Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.7, No.6, 2016
www.iiste.org

RSCCGA: Resource Scheduling for Cloud Computing by Genetic Algorithm

Nesa Khandoozi GholiAbad
Islamic Azad University, Gorgan Branch, Iran

The research is financed by Asian Development Bank. No. 2006-A171(Sponsoring information)

Abstract
Cloud computing, also known as on-the-line computing, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources, which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over a network. the scheduling problem is an important issue in the management of resources in the cloud, because despite many requests the data center there is the possibility of scheduling manually. Therefore, the scheduling algorithms play an important role in cloud computing, because the goal of scheduling is to reduce response times and improve resource utilization. The computing resources, either software or hardware, are virtualized and allocated as services from providers to users. The computing resources can be allocated dynamically upon the requirements and preferences of consumers. Traditional system-centric resource management architecture cannot process the resource assignment task and dynamically allocate the available resources in a cloud computing environment. This paper proposed a resource scheduling model for cloud computing based on the genetic algorithm. Experiments show that proposed method has more performance than other methods.
Keywords: Cloud Computing, Resource Management, Scheduling, Bandwidth Consumption, Waiting Time, Genetic algorithm

1. Introduction

Cloud computing is a model for enabling omnipresent, commodious, on-demand access to a shared network containing a pool of configurable computing resources that can be easily purveyed and released with minimal management effort or service provider interaction. Currently cloud computing is provides dynamic services like applications, data, memory, bandwidth and IT services over the internet (Whaiduzzaman,2014).

Traditionally, small and medium enterprises (SMEs) had to make high capital investment upfront for procuring IT infrastructure, skilled developers and system administrators, which results in a high cost of ownership. Cloud computing aims to deliver a network of virtual services so that users can access them from anywhere in the world on subscription at competitive costs depending on their Quality of Service (QoS) requirements , Therefore, SMEs have no longer to invest large capital outlays in hardware to deploy their service or human expense to operate it. In other words, Cloud computing offers significant benefits to these businesses and communities by freeing them from the low-level task of setting up IT infrastructure and thus enabling more focus on innovation and creating business value for their services. Due to such business benefits offered by Cloud computing, many organizations have started building applications on the Cloud infrastructure and making their businesses agile by using flexible and elastic Cloud services. But moving applications and/or Data into the Cloud is not straightforward (Dinh, 2013; athew,2014). Numerous challenges exist to leverage the full potential that Cloud computing promises. These challenges are often related to the fact that existing applications have specific requirements and characteristics that need to be met by Cloud providers. The reliability and performance of cloud services depends up on various factors like scheduling of tasks (Sadiku, 2014).

Scheduling can be done at task level or resource level or workflow level. Scheduling is performed on the basis of different parameters so that it increases the overall cloud performance. A task may include entering data, processing, accessing software, or storage functions. The data center classifies tasks according to the service-level agreement and requested services. Each task is then assigned to one of the available servers. In turn, the servers perform the requested task, and a response, or result, is transmitted back to the user (Garg, 2013). A good scheduling algorithm always improves the CPU utilization, turnaround time and cumulative throughput. Task scheduling can be performed based on different parameters in different ways. They can be statically allocated to various resources at compile time or can be dynamically allocated at runtime. To achieve this purpose, the scheduling system is responsible for different tasks in the cloud to increase completion rates and utilization increase of resources and also increase of computing power. Therefore, the scheduling problem is an important issue in the management of resources in the cloud, because despite many requests the data center there

