Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 2, Number 12; July-September, 2015 pp. 21-25

http://www.krishisanskriti.org/acsit.html

Workflow Scheduling Framework for Cloud Data Analytics Using ACO Algorithm

R.K. Jena

Institute of Management Technology, Nagpur, India
E-mail: rk_jena2@rediffmail.com

Abstract—*Cloud computing is an emerging computing paradigm with a large collection of heterogeneous autonomous systems with flexible computational architecture. Large numbers of tasks are submitted to the cloud environment in each moment. Therefore, the cloud data centers need not only to finish these massive tasks but also to satisfy the user's service demand in time. Reasonably allocation of virtual machine and schedule the tasks efficiently become a key problem to be solved in the cloud environment. Task scheduling is an important step to improve the overall performance of the cloud computing.. This paper focuses on task scheduling using Ant Colony Optimization (ACO) to optimize energy and processing time. The result obtained by ACO is simulated by an open source cloud platform (CloudSim). Finally, the results were compared to existing scheduling algorithms and found that the proposed algorithm provide an optimal balance results.*

Keywords*: Task Scheduling, Cloud Computing, ACO, CloudSim, PISA.*

1. INTRODUCTION

Cloud computing is the next generation computational paradigm. It is an emerging computing technology that is rapidly consolidating itself as the future of distributed on-demand computing [1, 2]. Cloud Computing is emerging as vital backbone for the varieties of internet businesses using the principle of virtualization. Many computing frameworks are proposed for the huge data storage and highly parallel computing needs of cloud computing [2]. On the other hand, Internet enabled business (e-Business) is becoming one of best business model in present era. To fulfill the need of internet enabled business, computing is being transformed to a model consisting of services that are commoditized and delivered in a manner similar to traditional utilities such as water, electricity, gas etc. Users can access services based on their requirements without regard to where the services are hosted or how they are delivered. Several computing paradigms have promised to deliver this utility computing [3]. Cloud computing is one such reliable computing paradigm. Cloud computing architecture typically consists of a front end and a back end connected by Internet or Intranet [4]. The front end comprises of client devices like thin client, fat client or mobile devices etc. The clients need some interface and applications for accessing the cloud computing system. The back end consists of the various servers and data storage systems. A central server is used for administering the cloud system. The central server monitors the overall traffic and fulfilling the client demands in real time. The main objective of cloud computing environment is to optimally use the available computing resources. Scheduling algorithms play an important role in optimization process. Therefore user tasks are required to schedule using efficient scheduling algorithm. The scheduling algorithms usually have the goals of spreading the load on available processors and maximizing their utilization while minimizing the total execution time [5]. Task scheduling is one of the most famous combinatorial NP complete problem problems [6]. The main purpose of scheduling is to schedule the tasks in a proper sequence in which tasks can be executed under problem specific constraints [7].

This paper presents an optimization algorithm for user job scheduling to achieve optimization of energy consumption and overall computation time. The rest of the paper is organized as, section 2 contains a literature survey about scheduling in cloud computing, section 3 describes about the model development. Section 4 discusses about multi-objective Genetic Algorithm followed by Section 5 outlines the proposed task scheduling model based on Multi-Objective Genetic algorithm. Section 6 discusses details about experimental setup and experimental results of the proposed model and the paper concludes with conclusion in Section 7.

