Available online at www.sciencedirect.com

ScienceDirect

Energy Procedia 141 (2017) 222–227

Energy Procedia

www.elsevier.com/locate/procedia

4th International Conference on Power and Energy Systems Engineering, CPESE 2017, 25-29 September 2017, Berlin, Germany

Energy Efficient Task Scheduling in Cloud Environment

R.K.Jena*

Institute of Management Technology, Nagpur, India

Abstract

Cloud computing is a style of computing in which dynamically scalable and other virtualized resources are provided as a service over the Internet. The energy consumption and makespan associated with the resources allocated should be taken into account. This paper focuses on task scheduling using Clonal Selection Algorithm (TSCSA) to optimize energy and processing time. The result obtained by TSCSA was simulated by an open source cloud platform (CloudSim). Finally, the results were compared to existing scheduling algorithms and found that the proposed algorithm (TSCSA) provide an optimal balance results for multiple objectives

Peer-review under responsibility of the scientific committee of the 4th International Conference on Power and Energy Systems Engineering.

Keywords: Task Scheduling, Cloud Computing, CloudSim, Clonal Selection Algorithm

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

* Corresponding author. Tel.: 91-9923960213
E-mail address: rkjena@imtnag.ac.in

1876-6102
Peer-review under responsibility of the scientific committee of the 4th International Conference on Power and Energy Systems Engineering.
10.1016/j.egypro.2017.11.096

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 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]. Many heuristic optimization algorithms have been developed and solved the task scheduling in cloud environment over the years [8]-[10]. Clonal Selection Algorithm (CSA) is a special class of Artificial Immune System which uses the clonal selection part of the Artificial Immune Systems as a main mechanism. This algorithm was initially proposed to solve nonlinear functions by De Castro and Van Zuben in 2000 [11,12].

This paper presents an optimization algorithm for user job scheduling to achieve optimization of energy consumption and overall computation time using Clonal Selection Algorithm (CSA). 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 Clonal Selection Algorithm (CSA) followed by Section 5 outlines the proposed task scheduling model based on Clonal Selection Algorithm (CSA). 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 [13-16]. In 2011, Hsu *et al*. [15] 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*. [17] used Stochastic Hill Climbing algorithm to solve load balance in Cloud computing. Hu *et al*. [18] 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*. [19] presented Genetic algorithm for scheduling in Cloud computing to increase the system performance. Li *et al*. [20] proposed a Load Balancing Ant Colony Optimization (LBACO) Algorithm to reduce makespan in Cloud. Karaboga *et al*., [21] presented ABC algorithm to solve the problem and find the most appropriate parameters in changing environment. Bitam *et al*. [22] proposed Bee Life algorithm for scheduling in Cloud. Mizan *et al*. [23] 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.. 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 Clonal Selection Algorithm (CSA) is proposed to optimize the energy and time.

3. Model development

To solve the problem of resource optimization using Clonal Selection Algorithm within the cloud framework, a typical cloud computing model is proposed. The cloud system consists of many data center that are distributed geographically all over globe and are accessible using internet. Each data center consists of many computing and saving elements and other resources. Processing Elements (PEs) in each data center are connected by a high bandwidth communication network. Therefore negligible communication delay is considered in this model. In the proposed model, user can access the cloud resources using user interface. The proposed task scheduling module in the framework is responsible for efficient allocation of user tasks into different available PE with an objective to optimize energy consumption and time.

3.1 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) ($D_1, D_2, D_3 \dots . D_M$) . 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 $\{P_1, P_2 \dots P_m\}$ 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 $G(V, E)$. 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. 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 datacentre. 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.

