International Journal of Computer Engineering and Sciences Research

VOL. 01, NO. 01, January 2019
Pages 18–24
Available online at: www.ijcesr.com

Optimization of Task Scheduling using Genetic Algorithm in Federated Cloud Environment

Krishnapal Singh, Sudhanshu Kulshrestha, Ayushi Kulshreshtha, Shivanshu Sharma

Abstract*. Federated Cloud Computing has emerged as a new paradigm where data services are provided by multiple cloud providers over the internet. The main challenge is task management which plays a crucial role in federated cloud environment. The core of this research is optimisation of resource using genetic algorithm. In this paper, we proposed a task scheduling algorithm based on genetic algorithm for executing the various tasks of applications whose aim is to minimize the completion time and increase the utilization of resources. The simulation of the proposed method is done using the CloudSim Toolkit and finally, compared with other algorithms.*

Keywords*: Cloud Computing, Cloud Federation, Task-scheduling, Genetic algorithm*

I. Introduction

Clouds are a large pool of easily usable and configurable computing resources available on demand to user which can be dynamically reconfigured to adjust to a variable load, also allowing for resource utilization [1]. This cloud environment provides a different platform which assists its users in fulfilling their requests in a cost effective manner and within a decent time by creating virtual machines without compromising on the quality of services.

1.1 Federated Cloud Computing

Recent implementations are done basically on a single cloud. In order to address the issue of coordination across different cloud based data centres and due to lack of ability to exchange data and information among different cloud providers, also limited scalability of single-provider clouds, cloud federation was introduced. Using a federated cloud, users are able to deploy their applications over a set of clouds from different cloud providers across different geographical locations, bringing various advantages [2]. Cloud federation allows the various service providers to cooperate with each other to fulfil different peak demand requests and discuss the use of inactive resources with others. Major advantages of federated cloud computing includes scalability, collaboration between partners, aggregation across data centres, reliability, deployment of services and energy consumption.

1.2 Task Scheduling in Federated Cloud Environment

Task scheduling is considered one of the main parameters in optimization problems and also plays an important role in federated cloud computing. The process of task scheduling is complex because it has to order the tasks (jobs) in a way so that it improves the quality of services and maintains efficiency and fairness among the tasks. The job of a scheduler is to see that the optimization of resources is efficiently minimized such that a desirable Quality of Service (QoS) and maximum throughput is achieved. Scheduling architecture, as described in diagram 1, explains that the customers submit their requests to the Data centre broker which behaves like a communicator between the customer and the data centre and helps in allocation of tasks on the virtual machines. Data centre contains a number of hosts where the virtual machines are allocated and these virtual machines are scheduled in accordance with the scheduling policies of the data centre broker.

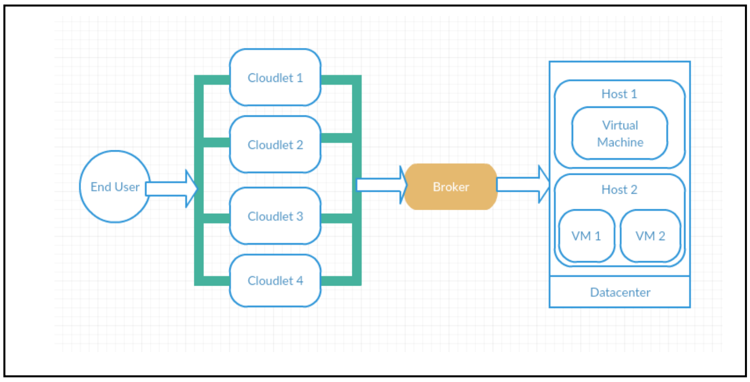

Fig. 1. Task Scheduling Architecture explaining the relationship between the data centre broker and the end user.

The aim that this paper focuses on is to propose an optimal scheduling algorithm based on genetic algorithm which minimizes the completion time for execution of tasks while at the same time, maximum and efficient utilization of cloud resources also happens. Proper resources are allocated at the right time to the right application.

II. Related Work

Task Scheduling in a federated cloud environment has been a topic of interest among many researchers. The main problem that is always addressed is the minimization of execution time. The scheduling of tasks also attracted the attention since it involves different factors like time of completion, the total cost for executing the tasks, resource utilisation, and consumption of power and fault tolerance.

GE Junwei [3] has presented a static genetic algorithm by considering total task completion time, average task completion time, and cost constraint.

One of the problems of allocation of correct resources to the arriving tasks was removed by S. Ravichandran and D.E. Naganathan [4] where they introduced a system which approved the tasks to wait in a queue and scheduling will be done accordingly by allocating the task to the resource of best fit using genetic algorithm.