is the possibility of scheduling manually. In terms of scheduling, the user must consider input parameters of the user such as the cost of implementation, timeline, the issue of efficiency, etc (Jula, 2014). One of the main goals of using cloud computing is to reduce costs of resource use and computing resources are presented as a virtual machine in a cloud computing system . Therefore, the scheduling algorithms play an important role in cloud computing, because the goal of scheduling is to reduce response times and improve resource utilization. The computing resources, either software or hardware, are virtualized and allocated as services from providers to users. The computing resources can be allocated dynamically upon the requirements and preferences of consumers. Traditional system-centric resource management architecture cannot process the resource assignment task and dynamically allocate the available resources in a cloud computing environment. Resource allocation is a hot topic and key factor in distributed computing and grid computing. For distributed computing, processing capacity resources are homogeneous and reserved. However, for grid computing, the resources are highly unpredictable (Vinothina, 2012) . The computers are heterogeneous, their capacities are typically unknown and changing over time, which may connect and disconnect from the grid at any time. Therefore, the same task is sent to more than one computer in Grid computing, and the user receives the output of the computer that completes the task first. Dynamic allocation of tasks to computers is complicated in the grid computing environment due to the complicated process of assigning multiple copies of the same task to different computers. Likewise, the resource allocation is also a big challenge in cloud computing. Cloud computing not only enables users to migrate their data and computation to a remote location with minimal impact on system performance, but also easy access to a cloud computing environment to visit their data and obtain the computation at anytime and anywhere. Cloud computing is attempting to provide cheap and easy access to measurable and billable computational resources comparing with other paradigms such as distribute computing, Grid computing, etc (Grace, 2014 ; Devi, 2014).

1. Background

1.1. Cloud Computing

Cloud computing, also known as on-the-line computing, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers. It relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over a network (Hamrouni, 2016). Advocates claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model. The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing. Companies can scale up as computing needs increase and then scale down again as demands decrease. Cloud computing has become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. Some cloud vendors are experiencing growth rates of 50% per year, but being still in a stage of infancy, it has pitfalls that need to be addressed to make cloud computing services more reliable and user friendly (DAS, 2013).

1.2. Genetic Algorithm

In the field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called a metaheuristic) 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 (Grefenstette, 2013 ; Roberge, 2013). In a genetic algorithm, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are

stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population (Bolaños, 2014 ; Balasubramanian, 2015).

2. Related Works

In order to efficiently allocate computing resources, scheduling becomes a very complicated task in a cloud computing environment where many alternative computers with varying capacities are available. Efficient task scheduling mechanism can meet users' requirements and improve the resource utilization. The cloud service providers often receive lots of computing requests with different requirements and preferences from users simultaneously. Some tasks need to be fulfilled at a lower cost and less computing resources, while some tasks require higher computing ability and take more bandwidth and computing resources. To improve the utility of resource and meet users' requirements, all tasks should be ranked according to available resources such as network bandwidth, complete time, task costs, and reliability of task, When the cloud computing service providers receive the tasks from users, the tasks can be pair wise compared using the comparison matrix technique. The cloud computing providers negotiate with the users on the requirements of tasks including network bandwidth, complete time, task costs, and reliability of task.

FIFO is an exploratory scheduling type that was used for the first time in the first heuristic. Scheduling keeps a new job in the queue based on the time its entry. The work which is in the beginning of expecting list is selected to run. Sincethis scheduling is minimal overhead and it is very easy to implement but functions with long implementation time lower throughput machines (Borthakur, 2007).

Min-Min works in such a way that at every stage, from the tasks and the set of processor selects duty couples / processor which this has the closest time expected to be completed. Then from this set selects the pair of the task / processor having the lowest expected completion time and it assigns the task to the processor. This method is the same as MTC based on the minimum completion time (Bhoi, 2013). However, the method Min-Min studies all tasks unallocated during each event assigned, whereas MTC checks only one task at a time.

Min-Max algorithm is the same as the way using Min-Min. Min-Max algorithm is a set of all unscheduled tasks for each task in the set and then calculates the minimum completion time. The difference between this algorithm with the previous method is that the tasks are completed and mapped with maximum time to the machine. Then, the scheduled task is removed from the collection. This process continues until all tasks are scheduled (Bhoi, 2013).

Scheduling algorithm RASA is presented in (Parsa, 2009) which is a combination of two scheduling algorithms Max-min and Min-min. The algorithm presented uses the advantages of these algorithms and fix the defects. Experimental results have shown that this algorithm has better performance than that of the scheduling algorithm available in large-scale distributed systems.

Scheduling algorithm for workflow SHEFT in (Parsa, 2013) has been presented for scheduling a flexible workflow on the cloud computing environment. Experimental results have shown that they have better performance not only in the number of workflow scheduling algorithm, in optimizing the workflow runtime, but also enable resources to be flexible scalable in runtime.

