International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
ISSN 2049-3444

International Journal of Engineering and Technology Volume 2 No. 6, June, 2012

Modified Bees Life Algorithm for Job Scheduling in Hybrid Cloud

Tasquia Mizan, [2] Shah Murtaza Rashid Al Masud, [3]Rohaya Latip
[1,2,3]College of Computer Science and Information Systems, Najran University
Najran, Kingdom of Saudi Arabia, P.O.Box-1988

ABSTRACT

Cloud computing is one of the major progenies of distributed system, having excellent service-oriented nature that differentiates it from other IT related knowledge domain. Today, the number of activities and its working capability and capacity in cloud computing environment have been increasing very hastily. Job scheduling is one the vigorous tasks performed in order to gain maximum profit. The efficiency of whole cloud computing services directly relate to the performance of cloud job scheduler associated with cloud data center. During our research we studied current Algorithms for this scheduling task and found that it could still achieve more improvement in the task scheduling process. Therefore, in this paper, we proposed a modified task scheduling algorithm based on the concept of Bees life algorithm and greedy algorithm to gain optimistic value of service in hybrid cloud. The main aim of the system is to achieve an affirmative response at the end users and utilization of resources is done in a very transient manner.

Keywords: *Job scheduling, Bees Life Algorithm, Greedy Method, Hybrid Cloud.*

1. INTRODUCTION

The increasing rate of cloud implementation, and services per year in all around the world is very rapid and noteworthy. Due to these important reasons the overall behavior and functionaries of cloud computing (CC) are changing every day which influence the architecture and its services. Many hardware and software industries, such as IBM, Intel, Microsoft, Cisco, as well as other Internet technology industries, including Google and Amazon, Security Company, such as Semantic, knowledge groups and even several business oriented industries, want to explore the possibilities and benefits of CC are joining the development of cloud services [1-8].

The CC environment is highly dynamic; the system load and the computing resource utilization exhibit a rapidly changing characteristic over time. Therefore the cloud service provider normally over-provision the computing resources to accommodate the peak load and the computing resources are typically left under-utilized at nonpeak times [9].
CC comes with one of the concepts of distributed data centers. Each data centre consists of physical machines to execute customer's tasks on virtual machines where applications, IT services and data are provided over the Internet. In CC system requested task needs to be scheduled as soon as it enters the system taking into account the input and output files location and its quality of service requirements. The Task management, one of the most famous combinatorial optimization problems, is an important issue to improve cloud's flexibility and reliability. The scheduling algorithms in distributed systems usually have the goals of spreading the load on processors and maximizing their utilization while minimizing the total task execution time [10].

A task in any distributed system is a chronological activity, where a set of outputs is produced from a set of inputs. Processes in fixed set are statically assigned to processors, either at compile-time or at start-up. In cloud computing, each application of users will run on a virtual operation system, the cloud systems distributed resources among these virtual operation systems. Every application is completely different and is independent and has no link between each other whatsoever, for example, some require more CPU time to compute complex task, and some others may need more memory to store data. Resources are sacrificed on activities performed on each individual unit of service. In order to measure direct costs of applications, every individual use of resources (such as CPU cost, memory cost, I/O cost, etc.) must be measured. When the direct data of each individual resources cost has been measured, more accurate cost and profit analysis can be done [11].

In this paper, a model of task scheduling is assumed where multiple cloud users request for data centre access. A non-primitive priority queue model/global queue model is used when multiple user request for jobs. Grid information system (GIS) is responsible to allocate the tasks property. Bees life algorithm (BLA) with different components are developed on the based technique for task scheduling with greedy method which will randomly select the set of tasks for one datacenter and also find the nearest idle datacenter and resources in the cloud using shortest path algoritm to reduce the makespan that is the execution time of the cloud computing based services.

This paper is organized in this way; Section 2 Related work with scheduling algorithm. Section 3 Description of the proposed model along with its notations description, algorithm, flowchart, and performance analysis. Section 4 Conclusion and future work.

