Task Scheduling in the Cloud Computing Based on the Cuckoo Search Algorithm

Nima Jafari Navimipour and Farnaz Sharifi Milani

Abstract— Cloud computing is the latest emerging trend in distributed computing, where shared resources are provided to end-users in an on demand fashion that brings many advantjuages, including data ubiquity, flexibility of access, high availability of resources, and flexibility. The task scheduling problem in Cloud computing is an NP-hard problem. Therefore, many heuristics have been proposed, from low level execution of tasks in multiple processors to high level execution of tasks. In this paper, a new evolutionary algorithm is proposed which named CSA to schedule the tasks in Cloud computing. CSA algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. The simulation results demonstrated that when the value of Pa is low, the speed and coverage of the algorithm become very high.

Index Terms—Cloud computing, task scheduling, cuckoo search algorithm and evolutionary algorithm.

I. INTRODUCTION

Cloud computing evolved through the recent advancements in hardware, virtualization technology, distributed computing, and service delivery over the Internet. While Cloud computing may not involve a lot of new technologies, it certainly represents a new way of managing IT. In many cases, this will not only change the workflow within the IT organization, it will often result in a complete reorganization of the IT department. Cost savings and scalability can be highly achieved from cloud computing [1]. The "Cloud" metaphor is a reference to the ubiquitous availability and accessibility of computing resources via Internet technologies [2]. Generally, cloud computing services can be categorized into three main types of services: Infrastructure as a Service, Platform as a Service and Software as a Service. These services can then be accessed through a cloud client, which could be a web browser, mobile app, and so on [3]. Cloud computing provides implementation agility, lower capital expenditure, location independence, resource pooling, broad network access, reliability, scalability, elasticity, and ease of maintenance [4].

The scheduling of a task workflow in a distributed computing platform is a well-known NP-hard problem [5]. The problem is even more complex and challenging when the virtualized clusters are used to execute a large number of

tasks in a Cloud computing platform [6]. For this reason, many heuristics have been proposed, from low level execution of tasks in multiple processors to high level execution of tasks in Grid and Cloud environments [7]. Recently, many papers are published which used evolutionary algorithms like genetic, ant colony, bee colony and PSO for optimization problems. Due to the advantages of Cuckoo Search Algorithm (CSA) [8], this paper addresses a task scheduling problem in a homogeneous Cloud infrastructure considering the minimization of the total waiting time of the tasks based on the CSA. CSA is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. Environmental features and the immigration of societies of cuckoos hopefully lead them to converge and find the best environment for breeding and reproduction [9].

The rest of this paper is organized as follows: the overview and background are reviewed in Section II; Section III, provides an introduction to cuckoo search algorithm; Section IV presents the cuckoo search algorithm for task scheduling in the Cloud systems; in Section V the simulation results are provided; and in the last section we will conclude the paper and suggest some indications for future researches.

II. RELATED WORKS

In this section, the state of the art mechanisms and approaches in task scheduling in the Cloud systems is briefly review and analyze. Pandey, Linlin [10] have proposed a Particle Swarm Optimization (PSO) based heuristic to schedule applications to Cloud resources that takes into account both computation cost and data transmission cost. Mezmaz, Melab [11] have investigated the problem of scheduling precedence-constrained parallel applications on heterogeneous computing systems like cloud computing infrastructures. They have proposed a new parallel bi-objective hybrid genetic algorithm based on dynamic voltage scaling (DVS) that takes into account, not only makespan, but also energy consumption. Abrishami and Naghibzadeh [12] have proposed a new QoS-based workflow scheduling algorithm based on a novel concept called Partial Critical Paths (PCP), which tries to minimize the cost of workflow execution while meeting a user-defined deadline. Wang, Zeng [13] have presented a Bayesian method based cognitive trust model, and a trust dynamic level scheduling algorithm named Cloud-DLS (Dynamic Level Scheduling) by integrating the existing DLS algorithm [14]. Laili, Tao [15] have combined the Service Composition Optimal Selection (SCOS) and Optimal Allocation of Computing Resources (OACR) into one-time decision in one console, named Dual Scheduling of Cloud Services and Computing Resources

Manuscript received June 3, 2014; revised October 9, 2014. This paper is selected from the 4th International Conference on Computer and Software Modeling, ICCSM 2014.

Nima Jafari Navimipour is with Department of Computer Engineering, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran (e-mail: jafari@iauasrb.ac.ir).

Farnaz Sharifi Milani is with Young Researchers and Elites Club, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran.