A scheduling algorithm for resource allocation is presented in (Bittencourt, 2010) which is mixed based on the new architecture. These algorithms have two types of prioritization. The prioritization of resources and prioritization of tasks and prioritization of resources are determined with the numbers of CPU cores than the price of those resources. To prioritize the requests received by the user, at first, the scheduling selects the requests that more time is needed to process and processes resources available in private cloud that it can use them at no cost, if they perform resources available in a certain deadline, then resources available in the public cloud are used that is possible to pay.

In (Ming, 2010), algorithm and a framework for scheduling processing are provided. This algorithm manages serving to requests by the management of the resources available for its management in a hierarchy way. Authors within cloud architecture on services SaaS presented an area as buffer- pool that its work is the division of users on various stages. The task of every stage is to create different methods and send them to resources in the cloud for processing requests.

A method to allocate requests to resources is provided with the aim to minimize total run time of completion in (Paul, 2011) which a cost matrix is created on credits used for each task that has been assigned to a source. Each task has a better chance of having more credibility in allocating the best resources available. In this way architecture is provided storing responsibility for user submissions in a buffer of central middleware, then this divides the tasks hierarchically among local middleware, after this classification, scheduling is performed by the software.

A scheduling algorithm for the tasks as a group in a cloud environment is provided in (Selvarani, 2010)

which the resources are different costs and different mathematical functions. According to classify operations, computing resources and optimized communication are allocated to great works. Grouping algorithm work is so that independent works are grouped for small processing and big works for larger processing.

The main objective of the algorithm (Zenon, 2011) is to minimize the system cost by moving the tokens around the system. But in a scalable cloud system agents cannot have the enough information of distributing the work load due to communication bottleneck. So the workload distribution among the agents is not fixed. The drawback of the token routing algorithm can be removed with the help of heuristic approach of token based load balancing. This algorithm provides the fast and efficient routing decision. In this algorithm agent does not need to have an idea of the complete knowledge of their global state and neighbour‟s working load. To make their decision where to pass the token they actually build their own knowledge base. This knowledge base is actually derived from the previously received tokens. So in this approach no communication overhead is generated.

In paper [26] author presented an optimized algorithm for task scheduling based on Activity Based Costing (ABC). This algorithm assigns priority level for each task and uses cost drivers. ABC measures both cost of the object and performance of the activities. The paper (Yang, 2008) presented transaction intensive cost constraint cloud Work flow scheduling algorithm. Algorithm considers execution cost and execution time as the two key considerations. The algorithm minimizes the cost under certain user designated deadlines. Our proposed methodology is mainly based on computational capability of Virtual Machines.

In paper (Gahlawat, 2013) author proposed an algorithm is Ant colony optimization in which random optimization search approach is used for allocating the incoming jobs to the virtual machines This algorithm uses a positive feedback mechanism and imitates the behavior of real ant colonies in nature to search for food and to connect to each other by pheromone laid on paths traveled.

In paper (Pawar, 2012) is analyzing and evaluating the performance of various CPU scheduling in cloud environment using Cloud Sim the basic algorithm OS like FCFS, Priority Scheduling and Shortest Job First , we test under different which scheduling policy perform better . In (Pawar, 2012) author also proposes a priority based dynamic resource allocation in cloud computing. This paper considers the multiple SLA parameter and resource allocation by pre-emption mechanism for high priority task execution can improve the resource utilization in cloud. The main highlight of the paper is that it provides dynamic resource provisioning and attains multiple SLA objectives through priority based scheduling. Since cost is the important aspect in cloud computing

3. RSCCGA Method

Resource allocation and scheduling of resources have been an important aspect that affects the performance of networking, parallel, distributed computing and cloud computing. Many researchers have proposed various algorithms for allocating, scheduling and scaling the resources efficiently in the cloud In this section, we propose the genetic algorithm resource scheduling method for cloud computing. We also explain how to create model and elements of the GA scheduling function. As mentioned in the previous section, a cloud user reaches a SLA with a cloud provider to process a task. A SLA document includes user requirements like time and budgetary constraints of the task, which indicate acceptable deadline and payable budget of the cloud user. Quality of service attributes like response time and throughput can be comprised in a SLA document besides time and budgetary constraints [2]. A cloud provider has to consider user requirements and virtual machine information before allocating tasks from to virtual machines. In this section we explain initial population generation, Crossover and mutation of proposed method.

