Accepted Manuscript

ummejannatunnesajulie

June 2023

Accepted Manuscript
Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution Mohamed Abd Elaziz, Shengwu Xiong, K.P.N. Jayasena, Lin Li PII: S0950-7051(19)30032-2 DOI: https://doi.org/10.1016/j.knosys.2019.01.023 Reference: KNOSYS 4652 To appear in: Knowledge-Based Systems Received date : 11 July 2018 Revised date : 18 January 2019 Accepted date : 19 January 2019 Please cite this article as: M.A. Elaziz, S. Xiong, K.P.N. Jayasena et al., Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution, Knowledge-Based Systems (2019), https://doi.org/10.1016/j.knosys.2019.01.023
This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlight
• An alternative method for cloud task scheduling problem. • The proposed method is a modified moth search algorithm using Differential Evolution. • The Differential Evolution is used as local search method to enhance moth search algorithm. • The proposed method is evaluated twenty optimization functions. • It is used to solve the cloud task scheduling using synthetic and real trace data

Abastract
This paper presents an alternative method for cloud task scheduling problem which aims to minimize makespan that required to schedule a number of tasks on different Virtual Machines (VMs). The proposed method is based on the improvement of the Moth Search Algorithm (MSA) using the Differential Evolution (DE). The MSA simulates the behavior of moths to fly towards the source of light in nature through using two concepts, the phototaxis and Levy flights that represent the exploration and exploitation ability respectively. However, the exploitation ability is still needed to be improved, therefore, the DE can be used as local search method. In order to evaluate the performance of the proposed MSDE algorithm, a set of three experimental series are performed. The first experiment aims to compare the traditional MSA and the proposed

algorithm to solve a set of twenty global optimization problems. Meanwhile, in second and third experimental series the performance of the proposed algorithm to solve the cloud task scheduling problem is compared against other heuristic and meta-heuristic algorithms for synthetical and real trace data, respectively. The results of the two experimental series show that the proposed algorithm outperformed other algorithms according to the performance measures. Keywords: Task scheduling; Cloud Computing; Makespan; Moth Search Algorithm (MSA); Differential Evolution (DE); Meta-heuristics algorithm.

1 Introduction

The resources in cloud computing are shared between cloud clients through the virtualization concept [1]. Virtualization technology [2, 3] is one of the main features in cloud data centers which allow dynamic sharing of physical resources, permitting numerous applications to executing in different platforms called virtual machines (VMs) in a single physical server [4, 5]. Through the virtualization concept, a cloud provider can confirm the quality of service (QoS) distributed to the different users while accomplishing maximum resource utilization and minimum power consumption. Email address: abd el aziz m@yahoo.com, xiongsw@whut.edu.cn, pubudu@appsc.sab.ac.lk, cathylilin@whut.edu.cn (Lin Li*) 1 *Revised Manuscript (Clean Version) Click here to view linked References Optimal task scheduling is an important topic in cloud virtualization [6]. The task scheduling procedure is an NP-complete problem where, the time needed to locate the solution varies by the size of the problem. The task scheduling process [7] is a very essential issue in cloud computing where effective searching and decisions were involved to discover the best optimal VM for each task. There are various computation-based performance metrics used in scheduling procedure such as makespan, system utilization, response time and networkbased metrics such as traffic volume, network communication cost and round trip. Optimal scheduling of tasks in cloud computing [8] can be classified as heuristic, meta-heuristics and hybrid task scheduling approaches. The heuristic task scheduling algorithms deliver ease to schedule the task and deliver the best possible solutions, but it doesn't guarantee the optimal result. The meta-heuristic approaches can handle massive search space to discover optimal solution for task scheduling problem within polynomial time. The hybrid task scheduling techniques combined with both the heuristic and meta-heuristic. Recently, the moth search algorithm (MSA) has been proposed as new swarm algorithm [9], through simulating the behavior of moths in the nature where the moths are a family insects coupled with the butterflies are belonging to the order Lepidoptera. The simulation is performed through using the phototaxis and Levy flights. Where the phototaxis is considered as one of the most features of moths which used to represent the movement of moth towards or away from the light source. Also, it has been established that the characteristics of moths follow Levy flights [9]. In general, the light source represents the best moth (solution) inside the population, also, the moths which close to the best moth are fly