(DS-CSCR). Wu, Liu [16] have proposed a market-oriented hierarchical scheduling strategy in Cloud workflow systems. Tao, Feng [17] have proposed Case Library and Pareto Solution based hybrid Genetic Algorithm CLPS-GA for Optimal Scheduling of Computing Resources (OSCR). Liu, Qu [18] have proposed a fuzzy clustering method to pre-process the Cloud resources. They have proposed a new directed acyclic graph based scheduling algorithm called earliest finish time duplication algorithm for heterogeneous Cloud systems by combining the list scheduling with the task duplication scheduling scheme. . Choudhary and Peddoju [19] have proposed a scheduling algorithm which addresses the major challenges of task scheduling in the Cloud. In this algorithm, the incoming tasks are grouped on the basis of the task requirement like minimum execution time or minimum cost and prioritized. Then, resource selection is done on the basis of task constraints using a greedy approach. Finally, Jung, Lim [20] have presented the workflow scheduling scheme that reduces the task waiting time when an instance occurs the out-of-bid situation.

III. CUCKOO SEARCH ALGORITHM

CSA algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. Let us briefly review the interesting breeding behavior of certain cuckoo species. Then, we will outline the basic ideas and steps of the CS algorithm.

A. Cuckoo Breeding Behavior

Cuckoos are beautiful birds, but their aggressive reproduction strategy is more interesting to us. Cuckoos lay their eggs in communal nests which are chosen by levy flight. They often choose a nest where the host bird just laid its own eggs. To increase the probability of hatching their own eggs, they perhaps remove the other eggs [21]. Some female cuckoos imitate the colors and patterns of the eggs of host species. Cuckoos reduce the probability of the eggs being abandoned to increase their re-productivity [22]. Thus, when a host bird discovers a strange egg, either will throw them away or leave the nests and go somewhere else to build a new one [23]. A cuckoo often chooses a nest with some eggs in it. The next section briefly explains cuckoos levy flight.

B. Lévy Flights

The natural behavior of animals to search food is in a quasi-random manner. Recently, various studies have shown that flight behavior of many animals and insects has demonstrated the typical characteristics of Lévy flights [24]-[26]. Generally, the foraging path of an animal is effectively a random walk because the next move is based on both the current location/state and the transition probability to the next location. Choosing direction depends implicitly on a probability which can be modelled mathematically. Various studies have shown that the flight behavior of many animals and insects demonstrates the typical characteristics of Lévy flights [21]. Subsequently, such behavior has been applied to optimization and optimal search, and preliminary results show its promising capability [8], [27].

C. Cuckoo Search Algorithm (CSA)

CSA is a new meta-heuristic approach that models the

natural behavior of cuckoos. To describe the new CSA simplicity, the following idealized rules are [21]:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nests with high quality eggs (solutions) will be carried out over to the next generations.
- The number of available host nests is fixed, say *n*, and the host can discover an alien egg by a probability *Pa* [0, 1].
- The host bird can either throw the egg away or abandon the nest in order to build a completely new nest in a new location.

For simplicity, this last assumption can be approximated by a probability of the n nest pa. Based on these rules, the basic steps of the CSA can be summarized as the pseudo code as follows [21]:

Algorithm 1. Pseudo code of the CSA					
Objective function $f(x)$, $x=(x_1,, x_d)$;					
Initial a population of n host nests xi $(I = 1, 2,, n)$;					
while (<i>t</i> < Maximum Generation) or (stop criterion);					
Get a cuckoo (say I) randomly and generate a new					
solution by Lévy flights;					
Evaluate its quality/fitness; F_i					
Choose a nest among n (say j) randomly;					
if $(F_i > F_j)$,					
Replace j by the new solution;					
end					
Abandon a fraction (Pa) of worse nests					
[and build new ones at new locations via Lévy flights];					
Keep the best solutions (or nests with quality					
solutions);					
Rank the solutions and find the current best;					
End while					
Post process results and visualization					

In this algorithm x is a vector with entries $x_i^{(t+1)}$ indicating the next generation of cuckoo from x_i^t of i^{th} one is calculated by

 $X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus L\acute{e}vy$ (u)

Lévy
$$u = t^{\lambda}$$
, $1 < \lambda \le 3$ (2)

(1)

where $\alpha > 0$ is the step size, depending on the scale of the given problem, \oplus means entry-wise multiplications [28].

The consecutive steps of a cuckoo essentially form a random walk process which follows a power-law step-length distribution with a heavy tail. In the real world, if a cuckoo's egg is very similar to host's eggs, then this cuckoo's egg is less likely to be discovered. Therefore, it is a good idea to perform a random walk in a biased way with some random step sizes [29].

D. Fitness Value

and

Fitness or quality value shows how fit the solution is, i.e. how well it adapts to its environment. For a maximization problem, the fitness of a solution can be proportional to the value of the objective function [8]. For simplicity, we suppose that each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The purpose is to use the new and potentially better solutions (cuckoos) to replace a

not-so-good solution in the nests. Here, we use the simplest approach where each nest has only a single egg [8].