2. RELATED WORK

Natural process and creature's behavior have inspired scholars to solve complex real-world problems. Optimization is at the heart of many natural processes such as Darwinian evolution, social group behavior and foraging strategies. The last two decades have witnessed notable increasing in the domain of nature-inspired search and optimization algorithms. Recently, these techniques are applied to variant problems. Evolutionary computing methods and the swarm intelligence algorithms are the main groups of that represent the field. In recent researches Swarm Intelligent SI techniques such as Particle Swarm Optimization PSO and firefly algorithm FA represented an alternative search technique, often performed better then genetic algorithm GA when applied to various problems [12]. Evolutionary algorithms, such as genetic algorithms, apply a limited range of movements; which decreases the possibility of trapping in sub optimal. However, evolutionary techniques are slower in finding optimal solutions due to the need of handling population movements [13]. Furthermore, evolutionary algorithms may have a memory to store previous status; this may help in minimizing the number of individuals close to positions in candidate solutions that have been visited before. However, this may also slow the converge since successive generations may die out. Swarm intelligence (SI) such as ant colony optimization (ACO) and particle swarm optimization (PSO) methods are populations of simple agents attempt to find the optimal solution by interacting with one another and with the environment [14][15][16]. In Fireflies Algorithm (FA), fireflies never die; wherein fireflies are considered as simple agents that move and interact through the search space and record the best solution that they have visited [17]. Moreover nature inspired algorithm such as Bees life Algorithm (BLA) use two main sections Reproductions and Food foraging. Reproduction randomly select a set of datacenter tasks(DCTasks) and Food foraging find the nearest datacenter(DC) where evaluation fitness find the tasks property to make the group [18].

In our proposed model we choose Bees Life algorithm (BLA) as an optimization algorithm for its simplicity of operation and power of effect. Each cycle of a bee population life consists of two bee behaviours: reproduction and food foraging respectively. In reproduction behaviour, the queen starts mating in the space by mating-flight with the drones using mutation and crossover operators. Our idea is the adaptation of the BLA operator's value (selection; mutation; crossover) during the run of the BLA. In this section task will be scheduled.In the food foraging part of BLA, we propose to use a greedy method which will find the nearest cloud storage center(CSC) using shortest parh algorithm. Therefore ,scheduler will assign each task to a nearest cloud storage center.

3. MODIFIED JOB SCHEDULING ALGORITHM FOR HYBRID CLOUD

In our proposed model the centralized scheduler refers to a global view of the whole system. The Figure 1 revealed our proposed model. The Greed Information System(GIS) specify the information related to processors which includes slot information, data replication information, workload information of processors and also predicted execution time. Task Model includes the job and tasks information to be processed in the queue using non-preemptive priority queue. Tasks enter the scheduler as a set and gone through BLA algorithm and greedy method to generate an optimal schedule.

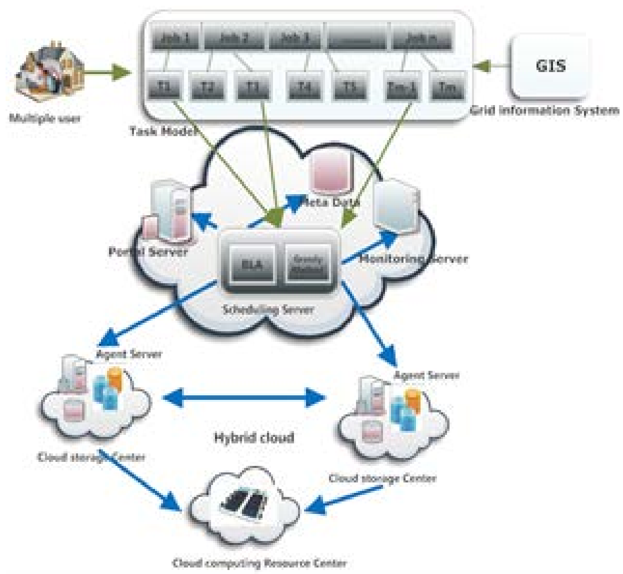

Fig.1. Proposed scheduling model

MapReduce tasks or more general tasks have dependence with each other and computing related factors. But it could be extended for a more comprehensive situation. Suppose that a cloud computing system consist of heterogeneous process unit. The tasks in this system features are as follows:

a. Tasks are aperiodic; i.e., task arrival times are not known a priori. Every task has the attributes arrival time, worst case computation time, and deadline. The ready time of a task is equal to its arrival time.

b. Tasks are non-preemptive; each of them is independent.

c. Each task has two ways of access to a process unit: (1) exclusive access, in which case no other task can use the resource with it, or (2) shared access, in which case it can share the resource with another task, where the other task also should be willing to share the resource [19].

In this model we assuming the scheduler is the centralized master node in the cloud. Scheduler will collect the set of tasks from a Task model. The master unit will communicate with other units. The model assuming that the scheduler always gets the set of new tasks from task model when it finished scheduling the current set of tasks. The master unit works in parallel with other units, scheduling the newly arrived tasks and allocating the cloud storage center or recourses in the cloud. In our model we assuming if a task could not allocate an ideal cloud storage center /resource it will wait for that cloud storage center /resource until it finished the current tasks. Tasks are sorted ascending by the priority. We choose BLA as an optimization algorithm for its simplicity of operation. The main challenge of optimization methods is to increase the chance of finding the global optimal. Greedy methods try to enhance each single step. They have the benefit of finding the optimal solution fast; however, they often trap in local optimal [20]**.** But, the hope we gained about greedy global optimization is from greedy choice property. Greed makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution [21] [22]. In the foraging part of BLA, we propose to use a greedy approach as local search process in order to reach the best individual in the neighborhood from Scheduling server. In this approach, one cloud storage center tasks can be randomly selected to be executed with nearest cloud storage center. In our proposed model, we assuming the greedy method will use shortest path algorithm to find the nearest cloud storage center and recourses in a hybrid cloud.

3.1 Description of Notations Used in Our Proposed Model

The main focus of Job Scheduling is assigning jobs to the cloud data centres and allocates the resources available in the cloud so that the total time of tasks execution (Makespan) is minimized. The scheduling process starts by querying the multiple users' requests and assigning the required resource characteristics in GIS. According to the tasks properties (such as CPU execution time, memory size) the tasks will be grouped based on priority then map the received tasks to the cloud resources through scheduler. The non-preemptive queue will partition the jobs in tasks according first in first serve principle then the GIS will group the tasks as a set of tasks depending the properties and also set the tasks priority (such as execution time) . We consider, Jobs = {J1, J2,..., Jn}, a set of '*n*' jobs to be scheduled in a certain time span and each job can be partitions as a set of '*m*' tasks(T) = {T1, T2,..., Tm } to allocate the CSC in order to be executed.

Suppose, a set of tasks is JiTk = {J1T2, J3T7,……….J10T4} which task properties and priority already assigned by the Global queue and GIS need to be scheduled. This set of tasks will go through the scheduler that is consisting of BLA and greedy method and the scheduler will find out the data centre where each task will be assigned to execute. Then consequently, each CSC can carries out a disjoint subset of the decomposed tasks set. For its assigned jobs CSC ensures the execution of their tasks in this way; CSCjT= {J1T2j, J4T3j,..., JnTmj} where '*j*' is any CSC from a number of data centre selected by greedy shortest path method. Therefore, the total execution time of all job's tasks ('*m*'tasks) in each CSC assigned to CSCj would be: Makespan(CSCjTasks) =Max (JTmj.StartTime + JTmj.ExeTime).