around their positions in way similar to the Levy flights. On the other hand, those moths that are far away from the best moth will fly toward it directly in a straight line and this is results from the phototaxis. The previous features of the moths are represent the exploration and exploitation abilities of moth search algorithm as a swarm algorithm. The MSA algorithm established its performance in [9], however, it suffer from some drawbacks such as, it can easy have stuck in local solution and its convergence becomes slow. These drawbacks result from the its exploitation ability is not as good as its exploration ability. This paper presents moth search algorithm (MSA) based DE method referred as MSDE for scheduling the tasks in VMs with balanced task distribution. The DE algorithm is used as local search technique, by this way, the exploitation ability of the MSA will be improved since the DE algorithm has been taken the properties of the genetic algorithm (GA) and the ES algorithm [10]. These properties are the large population of GA and self-adapting mutation of ES. Based on the properties it has been established its performance to improve the performance of other MH algorithms. For example, Zheng et al., [11] used the DE to enhance the performance of fireworks optimization algorithm to sharing the information among the fireworks and sparks. Also, in [12], the DE is combined with Harmony Search and used as multiobjective design of water distribution networks. Yuancheng et al., proposed a hybrid chaotic artificial bee colony differential evolution (CABC-DE) algorithm for Reactive Power Optimization problem [13]. The firefly algorithm is improved as global optimization algorithm through combining it with DE as presented in [14]. The major contributions of this paper are: • The first contribution of this work is to provide an alternative MH model based on improved MSA using DE algorithm for global optimization problem. 2 • The secondary contribution of this work is to design a task scheduling model based on the of MSDE algorithm. In order to achieve these goals, the performance of the MSDE algorithm is evaluated to find the global solution of a set of benchmark functions CEC2005 compared with MSA. In addition, the performance of proposed method as task scheduling in cloudsim is compared with PSO, Whale Optimization Algorithm (WOA) and some heuristics such as Round robin (RR) and Shortest Job First (SJF) for synthetic data. Moreover, the best approach to evaluate task scheduling strategies is applied to a realistic cloudlets (task) mixes. We evaluate the meta-heuristics algorithms by simulations using the CloudSim to create simulated cloud architecture consisted of identical heterogeneous VMs, physical machines, and with large number of cloudlets for real workload data.

2 Related Work

The heuristic and meta-heuristic methods are used to solve the task scheduling problem through discovering the optimal or near optimal solutions [8] . The heuristic algorithms [15] attempt to find the various solutions by applying problem features in a comprehensive way and work in dependent on the problem. The six rule based heuristic algorithms are applied in heterogeneous

and homogeneous cloud environments and schedule independent tasks with the objective of comparing their performance such as makespan and throughput, cost, degree of imbalance. The heuristic algorithms namely Minimum Completion Time (MCT), First Come First Serve (FCFS), Minimum Execution Time (MET), Maxmin, Minimum Completion Time (MCT), Sufferage and min-min are used in task scheduling in cloud computing. In [16], compare the various task scheduling policies consisting FCFS, SJEF, SJNF, LJEF and LJNF for cost optimization and resource utilization. Furthermore, [17] discuss five heuristic algorithms (Shortest Task First (STF), Greedy Scheduling (GS), FCFS, Sequence Scheduling (SS) and Balance Scheduling (BS))and evaluate the performance using CloudSim. The heuristic algorithms such as Opportunistic Load Balancing (OLB), MCT, MCT, Max-min, Min-min and Load Balancing Min-min (LBMM) and proposed Enhanced (LBMM) algorithm used in [18] for static task scheduling in cloud computing. Further, detailed descriptions of various task scheduling algorithms are illustrated for the cloud computing by [19]. Tabak, et al. [20] present a newly anticipated Min-min algorithm is combined with Max-min and Sufferage algorithm to reduces the execution time without affecting the quality of service. However, these previous methods easily stuck in local point because of the task scheduling problem is considered as a multi-modal problem. Therefore, the meta-heuristic can be used to avoid this limitation. The reason for using meta-heuristic (MH) techniques, since they are consist of simplicity, flexibility and ergodicity. Most MH algorithms are simple, easy to deploy and comparatively less complex. These algorithms are flexible to cover a comprehensive range of optimization problems which can't be tackle by classic algorithms. Ergodicity means MH algorithms can find multi-modal search spaces with adequate variety and avoiding local optima at the same time. The current research results [21, 22, 23, 24] shows that MH algorithms can provide efficient scheduling results than traditional scheduling algorithms. The MH approach [25] can be categorized into two ways such as bio-inspired (BI) and swarm intelligence (SI)-based MH. 3 The bio-inspired algorithms such as imperialism competitive algorithm (ICA)[26], Memetic algorithm (MA)[27] and Lion algorithm (LA)[28] are discussed in the task scheduling domain in cloud computing. A genetic algorithm is used for various kind of the task scheduling problems[29]. Keshanchi et al. [30], proposed an improved genetic algorithm with a heuristicbased HEFT named as N-GA used for static task scheduling in the cloud. Two-Stage-Task Scheduling Problem in Data-Centers solved by Johnson's-rule-based GA (JRGA) Algorithm [31]. In this model GA was used to assign tasks to suitable machines, and for decoding method to determine the correct order in which tasks are managed on each machine by the Johnson's Rule. Akbari et al. [32], improves the performance of genetic algorithm through significant changes and overview of new operators that assure sample diversity and reliable coverage of the whole space. For task scheduling, the memetic algorithm used in [33, 34], hill climbing and tabu search. Task scheduling decision is more complex in the heterogeneous cloud environment. Therefore, cloud scheduling methods that are based on SI techniques, for example, Ant Colony Optimization (ACO) [35, 36], Artificial Bee Colony (ABC) [37, 38] and Particle Swarm Optimization (PSO)[39, 40] are suit-