IV. PROPOSED APPROACH

In this section, the new method based on CSA is introduced to minimize the total execution time of tasks in Cloud system. As discussed in the previous section, population, host nests, fitness function, position of the nodes, discover probability, and levy flights steps must be defined for any CSA. The first step in any CSA is defining population and data representation. The population is initialized by a vector, in which the length of vector indicates the number of resources. For example, if there are 8 resources and 6 tasks, fig. 1. illustrate a possible solution.

R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
3	4	5	1	5	2	1	2
Fig. 1 One solution (nest) with is 8 resources and 6 tasks							

According to Fig. 1, T_1 is assigned to R_4 and R_7 ; T_2 is assigned to R_6 and R_8 ; and so on. First population is initialized randomly with 50 solutions. Equation 3 shows fitness function of each solution [30].

$$Fitness = \sum_{i=1}^{n} \sum_{j=1}^{r} R_{ij} \tag{3}$$

Where *n* is number of jobs, *r* indicates number of resources, R_{ij} indicates the time that the R_j responded to the $task_i$ request. Equation (3) evaluates the efficiency of each population.

x is a vector with entries $x_i^{(t+1)}$ indicating the next generation of cuckoo from x_i^t of i^{th} one is calculated by

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus \text{Lévy}(u)$$
(4)

and

Lévy
$$u = t^{-\lambda}, 1 < \lambda \le 3$$
 (5)

Algorithm 2. Pseudo Code of the Proposed Method
[Initialize]
Initial Population randomly
Choose best nest
Repeat steps 5 to 11 until termination condition
Perform get cuckoo by levy flight
Perform new nest
Fitness Function
Choose best nest
Perform best nest
Initial Population for the discover nest
Show the Best nest

where $\alpha > 0$ is the step size, depending on the scale of the given problem, \oplus means entry-wise multiplications. According to algorithm 2, after initializing first population randomly, the best nest is chosen. For the next generation, Lévy flight function is performed and opted new nest. Then fitness function is performed and the best nest is chosen. For nest discovering, new value is generated. Also the best nests are kept for next generations. This algorithm runs until no improvement in the fitness of the best member of the

population has been observed for 20 generations.

The next section shows the experimental results of the proposed method.

V. RESULTS

In this section, we present a results to evaluate the performance of the proposed method using Matlab. We suppose the number of resources in the network is 40 and the number of jobs are 120 where their execution and creation time are random in [0, 50]. Fig. 2 illustrates the execution time for this scenario when *Pa* is 0.2, 0.4 and 0.6.



Fig. 2. The execution time for the scenario when Pa is 0.2, 0.4 and 0.6.

The Fig. 2 shows that when the value of Pa is low, the speed and coverage of the algorithm become very high.

VI. CONCLUSIONS AND FUTURE WORKS

Cloud computing evolved through the recent advancements in hardware, virtualization technology, distributed computing and service delivery over the Internet. The task scheduling in Cloud computing is a well-known NP-hard problem. The problem is even more complex and challenging when the virtualized clusters are used to execute a large number of tasks in a Cloud computing platform. Therefore, many heuristics have been proposed, from low level execution of tasks in multiple processors to high level execution of tasks. In this paper, we proposed a new evolutionary algorithm named CSA to schedule the tasks in Cloud computing. CSA algorithm is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. We evaluate the proposed method using a Matlab tool. The result shows that when the value of Pa is low, the speed and coverage of the algorithm become very high. In the future, we plan to compare CSA with other algorithms like genetic and bee colony.

REFERENCES

- S. S. Manvi and G. K. Shyam, "Resource management for Infrastructure as a service (IaaS) in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 41, pp. 424-440, 2014.
- [2] T. Oliveira, M. Thomas, and M. Espadanal, "Assessing the determinants of cloud computing adoption: an analysis of the manufacturing and services sectors," *Information and Management*, vol. 51, no. 5, pp. 497-510, 2014.