Thus, the job scheduling problem in the cloud computing could be defined as searching of a set, CSCTasks = {CSC1Tasks, CSC2Tasks,…, CSCjTasks } for *j* number of CSC where each cloud storage centre's tasks already grouped as a set of tasks depending on priority ,which will reduce the makespan (CSCjTasks) as it will reduce the CSC Start Time.

Due to BLA is superiorly over other optimized scheduling algorithm, this paper proposes a scheduling model with BLA with greedy approach.

3.2 Our Proposed Overall Algorithm

In this algorithm, *N* means the number of population in bees colony, *D* mention the drones bee's population and *W* mention the worker bee's population. Pseudo code for the proposed job scheduling algorithm using BLA and Greedy Method is shown in Figure 2.

In BLA, at first the algorithm will choose a set of tasks randomly. Then the next step in fitness, the calculation of the makespan for that set of tasks for a particular CSC is done. After that The Algorithm will check the stopping criteria, if the total jobs not scheduled than generate a new set of tasks. In the reproduction stage the algorithm will find out which set of tasks will forwarded to which CSC by mutation and crossover. In the step 'e' fitness will compare the priority for the set of tasks chosen by step 'd' for a particular CSC. If the set of tasks chosen by step 'd' has higher priority then it will be scheduled first by replacing the queen for next generation in step 'f'. In the steps 'g' and 'h' will find out the next set of tasks with priority basis and **c**hoose the tasks that will go to the CSC concurrently. In food foraging behaviour, step 'v' the greedy method will start by initially reach the first CSC using the shortest path algorithm ,then find its successors and repeat the process until the next CSC is found by steps va, vb and vc. When the sets of tasks with less makespan allocated to a nearest CSC in the hybrid cloud then the other set of tasks will be selected with priority basis for scheduling in step 'i'. At the end the optimal solution will obtained. The flowchart for this proposed algorithm is shown in Figure 3.

I. Get new Jobs to be scheduled. The jobs to be scheduled include the uncompleted task and new jobs enter the globel queue and the queue size is predicted as open.
II. Generating tasks property by GIS.
III. Get the current state of the system.
IV. Go through the BLA to get optimized task schedule.
a. Initialize population *N* bees.
b. Evaluate fitness of population.
c. While stopping criteria is not satisfied Forming new population.
/* reproduction behavior */
d. Generate *N* broods by mutation and crossover.
e. Evaluate fitness of broods.
f. If the fittest brood is fitter than the queen then replace the queen for the next generation.
g. Choose *D* best bees among *D* fittest following broods and drones of current population to form next generation drones.
h. Choose *W* best bees among *W* fittest remaining broods and workers of current population to ensure food foraging.
/* food foraging behavior */
V. Use greedy method to find the neighborhood.
Va. Initially, the first neighborhood (priority basis, queue) is reached.
Vb. Use a priority queue to find its successors.
Vc. Repeat process *Vb* until neighborhood is reached
i. Evaluate fitness of population fittest bee is the queen, *D* fittest following bees are drones, *W* fittest remaining bees are workers .
j. End while
VI. Finally obtain the optimal solution.

Fig.2. Pseudo code for our proposed modified job scheduling algorithm

3.3 Efficiency and Performance Analysis

A number of different set of tasks with same number of instructions and assuming the same execution time has been used for examining the efficiency of scheduling methods. The common and significant evaluation methods are makespan and flowtime. Makespan is the time where system completes the latest task; and flowtime is the total of execution times of all tasks submitted to the cloud [17]. In order to evaluate the performance and the effectiveness of BLA scheduler, we assume that 3 jobs to be executed in 3 CSC and resources. Each job can be divided in tasks relevant to the task property and scheduled to the CSC. Each task is defined by the execution time. A simple simulation has been conducted to measure the performance of proposed scheduling algorithm. The results is shown in Figure 4 illustrates that the proposed method has less makespan than the other nature inspired algorithm such as firefly algorithm (FA) or even the genetic algorithm (GA).