able. In order to enhance the the task scheduling for heterogeneous platform and reduce energy consumption ACO based task scheduling implemented for real-time sensor node [41]. Furthermore a lot of researches implement [35, 42, 43] ACO based task scheduling for green cloud computing platform. An efficient task scheduling is always needed for optimum resource utilization to overcome the situation of over or under resource utilization. Through [44], they proposed the cuckoo search-based task scheduling method for efficient task scheduling for minimize QoS response time. The task scheduling algorithm based on whale optimization algorithm [45] can efficiently schedule the tasks to the VM while reducing the makespan and cost. Therefore, recently, the majority of the proposed task scheduling algorithms have focused on hybrid meta-heuristic methods for task scheduling. In [46], they presented a hybrid meta-heuristic method by using HEFT(Heterogeneous Earliest Finish Time) algorithm combining PSO and GA. The existing methods are not suitable for many scenarios, also, the task scheduling problem is considered as NP-complete problem. In addition, according to the No-Free-Lunch (NFL) theorem that states that the same algorithm can't solve all problems with the same accuracy [47]. Therefore, [9] provides an alternative method to solve the task scheduling based on moth search algorithm. In general, the proposed algorithm depends on improving the performance of MSA algorithm using the operators of DE and it is called MSDE algorithm. It starts by generating a random integer population in which each solution represents the virtual machines where the tasks will be allocated. Then the fitness function for each solution is computed where in this paper the makespan is used as the main metric (fitness function). It is defined as the overall completion time of all the tasks that is optimally scheduled in the VMs. Thereafter, the population will divided into two subpopulations, the first one will fly directly toward the source of light. Meanwhile, the second one will be updated using either the Levy flight or DE according to the probability of each fitness function of the solutions belong to this subpopulation. Then the fitness function is computed again and the previous steps are repeated until the stop conditions are reached. In order to evaluate the performance of the proposed method a set of experimental series are performed. The main objective of the first experimental series is to assess the performance of the proposed algorithm to find the global 4 solution of a set of benchmark functions CEC2005 [48]. Meanwhile, the second experimental series are performed to solve the task scheduling problem with minimum makespan in cloud computing using synthetic and real trace data. Comparisons with other algorithms refer to the high superiority of the proposed MSDE algorithm over the other algorithms. The rest of the paper is organized as follows. Section 3 introduces the preliminaries about Task Scheduling problem, the Moth search algorithm, and Differential Evolution. The proposed MSDE method is introduced in Section 4. Meanwhile, Section 5 presented the experiments and comparisons. Finally Section 6 presents the conclusions.