- [3] H. Y. Chong, J. S. Wong, and X. Wang, "An explanatory case study on cloud computing applications in the built environment," *Automation in Construction*, vol. 44, pp. 152-162, 2014.
- [4] C. Gurkok, "Securing cloud computing systems," in *Network and System Security* (Second Edition), J. R. Vacca, Ed, 2014, Syngress: Boston, ch. 4, pp. 83-126.
- [5] N. J. Navimipour and L. M. Khanli, "The LGR method for task scheduling in computational grid," in *Proc. International Conference* on Advanced Computer Theory and Engineering, Phuket, 2008, pp. 1062-1066.
- [6] F. Zhang et al., "Multi-objective scheduling of many tasks in cloud platforms," *Future Generation Computer Systems*, vol. 37, pp. 309-320, 2014.
- [7] J. O. G. Garcia and K. M. Sim, "A family of heuristics for agent-based elastic cloud bag-of-tasks concurrent scheduling," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1682-1699, 2013.
- [8] X. S. Yang and S. Deb, "Cuckoo search via levy flights," in Proc. the NaBIC 2009, World Congress on Nature and Biologically Inspired Computing 2009, 2009, pp. 210-214.
- [9] C. A. Santos, P. K. Freire, and S. K. Mishra, "Cuckoo search via lévy flights for optimization of a physically-based runoff-erosion model," *Journal of Urban and Environmental Engineering*, vol. 6, no. 2, 2012.
- [10] S. Pandey et al., "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in Proc. 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010.
- [11] M. Mezmaz et al., "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 11, pp. 1497-1508, 2011.
- [12] S. Abrishami and M. Naghibzadeh, "Deadline-constrained workflow scheduling in software as a service cloud," *Scientia Iranica*, vol. 19, no. 3, pp. 680-689, 2012.
- [13] W. Wang et al., "Cloud-DLS: Dynamic trusted scheduling for Cloud computing," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2321-2329, 2012.
- [14] A. Dogan and F. Ozguner, "Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing," *IEEE Transactions on Parallel and Distributed Systems*, vol.13, no. 3, pp. 308-323, 2002.
- [15] Y. Laili *et al.*, "A Ranking Chaos Algorithm for dual scheduling of cloud service and computing resource in private cloud," *Computers in Industry*, vol. 64, no. 4, pp. 448-463, 2013.
- [16] Z. Wu *et al.*, "A market-oriented hierarchical scheduling strategy in cloud workflow systems," *The Journal of Supercomputing*, vol. 63, no.1, pp. 256-293, 2013.
- [17] F. Tao et al., "CLPS-GA: A case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling," *Applied Soft Computing*, vol. 19, pp. 264-279, 2014.
- [18] Z. Liu et al., "Resource preprocessing and optimal task scheduling in cloud computing environments," Concurrency and Computation: Practice and Experience, 2014.
- [19] M. Choudhary and S. K. Peddoju, "A dynamic optimization algorithm for task scheduling in cloud environment," *International Journal of Engineering Research and Applications (IJERA)*, vol. 2, no. 3, pp. 2564-2568, 2012.

- [20] D. Jung *et al.*, "A workflow scheduling technique for task distribution in spot instance-based cloud," *Ubiquitous Information Technologies and Applications*, Springer Berlin Heidelberg, pp. 409-416, 2014.
- [21] X.-S. Yang and S. Deb, "Cuckoo search via Levy flights," in Proc. World Congress on Nature & Biologically Inspired Computing, 2009, pp. 210-214.
- [22] R. B. Payne et al., The Cuckoos: Cuculidae, Oxford University Press, 2005.
- [23] D. C. Lahti, "Evolution of bird eggs in the absence of cuckoo parasitism," in *Proc. the National Academy of Sciences of the United States of America*, vol. 102, no. 50, pp. 18057-18062, 2005.
- [24] C. T. Brown, L. S. Liebovitch, and R. Glendon, "Lévy flights in Dobe Ju/'hoansi foraging patterns," *Human Ecology*, vol. 35, no. 1, pp. 129-138, 2007.
- [25] L. Flights, "Related topics in physics," *Lecture Notes in Physics*, Springer-Verlag, Berlin, 1995.
- [26] T. Solomon, E. R. Weeks, and H. L. Swinney, "Observation of anomalous diffusion and Lévy flights in a two-dimensional rotating flow," *Physical Review Letters*, vol. 71, no. 24, pp. 3975, 1993.
- [27] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver press, 2010.
- [28] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no.4, pp. 330-343, 2010.
- [29] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268-308, 2003.
- [30] N. J. Navimipour and A. M. Rahmani, "The new genetic based method with optimum number of super node in heterogeneous wireless sensor network for fault tolerant system," *International Journal of Computer* and Electrical Engineering, vol. 2, no. 1, pp. 99-104, 2010.



Nima Jafari Navimipour received his B.S. in computer engineering, software engineering, and the M.S. in computer engineering, computer architecture from Tabriz Branch, Islamic Azad University, Tabriz, Iran, in 2007 and 2009 respectively. In 2014, he received his Ph.D. degree in computer engineering, computer architecture from Science and Research Branch, Islamic Azad University, Tehran, Iran.

He has published more than 50 papers in various journals and conference proceedings. His research interests include cloud/grid systems, expert cloud, traffic control, traffic modelling, computational intelligence, evolutionary computing, and wireless networks.

Farnaz Sharifi Milani is a M.Sc. student in Department of Computer Engineering, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran since 2013. She received her B.S. degree in Information Technology Engineering from Tabriz Branch, Islamic Azad University, Tabriz, Iran in 2010. Her current research interests are focused on wireless sensor networks, peer to peer networks, cloud computing and network-on-chip.