3.1. Initial population generation: Genetic algorithm works on fixed bit string representation of individual solution. So, all the possible solutions in the solution space are encoded into binary strings. A chromosome chk indicates the task allocation information, i.e. a task schedule. k is from 1 to z, which denotes the number of chromosomes in a population. In consideration of user satisfaction and provider's profit, the task scheduler determines where to allocate each task every scheduling cycle. A chromosome chk consists of $\alpha[i]$ and $\beta[i]$, which indicate the information of task processing and virtual machine allocation.

3.2. Crossover: The crossover operation is a simple mechanism to swap one part of a chromosome for that of another chromosome. In this paper, the two-point crossover is used for transferring genetic material of parent to children. The objective of this step is to select most of the times the best fitted pair of individuals for crossover. The fitness value of each individual chromosome is calculated using the fitness function as given in 3. This pool of chromosomes undergoes a random single point crossover, where depending upon the crossover point, the portion lying on one side of crossover site is exchanged with the other side. Thus it generates a new pair of individuals.

3.3. Mutation**.** The mutation operation is to expand the search space by changing one part of a chromosome. In the beginning of the genetic algorithm, the quality of generated chromosomes is not particularly good. As time goes by, the quality of chromosomes becomes more improved. In this paper a very small value (0.05) is picked up as mutation probability. Depending upon the mutation value the bits of the chromosomes, are toggled from 1 to 0 or 0 to 1. The output of this is a new mating pool ready for crossover.

The restart operation works as follow: If the best fitness value of the current population does not reach the minimum fitness threshold, chromosomes with high fitness values in the current population fill the half of the next population. And, the rest of the next population is filled with chromosomes generated randomly. This restart operation helps the GA scheduling function to prevent local optimum and explore various search places.

1. ***Start****:* Initialize a random population of chromosomes.
2. ***Fitness***: Evaluate the fitness value of each chromosome.
3. While either maximum number of iteration are exceeded or optimum solution is found Do:
3.1. ***Selection***: Consider chromosome with lowest fitness twice and eliminate the chromosome with highest fitness value to construct the mating pool.
3.2. ***Crossover****:* Perform single point crossover by randomly selecting the crossover point to form new offspring
3.3. ***Mutation*****:** Mutate new offspring with a 0.05 mutation value.
3.4. ***Accepting****:* Place new offspring as new population and use this population for next round of iteration.
3.5. ***Test:*** Test for the end condition.
4. End.

4. Evaluation

The complexity of an algorithm is a function describing the efficiency of the algorithm in terms of the amount of data the algorithm must process. Usually there are natural units for the domain and range of this function. There are two main complexity measures of the efficiency of an algorithm (Time and Space); Time complexity is a function describing the amount of time an algorithm takes in terms of the amount of input to the algorithm. "Time" can mean the number of memory accesses performed, the number of comparisons between integers, the number of times some inner loop is executed, or some other natural unit related to the amount of real time the algorithm will take. We try to keep this idea of time separate from "wall clock" time, since many factors unrelated to the algorithm itself can affect the real time (like the language used, type of computing hardware, proficiency of the programmer, optimization in the compiler, etc.). It turns out that, if we chose the units wisely, all of the other stuff doesn't matter and we can get an independent measure of the efficiency of the algorithm.

. The basic operations performed in genetic algorithm are fitness calculation and selection operation, crossover operation and mutation operation. In genetic algorithm, the population initialization is considered to be the preprocessing hence its complexity is not considered for analysis. For encoding into binary string a time complexity of at most s1, for evaluation of cost function 3 it is at most (b × w) for checking cost b of w number of chromosomes. Selection process has a time complexity of at most z, for single point crossover the time complexity is at most z, where z is chromosome length and for mutation at any place it is again z. The three operation of genetic algorithm are repeated iteratively till the stopping criteria is met so the total time complexity X is given by, $X = O\{s1 + (b \times w) + (s2 + 1)(m + m + m)\}$. We use Matlab for our simulations. Table 1 shows the simulation parameters.