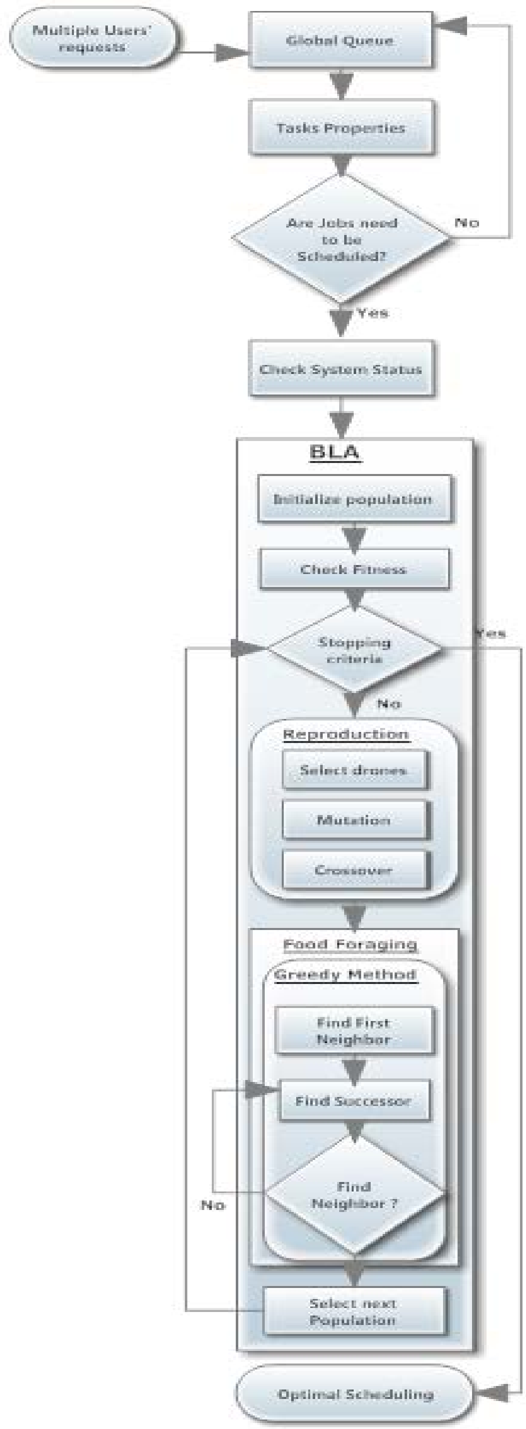

Fig.3. Flowchart using BLA and Greedy method.

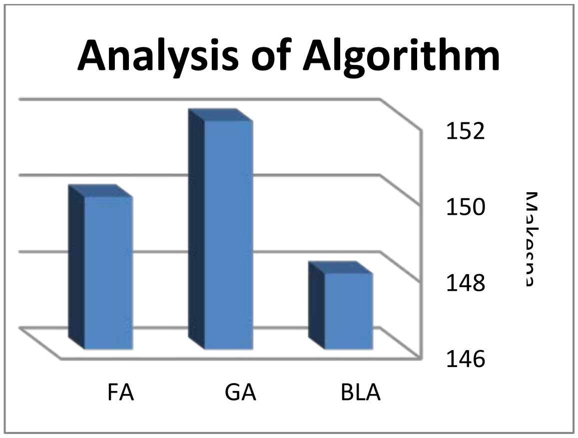

Fig. 4. Analysis of algorithms based on makespan

This figure demonstrate that the proposed BLA method needs less makespan than the others and hence, the delay of tasks reduce by reducing the tasks start time so it effects positively and enhance the efficiency of the scheduling in cloud.

