

GA-Based Task Scheduling Algorithms in Cloud Computing

Anqi Lu



Master of Science
Computer Science
School of Informatics
University of Edinburgh
January 2015

Abstract

This review covers task scheduling algorithms based on Genetic Algorithm in cloud computing. Cloud computing, relying on sharing of resources, provides platform, infrastructure or software as a service to users. As the task scheduling algorithm affects makespan, cost, Load Balancing etc., the performance of computing depends on it. Genetic Algorithm is one of the many existing algorithms focusing on solving this challenging problem. The three reviewed papers purpose GA-based algorithms with optimization in some steps. In the realization of these algorithms, they take less makespan and cost or have higher utilization of resources compared to Genetic Algorithm in dynamic cloud task scheduling.

Contents

Abstract	3
1 Introduction	7
1.1 Origin of Cloud Computing	7
1.2 Definition of Cloud Computing	7
1.3 Task Scheduling in Cloud Computing	7
1.4 Existing General Algorithms for Task Scheduling in Cloud Computing	7
1.5 Standard Genetic Algorithm	8
1.6 Task Scheduling Algorithms based on Standard Genetic Algorithm	8
1.7 Major Bottlenecks in Current GA-Based Task Scheduling Algorithms	8
2 A Review on Existing GA-based Task Scheduling Algorithms	11
2.1 Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm [1]	11
2.2 Impatient Task Mapping in Elastic Cloud using Genetic Algorithm [2]	12
2.3 An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment [3]	12
3 Comparison and Contrast among Algorithms	13
4 Conclusion	15

Chapter 1

Introduction

1.1 Origin of Cloud Computing

With the popularity and growth of computer service, enterprises and individuals, which provide or use IT services, face with several obstacles e.g. hardware allocation, software development, system maintenance, etc. In terms of hardware, challenges can be shown in allocation of resources, and costs of operating and maintaining of servers. Cloud computing emerges as a solution for these two concerns above.

1.2 Definition of Cloud Computing

Cloud computing refers to a business model, in which providers deliver software, platform and infrastructure as service, while users use them dynamically depending on their exact demand. According to National Institute of Standards and Technology, cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. (The NIST Definition of Cloud Computing, 2)

1.3 Task Scheduling in Cloud Computing

Task scheduling, the key in multitasking computing environment, means a scheme in which computing tasks (e.g. threads, processes or data flows) are allocated with resources (e.g. processor time, communications bandwidth) in an efficient order. In short, it means the order to carry out each computation.

In cloud computing, the task scheduling algorithm mainly determines whether the cloud servers can meet all deadlines of each cloud user, whether the cloud servers are operating in a more economical way, and whether there are crashes when multiple communications are processed simultaneously in a single channel. To achieve the positive answers of above questions, task scheduling algorithm is required to fulfill all commands as well as minimize the resource cost of the schedule.

1.4 Existing General Algorithms for Task Scheduling in Cloud Computing

According to [3], A good scheduling algorithm is that which leads to better resource utilization, less average makespan and better system throughput. Multiple task scheduling algorithms have been designed in order to achieve this target.

These algorithms are mainly built upon following fundamental algorithms: Simulated Annealing, Tabu Search, Genetic Algorithm etc. In this paper, optimized algorithms based on

Genetic Algorithm (GA) are compared and discussed in details.

1.5 Standard Genetic Algorithm

Genetic Algorithm is a computing model, whose process is originated from Darwin's biological evolution theory and mechanism of nature genetics. It is a search heuristic that imitates natural evolutionary process. The general steps of Genetic Algorithm are: Initialization, Selection, Crossover, Mutation, Termination.

There are several major problems with the schedule computed by standard Genetic Algorithm. In the step of initialization, the initial population is computed arbitrarily. It leads to the fact that different schedules do not fit with each other perfectly. Thus in the following mutation step, the probability of generating better children than parents is extremely low. Besides, there exist problems of load imbalance and high migration cost after system VM, which refers to a main system resource, being allocated for tasks. Therefore, we need optimized algorithms to improve this problem.

1.6 Task Scheduling Algorithms based on Standard Genetic Algorithm

There are many algorithms designed based on standard Genetic Algorithm. Based on Avitha. P and J Geetha Reddys review work [5], the briefs of these algorithms are classified as Table 1.1 with a survey on its progress and its comparison with standard Genetic Algorithm.

