 3: Chromosome Representation

- Job – an array list to store job length.
- Processor- an array list to store processor speed.
- Best Solution- an array list to hold best solution so far.
- Cost & Time are two characteristics of a genome.

a) LCFP (Longest Cloudlet to Fastest Processor)

1. Sort the cloudlets in descending order of length.
2. Sort the processors in descending order of processing power.
3. Map the cloudlets from sorted list to the sorted list of processors on one-to-one mapping basis.

b) SCFP (Smallest Cloudlet to Fastest Processor)

1. Sort the cloudlets in ascending order of length.
2. Sort the processors in descending order of processing power.
3. Map the jobs from sorted list to the sorted list of processors on one-to-one mapping basis.

4.2.3 Crossover

The crossover operators are the most important ingredient of any evolutionary-like algorithm. Indeed, by selecting individuals from the parental generation and interchanging their *genes*, new individuals (descendants) are obtained. The aim is to obtain descendants of better quality that will feed the next generation and enable the search to explore new regions of solution space not explored yet. There are so many crossover operators which can be used to get the better results. In our case we have chosen the two-point crossover.

4.2.4 Mutation

There are several mutation operators based on the permutation based representation of the schedule like Move, Swap, Move&Swap and Rebalancing. We chose simple Swap.

4.2.5 Evaluation

Evaluation is based on the execution time and execution cost. Those schedules will be selected for next generation whose makespan(execution time of all cloudlets) and execution cost is less than the standard genetic algorithm (SGA)

4.2.6 Termination Condition

Genetic Algorithm gets terminated after user specified number of generations. We generated 30 evolutions of genetic algorithm to get the better results from SGA.

V. Implementation & Results

The two algorithms are implemented on Intel core i5 machine with 500 GB HDD and 4 GB RAM on Windows 7 OS, Eclipse with Java version 1.6, with the help of JGAP (Java based Genetic Algorithm Package)

1. Standard Genetic Algorithm (SGA)
2. Modified Genetic Algorithm (MGA)

A good scheduling algorithm is that which leads to better resource utilization, less average Make-span and better system throughput. Make-span refers to the completion time of all cloudlets in the list. To formulate the problem we considered cloudlets (C1, C2,C3…..C_n) run on processors (P1, P2, P3…..P_n). . Our objective is to minimize Make-span. The speed of processors is expressed in MIPS (Million instructions per second) and length of job can be expressed as number of instructions to be executed. Each processor is assigned varying processing power and respective cost in Indian rupees. We have computed the make-span (completion time of cloudlets) and the corresponding cost of output schedules of above two algorithms and compared them.

All the algorithms are tested by:

- Varying the number of cloudlets.
- Randomly varying the length of cloudlets.

Experimental results show that under heavy loads our proposed algorithm that is modified Genetic Algorithm exhibits a very good performance.

Table 1: GA Parameters

Parameter	Value
Number of Cloudlets	10-30
Number of Processors	5
Number of Iterations	30
Population Size	10
Crossover Type	Two-Point Crossover
Crossover Probability	0.5
Mutation Type	Simple Swap
Mutation Probability	0.6
Termination Condition	Number of Iterations

Table 2: List of Processors

Processor Capacity(Mips)	Per Unit Cost
100	15
200	20
300	25
400	30
500	40

The figure 4 shows the Makespan refers to execution time calculated in seconds of all cloudlets in each of two algorithms. Experimental resulting values show that our proposed algorithm takes less execution time as compared to existing SGA which is based on the random generation of schedules. By modifying the

SGA with stochastic operators we got the better results and better resource utilization as cloudlet load is shared equally on all processors