3 Preliminaries

This section introduces the basic concepts of the task scheduling problem, moth search algorithm (MSA), and the differential evolution (DE). 3.1. Task Scheduling problem The cloud task scheduling problem can be defined as how to schedule and allocate various tasks to numerous virtual machines (VMs) practically and make all the tasks accomplished in a short execution time period. Considering the cloud system (CS) which consists of Npm physical machines (PM) and each physical machine consists of Nvm virtual machines (V Ms). CS = [PM1, PM2, ..., PMi , ..., PMNpm] (1) where PMi(i = 1, ..., Npm) denoted the PMs presented in the cloud and it can represent as: PMi = [V M1, V M2, ..., V Mk, ..., V MNvm] (2) where V Mk, k = 1, 2, ..., Nvm represents the k th virtual machine. Nvm is the number of virtual machines and V Mk denotes the k th virtual machine resource in the cloud environment. The feature of V Mk is defined as: V Mk = [SIDVk, MIP Sk] (3) Where SIDVk is the serial number of virtual machines and MIP Sk is the information processing speed of virtual machines (unit: millions-of-instructions-per-second, mips). T = [T ask1, T ask2, ..., T askl , ..., T askNtsk] (4) where Ntsk is the number of tasks l submitted by the users. T askl represents the l th task in the task sequence. The feature of T askl is defined as: T askl = [SIDTl , task lengthl , ECTl , P Il] (5) Where SIDTl is the serial number of tasks and task lengthl is the instruction length of the task (unit: million instruction). Time ECTl refers to the expected completion time for the T askl ; P Il refers to the task priority the number of tasks for Ntsk, the number of virtual machines for Nvm. The Expect Complete Time (ECT) matrix of size Ntsk ×Nvm denotes the 5 execution time required to run the task on each computing resource (virtual machine) that can be calculated by formula: ECT = ECT1,1 ECT1,2 ECT1,3 ECT1,4 ... ECT1,Nvm ECT2,1 ECT2,2 ECT2,3 ECT2,4 ... ECT2,Nvm ... ... ECTNtsk,1 ECTNtsk,2 ECTNtsk,3 ECTNtsk,4 ... ECTNtsk,Nvm (6) The main objective functions is to reduce the makespan by locating the best set of tasks to be executed on V Ms. ECTlk = task lengthl MIP Sk , k = 1, 2, 3, ..., Nvm, l = 1, 2, 3, ..., Ntsk (7) where ECTlk refers to the required execution time of lth task onkth VM where Nvm is the number of VMs and Ntsk is the number of tasks. The fitness value can be defined as [49]: f it = maxECTlk, l [1, Ntsk] mapped to kth V M, k = 1, 2, 3, ..., Nvm (8) 3.2. Moth search algorithm The mathematical model of the moth search algorithm (MSA) [9] is given in this section. Where the MSA simulates the behavior of the moths in natural by using the phototaxis and Levy flights which represents the exploration and exploitation of the algorithm, respectively. The MSA starts by generating a random population of moths (solutions) and then evaluate the quality of each moth using the fitness function. The moths that are nearest to the best moth with fly around it through using the Levy flights as in the following equation [9]: x t+1 i = x t i + Smax t 2 L(s), (9) where x t i represents the position of the i-th moth at iteration t. while Smax represents the maximum walk step and L(s) represents the step drawn from Levy flights, using parameter s, that defined using the gamma function as [9]: L(s) = (1)(1)sin((1) 2) s , s 0,

(10) where = 1.5 represents the parameter of Levy distribution [9]. Moreover, the moths that are far away from the source of the light they will fly in line towards this source or the towards the final position that is beyond the that source according to probability (P robs). x k+1 i = × (xi + × (xb xi)), if P robs 0.5 × (xi + 1 × (xb xi)), otherwise , (11) where xb represents the best moth position, while and are the scale and acceleration factor respectively.
Algorithm 1 Moth search algorithm (MSA) [9]

1: Input: The Number of solutions N, maximum number of iterations tmax, the dimension dim
2: Output: The best solution xb.
3: Initialize the maximum walk step Smax, the index , and acceleration factor .
4: Generate a random population X with size N and dimension dim.
5: Compute the fitness function for each solution xi X, i = 1, 2, ..., N.
6: while (t ¡ tmax) do
7: Sort the moths based on the fitness function values.
8: for i = 1 : N/2 do
9: Update xi using Levy flights as in Equation (9),
10: Compute the fitness function of xi.
11: end for
12: for i = N/2 + 1 : N do
13: Update xi using Equation (11).
14: Compute the fitness function of xi.
15: end for
16: t = t + 1.
17: end while

4 3.3. Differential Evolution

The basic concepts of the Differential evolution (DE) is introduced in this section [50] [51]. In general, the DE algorithm used three operators to improve the population, these operators are mutation, crossover, and selection. Considering the population X with size N is generated and the process of updating it using the three operators can be discussed as the following. The mutation operator is used to generate a mutant solution through combining a random solution vi with the difference between two other random solutions as in equation (12) [10]: v t i = x t r1 + × (x t r2 x t r3), (12) where represents the scaling factor ri , i = 1, 2, 3 are mutually exclusive random integers belong to [1, N], t is the current iteration. Thereafter, the DE used the crossover operator to generate an offspring solution u t ij , j = 1, 2, ..., Ntsk based on the mutant solution v t ij , and the current solution x t ij using the following equation: u t ij= v t ij if CP —j = jr x t ij otherwise (13) where jr [1, 2, ..., dim] represent random selected index, the CP [0, 1] is the crossover probability, and [0, 1] is random number. The selection operator is used to select the best solution from the offspring and

the current solution based on the fitness function to pass to the next iterations. This operator is defined as: x t+1 i = u t i if f it(u t i) f it(x t i) x t i otherwise (14) where f it(u) is the objective function value of the trial vector u