3.2 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:

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

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

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

where $T_{j=1 \dots n}$ 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:

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

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

$$\textbf{Minimize}\ \text{Makespan}_j \quad \text{j} = 1..\text{M} \tag{4}$$

$$\textbf{Minimize}\ E_j \quad \text{j} = 1..\text{M} \tag{5}$$

Subject to:

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

4. Clonal Selection Algorithm (CSA)

Artificial immune systems (AIS) are computation tools that emulate processes and mechanisms of the biological immune system. The immune system is one of the most important biological mechanisms humans possess since our life depends on it [24]. The clonal selection theory has become a widely accepted model for how the immune system responds to infection and how certain types of B and T lymphocytes are selected for the destruction of specific antigens invading the body. Clonal selection algorithm (CSA) is a special class of immune algorithms which are inspired by the clonal selection theory to produce effective methods for search and optimization [25,26]. CSA was first proposed by de Castro and Von Zuben [11] and was later enhanced and named as CLONALG [12]. CSA is not only an adaptive parallel algorithm based on the clonal selection theory but also represents an intelligent exploitation of a heuristic search in a vast feasible solution space. Once a new request for resource arrives, the system will run the CSA to adjust the overall allocation of the resources. Before finding the best solution with the CSA, we first change the mapping relationships between resources and tasks into a binary code as a set of initial population X(0). An individual $X_i^G = (X_{i1}^G, X_{i2,}^G, \ldots X_{ip}^G)$, where G denotes the current generation, i = 1, 2, …, s, and s denotes the population size. Each individual (antibody) means that a candidate solution is represented by a binary string of bits. The length of the bit string is suitably selected by the user to obtain a reasonable solution for the problem. Each gene in the chromosome is either 0 or 1. Once the initial population is generated, the affinity value of each individual is evaluated and stored for further operation. The CSA is applied in resource allocation to deal with the optimization problem, and the affinity function is designed in accordance with energy efficiency and makespan. The affinity function can be defined as follows [26].

$$aff(x) = e^{\min E_j + \min makespan_j} \tag{6}$$

The clonal operator is an antibody random map induced by the affinity. In the biological immune system, cloning means that a group of identical cells is generated from a single common ancestor and only antibodies with high affinity will be cloned to attack the pathogens. The antibodies are evaluated over an affinity function and sorted in decreasing order of affinity. Firstly, the affinity of each antibody is evaluated, and the ones with higher affinity are selected for the next generation. Then the selected antibodies proliferate into certain copies, and the copied and original ones are replicated in the current population. Afterwards, the antibodies in the population will implement mutation operation. The next section discuss the proposed algorithm in details.

5. Task scheduling using CSA

User submits their jobs into the system through a system interface. Each user job is assigned a timestamp indicating the time of arrival time of the job and added to the arrival queue of the system. Based on the number of available Data Center, the required number job is selected for execution using FCFS principle. A multi-objective CSA based method (TSCSA) is used to optimally schedule the tasks of the user jobs to Processing Elements of the respective data center(the details of the algorithm is excluded due to page constraints). In the proposed algorithm, user job is randomly distributed among available datacenter and tasks of each user job are optimally assigned to the PEs of each allocated datacenter by using CSA. The results of TSCSA algorithm are a set of Pareto solutions, providing a wide range of options to choose the best solution based on users degree of preference for a particular objective dynamically. The results of the TSCSA provide a pareto- archive (SS). So the last step of the algorithm provides an approach to pick up an optimal solution among the SS (Pareto archive according to the current requirement). A two dimensional vector $(2/\sqrt{2}\ ,\ 2/\sqrt{2})$ is introduced to represent the weighting for the objectives to find the optimal solution.

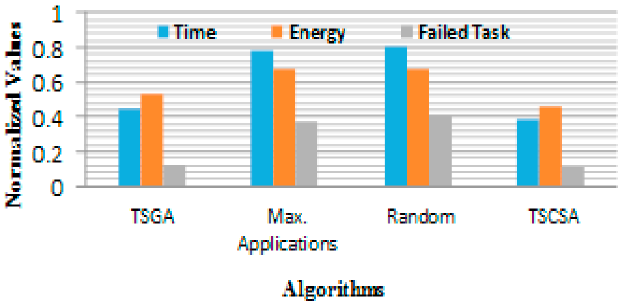
6. Implementation and result

The multi-objective CSA based framework is implemented using object oriented programming language C++. Experimental parameters are shown in table-4. CloudSim-3.0.1 is used to evaluate the scheduling of TSCSA. 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 .