4. CONCLUSION

In this paper job scheduling problem in cloud computing has been studied and a modified job scheduling algorithm based on bees life algorithm and greedy method is proposed for hybrid cloud. The makespan is one of the major issues in job scheduling in cloud computing. The modification carried out from our research through very simple simulation supports to minimize the makespan that is the execution time of the cloud computing based services. Further, a dynamic job scheduling and real time job scheduling aspects could be studied using the model we discussed throughout our paper.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," EECS Dept., Uni. of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb '09

[2] L. Qian, Z. Luo, Y. Du, and L. Guo, "Cloud computing: An overview," in CloudCom '09: Proceedings of the 1st International Conference on Cloud Computing. Springer-Verlag, 2009, pp. 626–631

[3] G. Lin, D. Fu, J. Zhu, and G. Dasmalchi, "Cloud computing: It as a service," IT Professional, vol. 11, no. 2, pp. 10–13, 2009

[4] P. Murray, "Enterprise grade cloud computing," in WDDM '09: Proceedings of the Third Workshop on Dependable Distributed Data Management. New York, NY, USA: ACM, 2009, pp. 1–1

[5] A. Weiss, "Computing in the clouds," NetWorker, vol. 11, no. 4, pp. 16–25, 2007

[6] D. Hilley, "Cloud computing: A taxonomy of platform and infrastructure-level offerings," CERCS, Georgia Institute of Technology, Tech. Rep. GIT-CERCS-09-13, April 2009

[7] Cloud-outlook-for-2012.html: http://cloudonomic.blogspot.com/2011/12/cloud-outlook-for-2012.html #!/2011/12/cloud-outlook-for-2012.html

[8] T. Jowitt, "Four out of five enterprises giving cloud a try," August 27, 2009. Computerworld UK, (visited: 2010, May 7). [Online]. Available: http://www.computerworlduk.com/management/ it-business/services-sourcing/news/index.cfm?newsId=16355

[9] RamenduBikashGuin, SayanChakrabarti, ChinmoyTarafdar, Modelling& Simulation of a Smarter Job Scheduling System for Cloud Computing Service Providers and Users, Computer Science & Engineering Department Kalyani Government Engineering College, Kalyani, Nadia- 741235, West Bengal (INDIA)

[10] A. Y. Zomaya, and Y. The, "Observations on using genetic algorithms for dynamic load-balancing," IEEE Transaction on Parallel and Distributed Systems, vol. 12, no. 9, 2001, pp. 899-911.

[11] Tasks scheduling Optimization for Cloud Computing;Sandeeptayal,University school of Information,GuruGabind Singh University,India

[12] Abraham, A., R. Buyya, and B. Nath. Nature's heuristics for scheduling jobs on computational grids. 2000: Citeseer

[13] Li, S., et al., A GA-based NN approach for makespan estimation. Applied Mathematics and Computation, 2007. 185(2): p. 1003-1014

[14] Liu, H., A. Abraham, and A.E. Hassanien, Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. Future Generation Computer Systems, 2010. 26(8): p. 1336-1343

[15] Dorigo, M. and T. Stützle, Ant colony optimization. 2004: the MIT Press

[16] Abraham, A., et al. Scheduling jobs on computational grids using fuzzy particle swarm algorithm. 2006: Springer

[17] AdilYousif, 'Intelligent Task Scheduling for Computational Grid', UniversitiTeknologi Malaysia. Kassala University, Sudan

[18] Salim Bitam, 'Bees Life Algorithm for Job Schedulingin Cloud Computing', Computer science department, Mohamed Khider University of Biskra, Po. Box 145 Biskra, Algeria

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

[20] Abraham, A., R. Buyya, and B. Nath. Nature's heuristics for scheduling jobs on computational grids. 2000: Citeseer

[21] Greedy choice property: http://vlsicad. ucsd.edu/ courses /cse101/cse101-handout6-6up.pdf

[22] Greedy Algorithm, http://web.mit.edu/k_lai/www/ 6.046/r6-handout.pdf