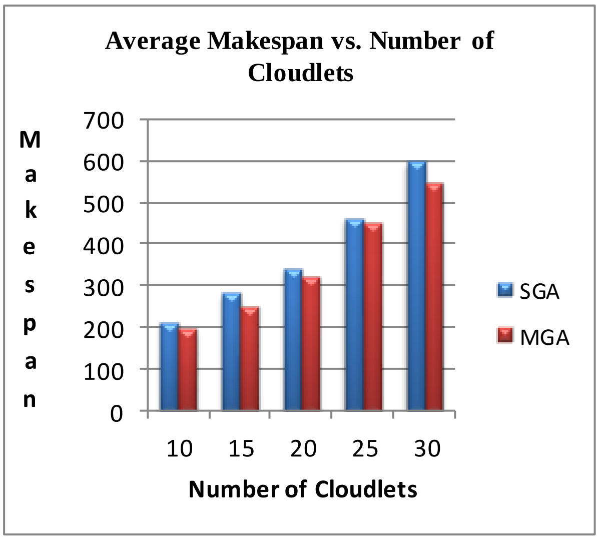

Fig. 4: Shows Average Makespan vs. Number of Cloudlets

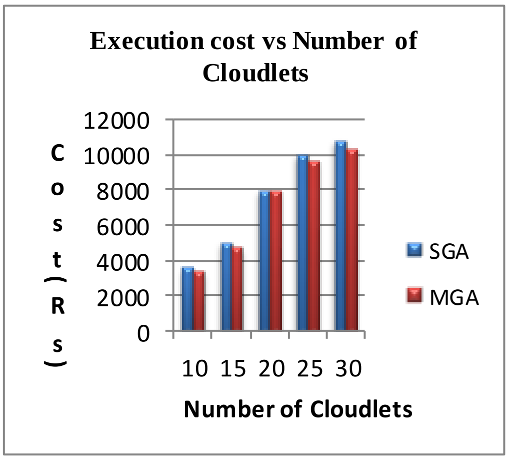

Fig. 5: Shows Average Execution Cost vs. Number of Cloudlets

Above figure compared the execution cost of two algorithms. Resulting values show that performance of proposed algorithm is better than the existing algorithm and keep on increasing with increase in workloads.

VI. Conclusion and Future Work

Clouds enable the users to use utility services. Users are required to pay for access to the services based on their usage and level of quality of service required. In this research we have proposed a modified genetic algorithm for single user jobs in which the fitness is developed to encourage the formation of solutions to achieve the time minimization and compared it with existing heuristics. Experimental results show that, under the heavy loads, the proposed algorithm exhibits a good performance.

In future, we will be further enhancing our algorithm by supporting runtime scheduling and also considering the user's quality of service and priority of jobs for multiple users.

References

[1] Kaur, P.D., Chana, I. "Unfolding the distributed computing paradigm" ,In: International Conference on Advances in Computer Engineering, pp. 339-342 (2010)

[2] Mei, L., Chan, W.K., Tse, T.H., "A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues", In: APSCC 2008, pp. 464-469 (2008)

[3] Silva, J.N., Veiga, L., Ferreira, P.: "Heuristics for Resource Allocation on Utility Computating Infrastructures. In: 6th International Workshop on Middleware for Grid Computing, New York (2008)

[4] Mell, P., Grance, T., "The NIST Definition of Cloud Computing", Version 15, 10-7-09. National Institute of Standard and Technology, Information technology Laboratory (2009)

[5] Salesforce Customer Relationship Management (CRM) system, http://www.salesforce.com/in/?ir=1

[6] http://code.google.com/appengine

[7] Dikaiakos, M., katsaros, D., Mehra, P., Vakali, A.: "Cloud Computing: Distributed Internet Computing for IT and Scientific Research". In:IEEE Transactions on Internet Computing 13(5), pp. 10-13 (2009)

[8] Sadhasivam, S.,Nagaveni, N.: "Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment". In: International Conference on Advances in Recent Technologies in Communication and Computing, pp. 884-886 (IEEE 2009)

[9] Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: "Cost Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In: 3rd IEEE International Conference on Cloud Computing, Miami (July 2010)

[10] Tayal, S.: "Tasks Scheduling Optimization for the Cloud Computing Systems". In: (IJAEST) International Journal of Advanced Engineering Sciences and Technologies, vol. 5, Issue No.2, pp. 111-115 (2011)

[11] Ge, Y., Wei, G.: "GA-Based Task Schedular for the Cloud Computing Systems". In: IEEE International Conference on Web Information Systems and Mining, pp 181-186, WISM.2010.87 (2010)

[12] Zhao, L., Ren, Y., Sakurai, K.: "A Resource Minimizing Scheduling Algorithm with Ensuring the Deadline and Reliability in Heterogeneous Systems". In: International Conference on Advance Information Networking and Applications, AINA.(IEEE 2011)

[13] Sindhu, S., Mukherjee S.: "Efficient Task Scheduling Algorithms for Cloud Computing Environment". In: International Conference on High Performance Architecture and Grid Computing (HPAGC-2011), vol 169, pp 79-83 (2011)

Shaminder Kaur: Post-graduate student for degree of Master of Engineering (ME) in University Institute of Engineering & Technology, Panjab University, Chandigarh, major in Cloud Computing.

Amandeep Verma: Assistant Professor, Department of Information Technology at University Institute of Engineering & Technology, Panjab University, Chandigarh.