Nupur Agrahari and Vikash [5] proposed an algorithm which helped in efficient task scheduling and then, did a comparative study on various scheduling algorithms with this proposed algorithm and evaluated their performance with respect to execution cost and completion time.

Furthermore, a comparative study was done by Isam Azawi Mohialdeen[6] in which a practical comparison study among four common job scheduling algorithms- Round Rubin (RR), Random Resource Selection, Opportunistic Load Balancing and Minimum Completion Time in cloud computing was revealed.

M.Srivenkatesh and K.Vanitha [7] have provided an idea for generating initial population by using the Multi level Queue Task scheduling using Genetic Algorithms and provided us with a new modified genetic algorithm.

Safwar A. Hamad and Fatma A. Omara [8] proposed an improved Genetic Algorithm for task scheduling problem in the Cloud computing environment which targets to minimize completion time and cost, and maximize resource utilization and compared its results with the default genetic algorithm.

III. Task Scheduling Techniques

3.1 Round Robin Algorithm

Round Robin Algorithm is one of the simplest and most used job scheduling techniques. The base of this algorithm is that it shares the time among all scheduled tasks in a ready queue. It tries to send the chosen jobs to the virtual machines, which are available, in a round form. The available virtual machine will equally handle each job that will be sent to it to schedule it.

The following algorithm describes how Round Robin Job Scheduling works [6]:

```
Input:

CList- The list of cloudlets (job), VMList-The list of available VMs

Algorithm:
Start
    for k ← 0 to NC do
          c ← CList.get(k);
          i ← (i+1) mod NV;
          v ← VMList.get(i);
          sin ← TimeofTransfer(c,v,in);
          sout ← TimeofTransfer (c,v,out);
          ex ← ExecutionTime(c,v);
          if (c.AT + sin + ex + sout + v.RT <= c.DL )
          then
          sendjob(c,v);
          update(v);
          else
          drop(c);
          failedjobs;
finish.
```

The index (i) for the current job is implemented in a round manner and the keynote step in the algorithm is denoted by the equation 1.

$$i = (i + 1) \bmod NV. \qquad (1)$$

where,
i = index of virtual machine, NV = no of virtual machines, NC = no of cloudlets, sin = stagein, sout = stageout, ex = execution time.

3.2 Genetic Algorithm

Genetic Algorithm, a method of scheduling inspired by Charles Darwin's theory of natural evolution, is considered an increasing area of interest in the field of Artificial Intelligence (AI). According to Darwin's theory, "Survival of the best, discard the rest" is used as a method of scheduling in which resources are assigned tasks according to the value of fitness function parameter [8]. The main terminology used in genetic algorithms is:

i. Individual – single solution to the problem
ii. Population – subset of all possible solutions to the given problem.
iii. Chromosomes – One such solution to the given problem or the raw 'genetic information'
iv. Gene – One element position of a chromosome
v. Allele- the value gene takes for a particular chromosome.
vi. Genotype – particular set of genes in genome.
vii. Phenotype – physical characteristics of genotype

The main principle behind this evolutionary algorithm of population generation is as follows [7] [8]:

A. Initial population

This is the subset of all possible individuals used in the genetic algorithm. Every individual is represented as a chromosome (which could be bit strings, real numbers, list of rules, and any data structure). Initially, the individuals are selected based on some specific criteria and certain operations are performed so that the next generation is formed.

B. Fitness Function

This is the motivating factor in the algorithm as the productivity of the individual depends on it. This problem dependent function measures the performance of the individuals in the population according to the objective of optimization achieved.

C. Selection

This mechanism is the guiding channel, based on Darwin's law of survival, where an intermediary solution is selected for the production of next generation. There is a variety of selection methods for choosing the best chromosomes where the fitness of each solution is rated and then, the possible best solution among them is selected. These include such as Tournament Selection, Roulette wheel, Boltzmann strategy and selection based on rank.

D. Crossover

The crossover technique is done by choosing initially the two parent individuals and then creating a new off-spring by swapping and exchanging the information of those parents. Various crossovers operations exist like single point crossover, two points cross over, uniform cross over. Hybridisation is a guiding process in this.

E. Mutation

Mutation takes place in order to ensure the individual in the population is not homogenous and to introduce genetic diversity in the population. Due to frequent use of reproduction and crossover operations, there might be a chance that any two individuals might be exactly the same. This process alters one or more alleles with a new value in the chromosome. With new genes added, the genetic algorithm maybe able to produce a better solution than before.