2. LITERATURE REVIEW

In cloud computing environment, user services always demand heterogeneous resources (e,g CPU, I/O, Memory etc.). Cloud resources need to be allocated not only to satisfy Quality of Service (QoS) requirements specified by users via Service Level Agreements (SLAs), but also to reduce energy usage and time to execute the user job. Therefore scheduling and load balancing techniques are very crucial to increase the efficiency of cloud setup using limited resources. Task scheduling in Cloud computing has been addressed by many researchers in the past [8-11]. In 2011, Hsu *et al*. [10] focused on energy efficiency in datacenter by using efficient task scheduling to physical servers. Heuristic based techniques have also been used in task scheduling in cloud environment. Mondal *et al*. [12] used Stochastic Hill Climbing algorithm to solve load balance in Cloud computing. Hu *et al*. [13] introduced the scheduling strategy on load balancing of PE resource in Cloud computing environment by using Genetic algorithm. It considered previous data and the current state of work in advance to the performance behavior of the system which can solve the problem of load imbalance in Cloud computing. In 2012, Wei *et al*. [14] presented Genetic algorithm for scheduling in Cloud computing to increase the system performance. Li *et al*. [15] proposed a Load Balancing Ant Colony Optimization (LBACO) Algorithm to reduce makespan in Cloud. Karaboga *et al*., [16] presented ABC algorithm to solve the problem and find the most appropriate parameters in changing environment. Bitam *et al*. [17] proposed Bee Life algorithm for scheduling in Cloud. Mizan *et al*. [18] also solved job scheduling in Hybrid Cloud by modifying Bee Life algorithm and Greedy algorithm to achieve an affirmative response from the end users.

There are many toolkit available to simulated and measure the performance of scheduling and load balancing algorithm in cloud environment. Simulation-based approaches can evaluate Cloud computing system and application behaviors. CloudSim toolkit package is one of the mostly used simulation tool used by many researchers [19,20]. Calheiros *et al*. [19] developed the CloudSim simulation for modeling and simulation of virtualized Cloud-based datacenter environments. The simulation environment consisting up dedicated management interface for PEs, memory, storage, and bandwidth etc. From the above discussion, it is found that most of the previous researches have focused on optimizing a single objective, but very few of them optimize more than two objectives at a time. Therefore it is a good idea to measure the effect of multiple objectives on cloud scheduling problem. To deal with these gaps, a multi-objective Genetic Algorithm is proposed to optimize the energy and time.

2.1 Problem Formulation

In the proposed model, a cloud application is considered as a collection of user task that carry out a complex computing task using cloud resources. During the scheduling process, the user tasks are assigned to the available data centers (DC's) ($D_1, D_2, D_3 \ldots . D_M$) . Each data center is associated with $< m >$. m is the number of available Processing Elements (PEs) to execute user tasks. Each data center has set of Processing Elements $\{P_1, P_2 \ldots P_m\}$ to compute user's task. Each Processing Elements is associated with a duplet < s, p > . 's' and 'p' denotes the execution speed and power consumption of each Processing Elements respectively. Each User Job is represented as a as a Directed Acyclic Graph (DAG), denoted as $G(V, E)$ (fig. 2). The set of nodes $V = \{ T_1 . . , T_n\}$ represents the tasks in user job, the set of arcs denotes precedence constraints and the control/data dependencies between tasks. An arc is in the form of $< T_i, T_j > \in E$, where T_i is called the parent task and T_j is the child task. The data produced by T_i is consumed by T_j . It is assumed that a child task cannot be executed until all of its parent tasks have been completed. In a given task graph, a task with no parent is referred as an *entry task*, and one without any child is called an *exit task*. In this model only one entry and one exit tasks node is considered. Therefore two dummy tasks T_{entry} and T_{exit} is added in the beginning and at the end of the DAG having zero execution time respectively.

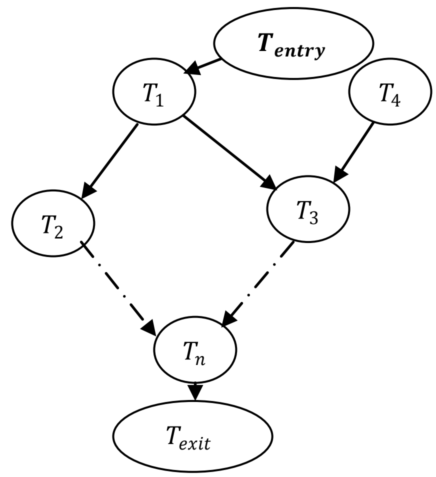

Fig. 1: Task Graph

Each vertex E in the DAG is associated with a value $< l >$, 'l' represents the length of the task in Million Instruction (MI). The problem of this model is how to optimally schedule user jobs to the Processing Elements available in the cloud under different data center. All the PEs is considered homogeneous, unrelated and parallel. Scheduling is considered as non preemptive, which means that the processing of any task can't be interrupted.