Table 1: simulation parameters

Parameter	Value
Number_of_cluster	4
Number_of_replicas	3
Size_of_each_replica_Min	100;%Megabit
Size_of_each_replica_Max	1000;%Megabit
Size_of_each_job_Min	100;%Megabit
Size_of_each_job_Max	1000;%Megabit
Number_of_generated_requests	10
The_inter_arrival_times_of_nodes_requests_min	0
The_inter_arrival_times_of_nodes_requests_max	99
Storage_space_for_every_client_node	50000;%Megabit
Storage_space_for_server_node	+inf
Number_of_replicas_within_each_group	100;
Network_bandwidth	1000;%Mbps
Distance_between_every_two_directly_connected_nodes_min	1;%km
Distance_between_every_two_directly_connected_nodes_max	1000;%km
Propagation_speed	6000;%Kmps
Frequency_specific_time_interval	10000

For simulation in Matlab environment we set the parameters as follow: Number-of-cluster=4, Number-of-replicas=3, Size-of-each-replica-Min=100 %Megabit, Size-of-each-replica-Max=1000 %Megabit, Size-of-each-job-Min=100 %Megabit, Size-of-each-job-Max=1000 %Megabit, Number-of-generated-requests=10, The-inter-arrival-times-of-nodes-requests-min=0, The-inter-arrival-times-of-nodes-requests-max=99, Storage-space-for-every-client-node=50000 %Megabit, Storage-space-for-server-node=+inf, Number-of-replicas-within-each-group=100, Network-bandwidth=1000;%Mbps, Distance-between-every-two-directly-connected-nodes-min=1 %km, Distance-between-every-two-directly-connected-nodes-max=1000 %km, Propagation-speed=6000 %Kmps, Frequency-specific-time-interval=10000. Figure 1 shows the fitness function and iterations of GA algorithm in iteration 39. This function is optimized by the passage of time.

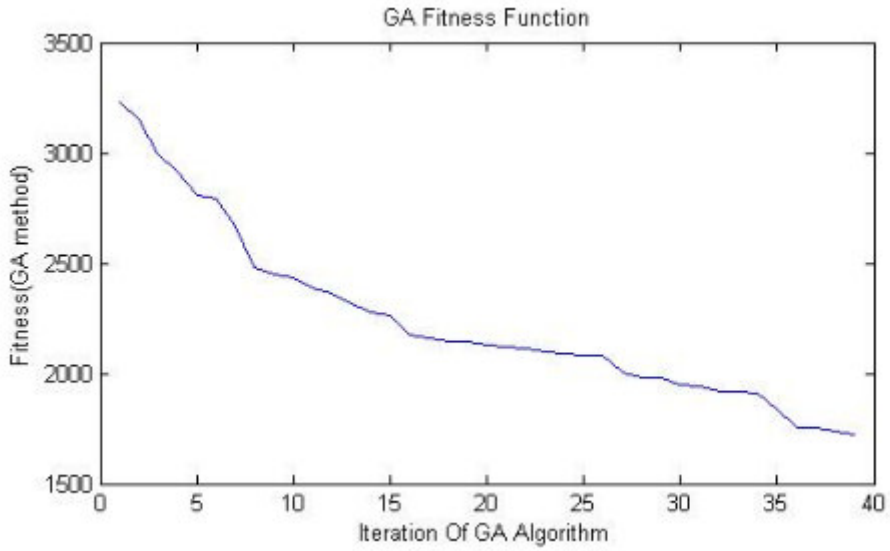

Figure 1: the fitness function and iterations of GA algorithm in iteration 39

Figure 2 shows the bandwidth consumption in proposed method, FIFO and Max-Min methods. As shown in figure 2, the proposed method has less consumption of bandwidth. So that, in iteration 40 bandwidth consumption proposed method was 715, FIFO was 830 and Max-Min was 825.