1.7 Major Bottlenecks in Current GA-Based Task Scheduling Algorithms

GA-Based algorithms complexity depends on every individual task, which means makespan of processing can be of great instability. In addition, in genetic algorithm, moving from one generation iteration to the next can only be executed when all the offspring have been generated. The generating process takes an unexpected waiting time, which tends to be extremely long.

Table 1.1: Classification of Task Scheduling Algorithms based on GA, with Comparison to Standard GA

Algorithm	Progress	Comparison to Standard Genetic Algorithm
Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm [1]	Less makespan	Use Min-Min and Max-Min algorithms to improve the fitness of population in generation
The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing [6]	Higher QoS, higher utilization of resource	Iterates genetic operations
Dynamic scheduling of data using genetic algorithm in cloud computing [7]	Higher processing time utilization, higher system resource utilization	Apply standard GA with the computational power and memory usage taken into consideration
Tasks Scheduling optimization for the Cloud Computing Systems [8]	Higher elasticity	Import fuzzy sets to better apply genetic operators computed by task with low precision
Impatient Task Mapping in Elastic Cloud using Genetic Algorithm [2]	Faster mapping process, more stable growth of mapping time with task quantity going up	Apply standard Genetic Algorithm with exist if satisfy
A Genetic Algorithm for Workload Scheduling In Cloud Based e-Learning [9]	More stable process with processor and IO stream more intensive	Apply conditions such as memory property to standard Genetic Algorithm
A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment [10]	More balanced workflow loading	Improve fitness of generations output from GA with processor utility taken in account
A Method of Cloud Resource Load Balancing Scheduling Based on Improved Adaptive Genetic Algorithm [11]	Index changing more smoothly, lower cost of processors, lower service response makespan	Take real-time system balancing parameters as key factors of scheduling algorithm based on GA
Advanced Task Scheduling for Cloud Service Provider Using Genetic Algorithm [12]	Less waiting time of the workflow and higher QoS	Improve the queue Sequencer, Job Scheduler and Resource Pool to achieve better result
An Efficient Approach to Genetic Algorithm for TaskScheduling in Cloud Computing Environment [3]	Less makespan, less cost of the cloud hardware	Mix SCFP and LCFP with standard Genetic Algorithm to improve fitness of population in initialization
Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm [13]	Lower energy intake, higher profits of the cloud service supplier	Apply Multi-Objective Genetic Algorithm with economical factors taken as determine variables
A Cloud Computing Resource Scheduling Policy Based on Genetic Algorithm with Multiple Fitness [14]	Better performance in virtual machines migration, better load balancing in virtual machines load	Adopt three sub fitness functions (CPU load, network throughput and disk I/O load) to improve standard Genetic Algorithm

Chapter 2

A Review on Existing GA-based Task Scheduling Algorithms

2.1 Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm [1]

According to Genetic Algorithm, better generations tend to emerge with highly-matched solutions, considering genetic operators will be applied to them in the following. In this paper, in order to achieve better solutions, writers propose to apply Min-Min and Max-Min to the individual generation.

Here attaches the two primary algorithms used in the optimization:

- **Min-Min**
Min-Min begins with meta-task which refers to a set collecting and mapping tasks at prescheduled time. Then, it searches for task has the earliest expected completion time on the corresponding machine. After that, the task with the overall least expected completion time from whole meta-task set is picked out and granted with access to resource. And then this task is removed. The termination is executed when all tasks have been mapped with corresponding resource through repeat of above steps.
- **Max-Min**
Similarly, Max-Min initializes meta-task set and looks for task with minimum expected completion time. In contrast to the following step in Min-Min algorithm, it then assigns task which is expected to take the longest makespan to resource.

The optimized algorithm has steps as below:

1. Begin
2. Compute the solution by Min-Min and Max-Min
3. Initialize population by the result of step 2. This step is an improved substitute of the step initialization in standard GA.
4. Evaluation
5. Repeat following steps until termination:
 - (a) Select parents
 - (b) Update combination of pairs of parents
 - (c) Mutate the resulting offspring
 - (d) Evaluation
 - (e) Select individuals for next generation

6. End

The simulation of the improved Genetic Algorithm is implemented with two experiments. In terms of performance, with Virtual Machines as resource and Cloudlets as tasks, the mix of Min-Min, Max-Min and standard Genetic Algorithm proves better schedules with less makespan than standard Genetic Algorithm.