2.2 Objective Function

Suppose user job U_i is assigned to Data center D_j and T_j (a set of tasks of user job (U_i)) is assigned to a Processing Element (P_j). If the time require executing T_j using P_j is denoted by Γ_j. The finishing time of T_j can be expressed as:

$$Finish(T_j) = \text{start}\ (T_j) + \Gamma_j \quad (1)$$

So, the total time spend to complete the user job by D_j (Makespan_j) can be defined as:

$$\text{Makespan}_j = \max\{Finish\ (T_j)\} \quad (2)$$

Where $T_{j=1 \ldots n}$ the tasks are assign to D_j

The Energy consumption to compute the user job (U_i) by Datacenter D_j is calculated as follows:

$$E_j = \sum_{k=1}^{N} (\Gamma_k \times p_k) \quad (3)$$

The objective functions of this proposed model can be expresses as:

Minimize $\text{Makespan}_j \; \text{j} = 1..\text{M}$ (4)

Minimize $E_j \; \text{j} = 1..\text{M}$ (5)

Subject to:

1. The user job must finish before deadline (d_i)
2. Each user job can be allocated to only one Data center.

3. ANT COLONY OPTIMIZATION(ACO) ALGORITHM

Ant Colony Optimization (ACO) [21, 22] algorithm is a new kind of simulated evolutionary algorithm [23] has been successfully applied to several NP-hard combinatorial optimization problems [24]. The ACO algorithm was proposed by Italian scholar M. Dorigo [25] in 1996. It is based on food-seeking behavior of ants in 1996. The ants release some pheromone on their way and they never walked when these ants reach a crossing. From the crossing they choose a path randomly and release some pheromone proportional to the length of path. The following ants follow the trail of the other ants to the food source by sensing the pheromone on the path. As this process continues, most of the ant likely to choose the shortest path with a huge amount of pheromones. During this occasion a mechanism of positive feedback is formed, this mechanism ensures that the good information can be preserved and finally ants can find an optimal [26, 27]. In this case, there are more and more pheromones on this path. As the time on, the amount of information on other paths will be gradually reduced, eventually the path most ants moved to will be the optimal path.

Let us apply the ACO algorithm to find shortest distance between two cities among 'n' cities. Suppose there are 'm' numbers of ants. Firstly the algorithm set any ants into some randomly selected cities. Ant *k* *(k=1, 2, 3,…, m)* will determine the transfer direction according to the concentration of the pheromone on each path in the searching for target city. At first, the ants will randomly select a path because of the minor difference of the pheromone quantity among paths. The tabu list $tabu_k$ *(k=1, 2, 3,…, m)* is to record the path which ant *k* has walked and it will adjust dynamically along with the changing movement of the ant.

Let $P_{ij}^k(t)$ be the state of transition probability of ant 'k' for selecting city 'j' as the target city at a moment 't'.

$$P_{ij}^k(t)=\begin{cases} \dfrac{[\tau_{ij}(t)]^{\propto}\,[\partial_{ij}(t)]^{\beta}}{\sum_{s\,c\,allowed\;\;k}[\tau_{is}(t)]^{\propto}\,[\partial_{is}(t)]^{\beta}}\;, if\; j\epsilon allowed_k \\ 0,\quad else \end{cases}$$

In the above equation, $allowed_k$ represents the allowed cities of ant k in the next step. $\tau_{ij}(t)$ represents the pheromone of the path between city *i* and city *j* at time *t*. At the initial time, the pheromone of each path is equal. $\partial_{ij}(t)$ is a heuristic function and defined as $\partial_{ij}(t) = {}^{1}/_{d_{ij}}$ (d_{ij} is the distance between nodes) to represent the desired transfer degree of ants from the city i to the city *j*. α is the heuristic factor of pheromone, which indicates the relative importance of the track. It reflects the guiding effect of the information accumulated in the track on the ant's movement. The greater the values is, the more likely the ant chooses the track which other ants passed by. β is expected heuristic factor, which indicates the relative weight of calculation ability. It reflects the importance degree of heuristic information in ant's choice. The greater the value is, the closer the state transition probability to the greedy rule.