IV. The Proposed Genetic Algorithm

In this paper, a modified version of Genetic algorithm has been proposed to resolve the problem that was arising with scheduling of tasks in a federated cloud environment to minimize the execution time on the virtual machines and at the same time maximizing the utilisation of the resources. The main idea that is implemented in our algorithm is that a different selection technique has been used which is eliminating the solution with the least fitness function. By doing this, only the best population goes ahead for crossover process and thus, have the chances of generating the best of the best chromosomes. The best crossover process is selected from a variety of techniques leading to the production of next generation. This way no homogenous solution would be there in the population and the entire population shall contain the best solution. The steps would be as follows:

A. Initial Population

According to our algorithm, the population is generated randomly and each individual is represented as chromosome. The chromosome representation in this algorithm can be Bit Strings, Real Numbers, Permutation of Element, List of Rules, Program Elements (Generic Programming), Any Data Structure, etc.

$$Q = \{1,1,1,0,0,0,1,0\} \quad (2)$$

Imagine the chromosome representation to be an array, as show in equation 2, where the digits used are the genes that represent the Virtual Machine Identifier (VM_ID) and the index at which the numbers are placed is the Cloudlet ID (C_ID).

B. Fitness Function

This is the decision making part for the generation of chromosomes. The chromosome with a good fitness value is selected for further reproduction. The fitness function in our proposed algorithm is to calculate the expected time of completion and execution of each task in the virtual machine which is denoted by the equation 3:

$$Ti = (CLength/Pe \times MIPS) + (Size/BW) \quad (3)$$

where,

CLength = Cloudlet / Task Total length, Size = Cloudlet / Task Input Size,
Pe = Pe count of VM, MIPS = MIPS of VM, BW = Bandwidth of VM

The processing time of all the tasks in the virtual machines can be calculated by the equation 4 as follows:

$$T = \sum_{i=1}^{n} \quad (4)$$

C. Selection Process

Roulette Wheel Selection is used in the proposed algorithm which is more efficient. In the roulette wheel selection method, the fitness value of ever parent chromosomes is evaluated and placed. Imagine a roulette wheel where all the chromosomes are put and has its place accordingly to its fitness function value.

A point is fixed at either side of the wheel and two points are selected as two parent chromosomes for reproduction. The simple base of this selection says the chromosome with bigger fitness value will be selected ample number of times than any other chromosome.

The algorithm for this selection can be simulated as follows:

i. Summation(S) of fitness value of all chromosomes in the population
ii. Generation of a random number from the interval (0,S) – m
iii. When sum S exceeds m, stop and return the chromosome

D. Crossover Operator

Two point crossover method is used as the crossover operator in this algorithm. Two crossover points are chosen. The information from the beginning of first chromosome till the first crossover point and the path from the second crossover point to the end of first chromosome become the end points of the new off-spring. The middle portion of the second chromosome becomes the middle portion of the new chromosome generated. Imagine the above theory being implemented as such where two chromosome material is exchanged producing a new chromosome

1100100**10** + 00**1110**01= **111110110** **(5)**

E. Mutation and Initialization of Subpopulation

After each iteration, the new population, i.e., the child, created is added into the old population, i.e., the parents. In any case if the new off- spring generated is homogenous to any already present chromosome, mutation is done where the alleles of the chromosomes are exchanged internally to produce a new genetic material of the chromosome making it different from the rest of the population. This step improves the variety of the population.

The new improved Genetic Algorithm works in the following manner:

a. Start
b. Randomly initialize population
c. Determine Fitness function value of the virtual machine
d. Repeat until termination condition
a. Roulette wheel selection used to select parents
b. Swapping of genes(mutation) of the new individuals
c. Evaluation of resulting off-springs
d. Selection for new population
e. Finish

V. Results and Performance Analysis

In this section, the experimental evaluation of our algorithm and the other scheduling algorithm is presented, initially describing the simulation environment used for the implementation of proposed algorithm.

5.1 Simulation Tool

CloudSim toolkit is used as the simulation tool for the implementation part of the paper. It is a generalized and extensible simulation framework that enables seamless modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services [9]. This toolkit implements the time-shared and space-shared scheduling policies which are defined as follows: Space-shared scheduling policy is one where a single task is scheduled on the virtual machine at a given instance of time. It behaves as the first come first serve algorithm. Time-shared scheduling policy is one where the scheduling of all tasks takes place on the virtual machine at the same time. The concept of Round-Robin algorithm is used in this policy [10].

According to CloudSim, the user sends a job request in the form of cloudlets to the service provider. Since a lot of cloudlets keep coming simultaneously from the same client, they can't be executed at the same time, therefore they add up in the queue. The classifier classifies the jobs according to different criteria such as number of cloud cores, RAM, secondary storage and others. It's time for the job scheduler to schedule the jobs to system queue. The job are scheduled according to smallest execution time and if found anything insufficient or invalid, rejects the job otherwise, accordingly they are sent to the allocated virtual machines where the execution will be done. After the task requested is completed, they are sent back to the client who made the request in the first place.