2.2 Impatient Task Mapping in Elastic Cloud using Genetic Algorithm [2]

This paper is focusing on the problem of impatient task, which refers to a job has high demand on the completion time, in consideration of input and output files location and requirement for its Quality of Service (QoS for short term).

An improved GA-based algorithm, which takes exist if satisfy as condition compared to the standard genetic algorithm, is proposed to enable an immediate mapping of task to resource.

Steps of the proposed algorithm are shown as below:

1. Begin
2. Initialization
3. Evaluation
4. Maintenance of the best
5. Loop the following steps until termination:
 - (a) Selection
 - (b) Crossover
 - (c) Mutation
 - (d) Evaluation
 - (e) Elitist
6. End

To achieve the less makespan as well as the better QoS of a given task, the algorithm is aiming at improving the speed of system. The distance between the input files (e.g. task instructions, task data) and processors plays a key role in this study, since the mapping time will go up acceleratingly, if the search space increases. Thus, in the algorithm, beyond what done in standard genetic algorithm, the task scheduler assigns tasks to the nearest local processor when other conditions satisfied.

Two experiments are carried out for simulation of the algorithm, one for testing mapping time of tasks and the other for makespan of tasks.

2.3 An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment [3]

This paper focuses in reduction of execution cost and completion time. A meta-heuristic based algorithm is designed in consideration of tasks computational complexity and computing capacity of processing elements. Compared to standard Genetic Algorithm, the proposed new algorithm performs better in average makespan and average execution cost, especially in heavy load environment.

The descriptions of two general algorithms, LCFP [4] and SCFP [4], which are especially applied to private cloud environment, used besides standard Genetic Algorithm are shown as below:

- LCFP (Longest Cloudlet to Fastest Processor):

This algorithm computes scheduling results with computational complexity taken as key parameter. To make the completion time less, the longer tasks are allocated to the stronger processors. This algorithm shows a better performance in makespan than those not taking demands of tasks into consideration. Here in the algorithm name, cloudlet refers to task. LCFP takes process as following:

1. Sort the cloudlets in descending order of length
2. Sort the processors in descending order of processing power, in scope of all hosts
3. Create virtual machines in the sorted list of processors by packing the same amount of virtual machines as in the processor with strongest power
4. Map the sorted cloudlets to the created virtual machines in the computed order

- SCFP (Smallest Cloudlet to Fastest Processor):

In contrast to LCFP, this algorithm assigns shorter cloudlets to the processors having high computational power, while making sure that longer tasks not exceeding deadlines. Therefore, it overtakes the former algorithm in flow time, which refers to the completion time of a set of jobs in total.

The optimized Genetic Algorithm takes process as below:

1. Begin
2. Initialize population of individuals, using result of LCFP, SCFP as schedules, combined with 8 arbitrary schedules
3. Evaluation
4. Loop the following steps until termination:
 - (a) Look for individuals having the least completion time, and choose them to reproduce generations
 - (b) Operate two point crossover
 - (c) Mutate individuals by simple swap operator
 - (d) Evaluate the fitness of the modified individuals having relevant fitness
 - (e) Generate a new population
5. End

The author implemented the theoretical algorithm to see its makespan and cost. The experiment is carried out on Intel core i5 machine, with 500 GB HDD as storage and 4 GB RAM as computational capacity on Windows 7 OS. A Java package is applied to simulate the improved algorithm and the standard Genetic Algorithm. Besides, the exact values of algorithm parameters involved in the implementation are listed out.

To make the result able to be quantized, the author measures length of tasks (cloudlets) with the quantity of instructions to be executed. To some extent, such research methodologies are not reasonable, since there is no proof of the relationship between the number of instructions to be executed and the makespan of the whole task. In addition, it should be clearly state that the makespan has a proportionality relationship with the number of tasks to be executed.

Additionally, the author declares that each processor is assigned with different computational ability, which is expressed in processor capacity, in a range from 100 Mips (Million Instructions Per Second) to 500 Mips. As is mentioned, the length of cloudlet is calculated in the quantity of instructions to execute. Consequently, such measurement method is reasonable.