4. IMPLEMENTATION AND RESULT

The ACO algorithm is considered for this experiment. Other GA related parameters used in shown in table-4. CloudSim-3.0.1 is used to evaluate the scheduling of TSGA. The experiments consist of 20 datacenters and 180-360 tasks under the simulation platform. The parameters setting on the proposed algorithm is shown in Table -1 and Table-2.

Table 1: Workload Parameters

Type	Parameters	Values
Datacenter	Number of Datacenter	20
	Number of PE per Datacenter	10-20
Processing Elements(PEs)	Speed of PE	1000-200000MIPS
	Power Consumption	0.28-3.45kW
Task	Total Number of Tasks	180-360
	Length of Tasks	5000-15000 Million Instruction

Table 2: ACO Parameters

Parameters	Values
α	0.8
β	0.8
ρ	0.4
m	50
NCmax	50

Several experiments and with different parameter setting are performed to evaluate the efficiency and efficacy of ACO algorithm. Comparison between proposed algorithm (ACO) with Maximum Applications Scheduling Algorithm(MASA) and Random Scheduling Algorithm(RSA) are given below. The MASA aims to maximize the number of scheduled applications, while the RSA randomly assigns the applications to the cloud.

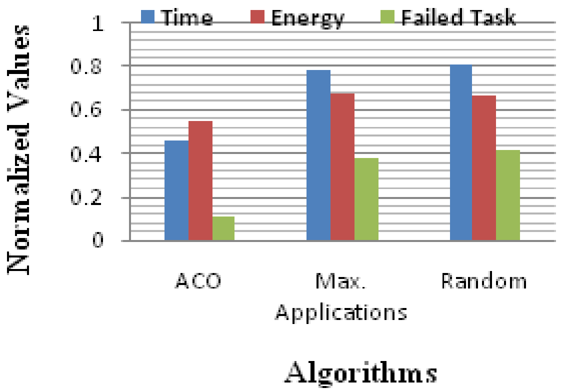

Fig. 2: Comparisons of different approaches

Fig 2 shows a comparison of results between ACO and Maximum Scheduling Algorithm(MSA) and Random Scheduling Algorithm(RSA). The proposed algorithm (TSGA) reduced 25% of energy consumption and 22% of time (Makespan) in compare to other scheduling algorithm. The figure (fig. 3) also shows that TSGA drastically reduced the number of failed tasks, which generally increase the profitability of the cloud environment.

5. CONCLUSION

This paper presented ACO based optimization algorithm which can solve the task scheduling problem under the computing environment, where of the number of data center and user job changes dynamically. But, in changing environment, cloud computing resources needs to be operated in optimally manner. Therefore, ACO based algorithm is suitable for cloud computing environment because the algorithm is able to effectively utilize the system resources to reduce energy and makespan. The experimental results illustrated that the proposed methods using ACO out-performed the maximum applications scheduling algorithm and random scheduling For further studies, the optimization model should add more essential objectives (bandwidth, load balancing, cost etc) and should focus more robust algorithm.

REFERENCE

[1] Message Passing Interface (MPI), www.mcs.anl.gov/mpi/,(accessed 10.01.2011)

[2] 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.

[3] Rajkumar Buyyaa, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic . "Cloud Computing and Emerging IT platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", *Journal Future Generation Computer Systems*, 25(6), 2009.

[4] Liu, K.. "Scheduling Algorithms for Instance Intensive Cloud Workflows". *Ph.D. Thesis*, Swinburne University of Technology, Australia, 2009.

[5] A. Y. Zomaya, and Y. "The. Observations on using genetic algorithms for dynamic load-balancing*", IEEE Transaction on Parallel and Distributed Systems*, 12(9), 2001, pp. 899-911.

[6] Yu, J., and Buyya, R.. Workflow Scheduling Algorithms for Grid Computing. Xhafa F, Abraham A (eds), *Metaheuristics for scheduling in distributed computing environments*, Springer, Berlin, 2008.

[7] XIAO Zhi-Jiao, CHANG Hui-You,YI Yang. "An Optimization M ethod of W orkflow Dynamic Scheduling Based on Heuristic GA" , *Computer Science*, .34(2), 2007.