5.2 Experimental Results

With the help of CloudSim Toolkit, our proposed algorithm and round robin algorithm is implemented and the results have been compared among the two on the basis of the parameter of completion time.
A single data centre is taken which consist of two hosts and its configurations have been shown in table 1.

Table 1. The configuration details of the datacenter

VM Specifications	Host 1	Host 2
RAM(MB)	16384	16640
Processing power(MIPS)	1000	1000
Storage(MB)	10^6	$2(10)^6$
Bandwidth	10000	10500

The configuration for the three different virtual machines used is described in table 2 shown below:

Table 2. The configuration details of the virtual machine

VMs	VM0	VM1	VM2
RAM	256	512	768
MIPS	500	700	900
Pes number	1	1	1
Bandwidth	1000	2000	3000

Performance with respect to Completion Time

The proposed algorithm is implemented and the result is compared with the results of round robin algorithm on CloudSim. The performance between the two algorithms is evaluated using two policies of the CloudSim-Timeshared and Spaceshared. The percentage of difference of completion time between the two algorithms is calculated as:

$$Percentage = \frac{Time\ of\ RoundRobin - Time\ of\ Proposed\ Genetic}{Time\ of\ RoundRobin} * 1 \quad (5)$$

The result for both the policies of both the algorithms is described in the following table 3 below:

Table 1. Comparison of execution time

Cloudlets	Round Robin		Proposed Genetic Algorithm		Percentage	
	Timeshared	Spaceshared	Timeshared	Spaceshared	Timeshared	Spaceshared
10	35.59	35.6	35.1	35.12	1.38 %	1.35%
25	93.59	93.6	92.2	92.6	1.49 %	1.06 %

50	217.57	217.60	215.68	215.56	0.87 %	1.94 %
75	379.6	380.08	377.50	377.78	0.55 %	0.61 %
100	608.54	608.68	607.50	607.99	0.17 %	0.11 %

In accordance with the results in Table 3, it is found that the completion time of our proposed algorithm is reduced by 0.79 % in timeshared policy and by 0.91 % in spaceshared policy when compared with the Round Robin algorithm correspondingly.

VI. Conclusion and Future Work

Task Scheduling is a crucial factor for successful achievement of best results with available resources in the cloud computing environment. This paper proposes a modified version of Genetic Algorithm which addressed the issue of task scheduling in cloud computing. This algorithm targets to minimize completion cost and the results proved it so. The execution time in our algorithm is reduced by 0.79% in timeshared and by 0.91 % in spaceshared policy, about the Round Robin algorithm accordingly.

Suggestions for future work:

a. The future work can be on other factors for proper scheduling of tasks like cost optimization, speedup and efficiency.
b. The resource is optimized but the possibility of calculation of unused resources can also be taken into account.
c. Any other meta-heuristic algorithm other than genetic algorithm can be used as a basis for implementation for giving better result on task scheduling.

References

[1.] Vaquero, Luis M., et al.: A break in the clouds: towards a cloud definition, ACM SIGCOMM Computer Communication Review 39.1 (2008) 50-55.

[2.] Wen, Zhenyu, et al.: Dynamically partitioning workflow over federated clouds for optimising the monetary cost and handling run-time failures. , IEEE Transactions on Cloud Computing (2016).

[3.] Ge, Jun Wei, and Yong Sheng Yuan.: Research of cloud computing task scheduling algorithm based on improved genetic algorithm. , Applied Mechanics and Materials. , Vol. 347. Trans Tech Publications (2013).

[4.] Ravichandran, S., and Dr ER Naganathan.: Dynamic scheduling of data using genetic algorithm in cloud computing. , International Journal of Computing Algorithm 2.01 (2013) 127-133.

[5.] N. Agrahari, E. Vikas: A Comparative Study on Various Scheduling Algorithms of Cloud Computing, Vol. 7 (3), International Journal of Computer Science and Information Technologies, (2016) 1053-1057

[6.] Mohialdeen, Isam Azawi.: Comparative study of scheduling al-gorithms in cloud computing environment. , Journal of Computer Science 9.2 (2013) 252-263.

[7.] M.Srivenkatesh, K, Vanitha: Task Scheduling in Federated Cloud by Multilevel Queue based on Genetic Algorithm, Vol. 4, International Journal of Advanced Research in Computer and Communication Engineering (2015).

[8.] Hamad, Safwat A., and Fatma A. Omara.: Genetic-based task scheduling algorithm in cloud computing environment., International Journal of Advanced computer Science and Applications 7.4 (2016) 550-556.

[9.] Calheiros, Rodrigo N., et al.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. , Software: Practice and experience 41.1 (2011) 23-50.

[10.] Sidhu, Harmanbir Singh.: Comparative analysis of scheduling algorithms of Cloudsim in cloud computing. , International Journal of Computer Applications 975 (2014).