Chapter 3

Comparison and Contrast among Algorithms

3.1 Purposes

The common target of these three algorithms reviewed is to improve the performance of job scheduling framework in cloud computing in mainly these aspects: deadline of each task, cost of computation, Load Balancing and makespan. With further development based on standard Genetic Algorithm, these algorithms achieve targets by adopting customized conditions to GA or using other conventional algorithms to optimize some steps in GA.

To minimizing the whole completion time spent for all tasks, the algorithm in [1] increases fitness in the initialization step in GA, thus making the output schedules better fit under the condition of makespan. Similarly, in [3], the focus of algorithm is on improving quality of descendant work flow schedule by optimizing initial population.

However, different methods are taken in [1] and [3], shown in steps of algorithms, which will be reviewed in following section.

In contrast to [1] and [3], in [2], the modified GA-based algorithm aims at reducing mapping time in other steps in GA, though the final performance is also represented as a decrease in completion time. Different with measures taken in the other two paper, this algorithm deals with evaluation step in GA. It searches the best physical machine to process a specific task depending on the distance, which heavily affects the transmitting time.

3.2 Steps of Algorithms

Generally, the procedure of all 3 algorithms is much similar to that of standard Genetic Algorithm, with the basic steps: initialization, selection, crossover, mutation and repetition until termination.

There are also modifications concluded as below. In [1] and [3], on purpose of optimize the quality of initial population as mentioned above, the initialization process is optimized by constructing a root generation fitter to specific requirements, thus improving the multiplication. In the former one, Min-Min and Max-Min algorithms are used, while in the latter one, Longest Cloudlet to Fastest Processor and Smallest Cloudlet to Fastest Processor algorithms are used.

Different from the two algorithms mentioned above, [2] involves a more complicated evaluation process and an elitist process to maintain the fittest chromosome (execution schedule). In the Evaluation process, a total makespan, composed of virtual machine creating time, expected time to execute the job, reading input files, and time spent on outputting the result data to destination physical machines, is used to assess the fitness level of current generation.

3.3 Simulation

3.3.1 CloudSim

CloudSim is used as simulation toolkit in each paper. The realization environment of each experiment, as described in details above, is not a distributed system. It is a single machine with private hardware and software. Therefore, CloudSim is used as a toolkit to simulate the distributed environment. Therefore, regardless of the real operating machine, the experiment result tends to be objective. Specifically, many typical factors (e.g. data transmitting, distributed storage, distributed hardware, dynamic host management etc.) are taken into consideration, since CloudSim can mimic cloud environment with creating datacenters, VMs and physical machines. Additionally, system storage and system brokers can be configured as parameters.

3.3.2 Evaluations

Makespan is taken in each paper as an assessment index to illustrate the performance of algorithms. As it can be precisely calculated and easily compared among different algorithms, it is a convincing measurement of the efficiency of job scheduling mechanism. Also, cost is calculated in [3] as an evaluation index.

3.3.3 Methods

In [1], controlling variables method is used in simulation. In first experiment of [1], quantity of virtual machines is fixed and quantity of tasks is varied. In the second experiment, the fixed variable and the varied variable are exchanged. The experiment method that adopted here helps to illustrate the effect of the change of initialization population, as well as the comparison between improved algorithm and original GA, clearly.

The Minimum Completion Time heuristic is used to compare the proposed algorithm in [2]. Nevertheless, it is confusing here in this paper to import a new algorithm without describing it clearly. The advantage of the algorithm can be represented in a more distinct way by simply compare the result with that of an original GA without I/O transmitting taken into consideration.

In [3], cost of computation, which is not considered in former two papers, is calculated and shown in the result. Paying attention on the cost is of great importance due to the nature of cloud computing includes economic purpose. Experiment results are shown with tasks ranging in number from 10 to 30. The parameter configuration is not reasonable with a small range of task number experimented since in real world the work flow is extremely large.

3.3.4 Results

Though displayed in different ways and different dimensions, the results of purposed algorithms are illustrated with quantity of jobs as variables.

The algorithm in [1] has a better result in makespan than the original GA with a variation of tasks and resources. Also, when quantity of tasks controlled with virtual machines decreasing, the result shows a better utilization of system resources.