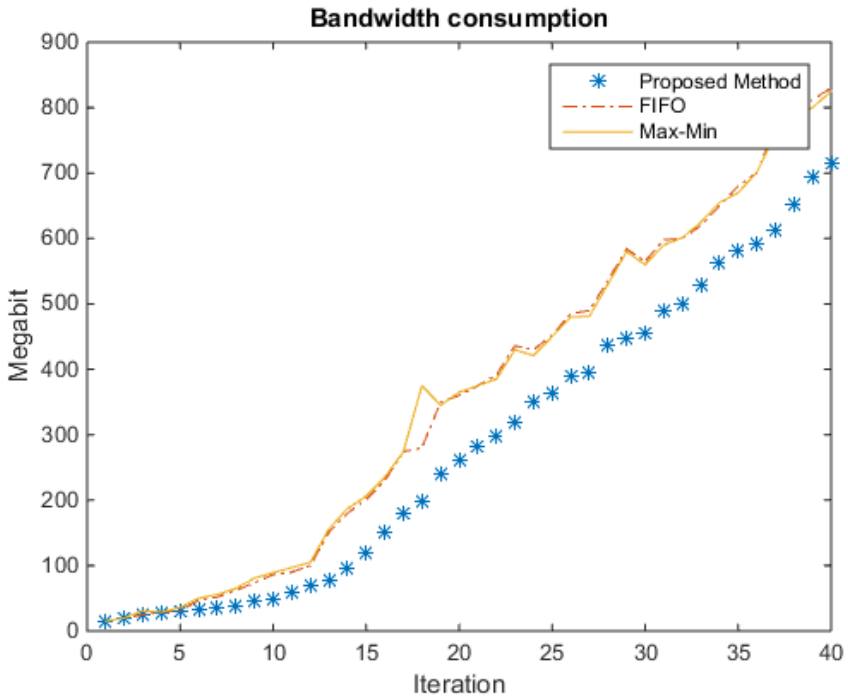

Figure 2: the bandwidth consumption

Figure 3 shows the waiting time for 20 job in proposed method, FIFO and Max-Min methods. As shown in figure 3, waiting time of proposed method was less than FIFO and Max-Min methods. To calculate the waiting time we used the Waiting time = Taking time - Time of arrival equation.