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

Several experiments and with different parameter setting are performed to evaluate the efficiency and efficacy of TSCSA algorithm. Comparison between proposed algorithm (TSCSA) with genetic algorithm based TSGA, 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.



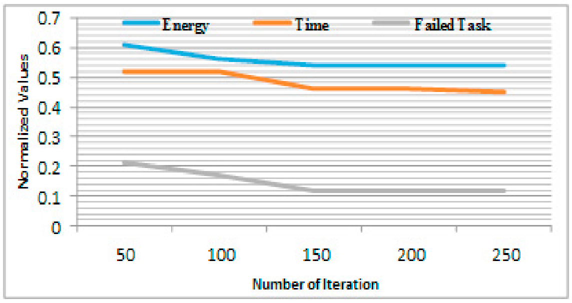


Fig. 1(b) : Optimal values with respect to number of Iteration

Fig. 1(a) : Comparisons of different approaches

Fig 1(a) shows a comparison of results between TSCSA, TSGA, Maximum Scheduling Algorithm (MSA) and Random Scheduling Algorithm (RSA). The proposed algorithm (TSCSA) reduced (10-30)% of energy consumption and (5-25)% of time (Makespan) in compare to other scheduling algorithm. The figure (fig. 2) also shows that TSCSA drastically reduced the number of failed tasks, which generally increase the profitability of the cloud environment.

The fig 1(b) shows the effect of optimal solutions with respect to increased number of iteration. The increased number of iteration improves the quality of solution up to a certain limit. The solution doesn't change much after that. Result shows (fig. 2) after 200 iterations the quality of solution doesn't improve significantly. Again the number of maximum iteration depends on the complexity of the scheduling i.e the number of user job, number of data center etc.

7. Conclusion

This paper presented multi-objective CSA 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, multi-objective CSA 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 (TSCSA) 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), pp. 1-13, Dec. 2004.

[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 5^{th} 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), pp. 899-911, 2001.

[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. A. Hofmeyr, S. Forrest, 'Immunity by Design: An Artificial Immune System,' Proceedings Genetic and Evolutionary Computation Conference, July 1999, pp. 1289-1296.

[9] L. N. De Castro, J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach. London, United Kingdom, Springer-Verlag, 1996.

[10] L. Hong, 'A Particle Swarm Optimization Based on Immune Mechanism,' 2009 International Joint Conference on Computational Sciences and Optimization, IEEE Computer Society 2009, pp. 670- 673.

[11] L. De Castro, F. J. Von Zuben, 'The Clonal selection algorithm with engineering applications,' GECCO 2000, Workshop Proceedings, Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA, 2000, 36-37.

[12] L. N. De Castro, F. J. Von Zuben, 'Learning and Optimization Using the Clonal Selection Principle,' IEEE Transactions on Evolutionary Computation, Vol. 6, No. 3, June 2002, pp. 239-251.

[13] 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, pp. 79-83, 2011.

[14] 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.

[15] 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)*,, pp. 212-219 , 2011.

[16] 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, pp. 271-277,2010.

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

[18] 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),* pp. 89-96,2010.

[19] 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)*, pp. 2651-2656,2012.

[20] 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.*, pp. 3-9, 2011.

[21] 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.

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

[23] 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.

[24] MG Gong, HF Du, LC Jiao, Optimal approximation of linear systems by artificial immune response. Sci. China F 49(1), 63–79 (2006)

[25] D Haifeng, J Licheng, Clonal operator and antibody clone algorithms, in Proceedings of the 2002 International Conference on Machine Learning and Cybernetics (Piscataway, IEEE, 2002), pp. 506–510

[26] Shu et al.: A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing. EURASIP Journal on Wireless Communications and Networking 2014 2014:64.