Compared with Minimum Completion Time algorithm, the algorithm of [2] takes less completion time over tests on two different datasets. And the great advantage of the purposed algorithm is that the makespan grows much more slowly and steadily, especially when the tasks grow in quantity.

In [3], the algorithm purposed shows an ability to finish the task faster than standard GA with tasks ranging in a small scope. The result lacks precision since the makespan growing speed and Load Balancing are not covered in the report.

Chapter 4

Conclusion

Cloud computing gets its popularity as a pay-per-use business model with three service framework: Infrastructure as a Service, Platform as a Service and Software as a Service. Since cloud provides resources as a shared pool, well-scheduling computational resources plays a key role in cloud computing. Among many conventional algorithms, Genetic Algorithm is the most popular one in task scheduling. Many newly-purposed algorithms are built upon standard GA to achieve better performance in utilization of resource, Quality of Service, cost of computation and makespan etc. The writer mainly reviewed three papers on optimized task scheduling algorithms based on GA. In these algorithms, progress has been made mainly in shortening makespan, reducing cost, balancing system load and improving utilization of resources. There are several flaws in the realization part of the three algorithm, which have been pointed out above. In terms of future work, more distributed system parameters should be taken into account in the same time when task scheduling algorithm in cloud computing is designed.

Bibliography

- [1] Kumar, P., and Verma, A. (2012). Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(5), 111-114.
- [2] Mehdi, N. A., Mamat, A., Ibrahim, H., and Subramaniam, S. K. (2011). Impatient task mapping in elastic cloud using genetic algorithm. *Journal of Computer Science*, 7(6), 877.
- [3] Kaur, S., and Verma, A. (2012). An efficient approach to genetic algorithm for task scheduling in cloud computing environment. *International Journal of Information Technology and Computer Science (IJITCS)*, 4(10), 74.
- [4] Sindhu, S., and Mukherjee, S. (2011). Efficient task scheduling algorithms for cloud computing environment. In *High Performance Architecture and Grid Computing* (pp. 79-83). Springer Berlin Heidelberg.
- [5] Savitha, P., and Reddy, J. Geetha. (2013, August) A Review Work On Task Scheduling In Cloud Computing Using Genetic Algorithm. *International Journal of Scientific & Technology Research*, 2(8), 241.
- [6] Jang, S. H., Kim, T. Y., Kim, J. K., and Lee, J. S. (2012). The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing. *International Journal of Control and Automation*, 5(4), 157-162.
- [7] Ravichandran, S., and Naganathan, D. E. (2013). Dynamic scheduling of data using genetic algorithm in cloud computing. *International Journal of Computing Algorithm*, 2(01), 127-133.
- [8] Tayal, S. (2011). Tasks scheduling optimization for the cloud computing systems. *International Journal of Advanced Engineering Sciences And Technologies (IJAEST)*, 5(2), 111-115.
- [9] Morariu, O., Morariu, C., and Borangiu, T. (2012, April). A genetic algorithm for workload scheduling in cloud based e-learning. In *Proceedings of the 2nd International Workshop on Cloud Computing Platforms* (p. 5). ACM.
- [10] Gu, J., Hu, J., Zhao, T., and Sun, G. (2012). A new resource scheduling strategy based on genetic algorithm in cloud computing environment. *Journal of Computers*, 7(1), 42-52.
- [11] LUa, X., ZHOu, J., and LIUb, D. (2012). A Method of Cloud Resource Load Balancing Scheduling Based on Improved Adaptive Genetic Algorithm.
- [12] Banerjee, S., and MainakAdhikary, U. (2012). Advanced task scheduling for cloud service provider using genetic Algorithm. *IOSR Journal of Engineering (IOSRJEN)*, 2, 141-147.
- [13] Liu, J., Luo, X. G., Zhang, X. M., Zhang, F., and Li, B. N. (2013). Job scheduling model for cloud computing based on multi-objective genetic algorithm. *IJCSI International Journal of Computer Science Issues*, 10(1), 134-139.
- [14] Chen, S., Wu, J., and Lu, Z. (2012, October). A cloud computing resource scheduling policy based on genetic algorithm with multiple fitness. In *Computer and Information Technology (CIT)*, 2012 IEEE 12th International Conference on (pp. 177-184). IEEE.