[8] S.Sindhu, and S. Mukherjee." Efficient task scheduling algorithms for cloud computing environment." In *High Performance Architecture and Grid Computing, Communications in Computer and Information Science*, 169, 2011, pp. 79-83.

[9] S. T. Maguluri, R. Srikant, and L. Ying. "Stochastic models of load balancing and scheduling in cloud computing clusters", In *Proc. IEEE Infocom*, pp. 702-710, 2012.

[10] Y. C. Hsu, P. Liu, and J. J. Wu. "Job sequence scheduling for cloud computing". In *Int. Conf. on Cloud and Service Computing (CSC 2011)*, 2011, pp. 212-219.

[11] Y. Fang, F. Wang, and J. Ge. "A task scheduling algorithm based on load balancing in cloud computing". In *Web Information Systems and Mining, Lecture Notes in Computer Science* , 6318, 210, pp. 271-277.

[12] B. Mondal, K. Dasgupta, and P. Dutta. "Load balancing in cloud computing using Stochastic Hill Climbimg-A soft computing approach", In *Procedia Tachnology*, 4, 2011, pp. 783-789.

[13] J. Hu, J. Gu, G. Sun, and T. Zhao,. "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment", In *3rd Int. Symp. on Parallel Architectures, Algorithms and Programming(PAAP)*, 2010, pp. 89-96.

[14] Y. Wei, and L. Tian. "Research on cloud design resources scheduling based on genetic algorithm", In *2012 Int. Conf. on Systems and Informatics (ICSAI 2012)*, 2012, pp. 2651-2656.

[15] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang. "Cloud task scheduling based on load balancing Ant Colony Optimization". In *6th Annual ChinaGrid Conf.*, 2011, pp. 3-9.

[16] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga. "A comprehensive survey: artificial bee colony (ABC) algorithm and applications", In *Artificial Intelligence Review 2012, Springer Science Business Media B.V. 2012*, March 2012.

[17] S. Bitam. "Bees life algorithm for job scheduling in cloud computing". In *Conf. on Computing and Information Technology* (ICCIT *2012)*, 2012, pp. 186-191.

[18] T. Mizan, S. M. R. A. Masud, and R. Latip. "Modified bees life algorithm for job scheduling in hybrid cloud", In *Int. Journal of Engineering and Technology(IJET)*, 2012, 2(6), pp. 974-979.

[19] R. N. Calheiros, R.Ranjan, C. A. F. D. Rose, and R. Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Software : Practice and Experience*, 41(11), 2011, pp. 23-50.

[20] R. N. Calheiros, R.Ranjan, C. A. F. D. Rose, and R. Buyya. "CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services", *arXiv preprint arXiv: 0903.2525*, 2009.

[21] M. Dorigo, C. Blum, "Ant colony optimization theory: Asurvey," in Theoretical Computer Science 344 (2–3)(2005), 2005, pp.243–278.

[22] M. Dorigo, M. Birattari, T. Stutzel, "Ant colony optimization," In IEEE Computational Intelligence Magazine, 2006, pp. 28-39.

[23] Huang L, Chen H, Hu T, "Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing1," Journal of Software, 2013, pp. 480-487.

[24] Tang R, Qin Y, Zhang L, "Research on Heuristics Logistics Distribution Algorithm Based on Parallel Multiant Colonies," Journal of Software, 2011, pp. 612-619.

[25] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R.Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization" In ACM SIGOPS Operating Systems Review, vol. 37, 2003, pp. 164-177.

[26] Liang Bai, Y.L. Hu, S.Y, Lao, W.M, Zhang, "Task Scheduling with Load Balancing using Multiple Ant Colonies Optimization in Grid Computing" In 2010 Sixth International Conference on Natural Computation (ICNC 2010), 2010, pp.2715-2719.

[27] B. Tang, Y.Y, Yin, Quan, Liu, Z.D. Zhou, "Research on the Application of Ant Colony Algorithm in Grid Resource Scheduling" In Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on IEEE, 2008, pp.1-4.