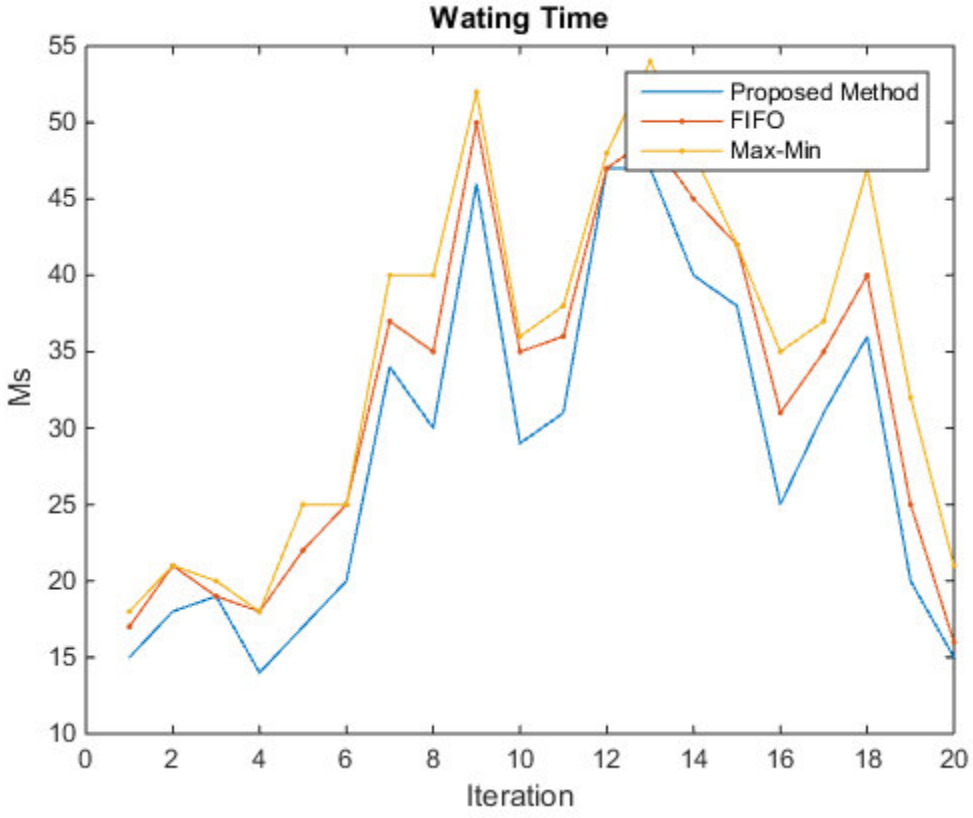

Figure 3: the waiting time for 20 job

5. Conclusion

In this paper we proposed an efficient method for resource scheduling in cloud computing based on genetic

algorithm. The scheduling problem is an important issue in the management of resources in the cloud, because despite many requests the data center there is the possibility of scheduling manually. In terms of scheduling, the user must consider input parameters of the user such as the cost of implementation, timeline, the issue of efficiency, etc. Therefore, the scheduling algorithms play an important role in cloud computing, because the goal of scheduling is to reduce response times and improve resource utilization. Resource allocation is a hot topic and key factor in distributed computing and grid computing. For distributed computing, processing capacity resources are homogeneous and reserved. a cloud user reaches a SLA with a cloud provider to process a task. A SLA document includes user requirements like time and budgetary constraints of the task, which indicate acceptable deadline and payable budget of the cloud user. Quality of service attributes like response time and throughput can be comprised in a SLA document besides time and budgetary constraints. A cloud provider has to consider user requirements and virtual machine information before allocating tasks from to virtual machines. After receiving a SLA our proposed method start and schedule resources for SLA. This method has 3 steps: initial population generation, Crossover and mutation. To evaluate the proposed method we simulate the method in MATLAB. Experiments show that proposed method has more performance than other methods. We use bandwidth consumption and waiting time for comparison the proposed algorithm with FIFO and Max-Min algorithms. Experiments show that in iteration 40 bandwidth consumption proposed method was 715 Megabits, FIFO was 830 Megabits and Max-Min was 825 Megabits. This means the proposed method has less consumption of bandwidth. And also waiting time of proposed method was less than FIFO and Max-Min methods.

References

Whaiduzzaman, M., Sookhak, M., Gani, A., & Buyya, R. (2014). A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, *40*, 325-344.

Dinh, Hoang T., et al. "A survey of mobile cloud computing: architecture, applications, and approaches." Wireless communications and mobile computing 13.18 (2013): 1587-1611

Mathew, T., Sekaran, K. C., & Jose, J. (2014, September). Study and analysis of various task scheduling algorithms in the cloud computing environment. In *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on* (pp. 658-664). IEEE.

Sadiku, M. N., Musa, S. M., & Momoh, O. D. (2014). Cloud computing: Opportunities and challenges. *Potentials, IEEE*, *33*(1), 34-36.

Garg, Saurabh Kumar, Steve Versteeg, and Rajkumar Buyya. "A framework for ranking of cloud computing services." Future Generation Computer Systems 29.4 (2013): 1012-1023.

Jula, A., Sundararajan, E., & Othman, Z. (2014). Cloud computing service composition: A systematic literature review. *Expert Systems with Applications*, *41*(8), 3809-3824.

Vinothina, V. V., R. Sridaran, and Padmavathi Ganapathi. "A survey on resource allocation strategies in cloud computing." International Journal of Advanced Computer Science and Applications (IJACSA) 3.6 (2012)

R. K. Grace and R. Manimegalai, "Dynamic replica placement and selection strategies in data grids—A comprehensive survey," Journal of Parallel and Distributed Computing, vol. 74, pp. 2099-2108, 2014.

K. N. Devi and A. Tamilarasi, "IMPROVING FAULT TOLERANT RESOURCE OPTIMIZED AWARE JOB SCHEDULING FOR GRID COMPUTING," Journal of Computer Science, vol. 10, p. 763, 2014.

T. Hamrouni, et al., "A survey of dynamic replication and replica selection strategies based on data mining techniques in data grids," Engineering Applications of Artificial Intelligence, vol. 48, pp. 140-158, 2016.

DAS , PRANESH . , MOHAN KHILAR , PABITRA . 2013 LBVH :a load balancing technique for virtualization and fault tolerance in cloud computing . international journal of computer applications (0975_8887) . volume 69_ NO .28 .

Grefenstette, John J. Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms. Psychology Press, 2013.

Roberge, Vincent, Mohammed Tarbouchi, and Gilles Labonté. "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning." Industrial Informatics, IEEE Transactions on 9.1 (2013): 132-141.

Bolaños, R., M. Echeverry, and J. Escobar. "A multiobjective non-dominated sorting genetic algorithm (NSGA-II) for the Multiple Traveling Salesman Problem." Decision Science Letters 4.4 (2015): 559-568.

Balasubramanian, Karthik, et al. "Critical Evaluation of Genetic Algorithm Based Fuel Cell Parameter Extraction." Energy Procedia 75 (2015): 1975-1982.

Borthakur, D. (2007). "The hadoop distributed file system: Architecture and design." Hadoop Project Website 11(2007): 21.

Chen, R.-M. and Y.-M. Huang (1998). Multiconstraint task scheduling in multi-processor system by neural network. Tools with Artificial Intelligence, 1998. Proceedings. Tenth IEEE International Conference on,

IEEE.
Bhoi, U. and P. N. Ramanuj (2013). "Enhanced Max-min task scheduling algorithm in cloud computing." International Journal of Application or Innovation in Engineering and Management (IJAIEM): 259-264.
Parsa, S. and R. Entezari-Maleki (2009). "RASA: A new task scheduling algorithm in grid environment." World Applied sciences journal 7: 152-160.
Wu, Z., et al. (2013). "A market-oriented hierarchical scheduling strategy in cloud workflow systems." The Journal of Supercomputing 63(1): 256-293.
Bittencourt, L. F., et al. (2010). Scheduling service workflows for cost optimization in hybrid clouds. Network and Service Management (CNSM), 2010 International Conference on, IEEE.
Ming, C., et al. (2010). A model of scheduling optimizing for cloud computing resource sevices based on Buffer-pool Agent. Granular Computing (GrC), 2010 IEEE International Conference on, IEEE.
Paul, M. and G. Sanyal (2011). Survey and analysis of optimal scheduling strategies in cloud environment. Information and Communication Technologies (WICT), 2011 World Congress on, IEEE.
Selvarani, S. and G. S. Sadhasivam (2010). Improved cost-based algorithm for task scheduling in cloud computing. Computational intelligence and computing research (iccic), 2010 ieee international conference on, IEEE.
Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh, Christopher Mcdermid (2011)"Availabity and Load Balancing in Cloud Computing" International Conference on Computer and Software Modeling IPCSIT vol.14 IACSIT Press,Singapore 2011.
Q. Cao, W. Gong and Z. Wei, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing," In Proceedings of Third International Conference on Bioinformatics and Biomedical Engineering, 2009, pp. 1-3
Y. Yang, Kelvin, J. chen, X. Lin, D.Yuan and H. Jin, "An Algorithm in Swin DeW-C for Scheduling Transaction Intensive Cost Constrained Cloud Workflow," In Proceedings of Fourth IEEE International Conference on eScience, 2008, pp. 374-375
Monica Gahlawat, Priyanka Sharma (2013) " Analysis and Performance Assessment of CPU Scheduling Algorithm in Cloud Sim" International Journal of Applied Information System(IJAIS)- 1SSN: 2249-0868 Foundation of Computer Science FCS, New York, USA Volume5- No 9, July 2013
Pawar, C. S., & Wagh, R. B. (2012). "Priority Based Dynamic resource allocation in Cloud computing", International Symposium on Cloud and Services Computing, IEEE, 2012 pp 1-6