Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm

Nabila Tabassum

June 2023

Abstract

Cloud computing is the delivery of computing services over the internet. Cloud services allow individuals another businesses organization to use data that are managed by third parties or another person at remote locations. Most Cloud providers support services under constraints of Service Level Agreement (SLA) definitions. The SLAs are composed of different quality of service (QoS) rules promised by the provider. A cloud environment can be classified into two types: computing clouds and data clouds. In computing cloud, task scheduling plays a vital role in maintaining the quality of service and SLA. Efficient task scheduling is one of the major steps for effectively harnessing the potential of cloud computing. This paper explores the task scheduling algorithm using a hybrid approach, which combines desirable characteristics of two of the most widely used biologically-inspired heuristic algorithms, the genetic algorithms (GAs) and the bacterial foraging (BF) algorithms in the computing cloud. The main contributions of this article are twofold. First, the scheduling algorithm minimizes the makespan and second; it reduces the energy consumption, both economic and ecological perspectives. Experimental results show that the performance of the proposed algorithm outperforms than those of other algorithms regarding convergence, stability, and solution diversity.

1 Introduction

With the ubiquitous growth of Internet access and big data in their volume, velocity, and variety through the Internet, cloud computing becomes more and more proliferating in the industry, academia, and society. Cloud computing is composed of distributed computing, grid computing, utility computing, and autonomic computing [1]. Cloud computing provides on-demand computing and storage services with high performance and high scalability. Several computing paradigms have promised to deliver this utility computing. Cloud computing environment by considering different factors like completion time, the total cost for executing all users' tasks, utilization of the resource, power consumption, and fault tolerance. The problem of finding the right compromise between the resolution time and the energy consumed by a precedence-constrained parallel application is a bi-objective optimization problem. The solution to this problem is a set of Pareto points. Pareto solutions are those for which improvement in one objective can only occur with the worsening of at least one other objective. Thus, instead of a unique solution to the problem, the solution to a bi-objective problem is a (possibly infinite) set of Pareto points. Task scheduling has been proved as an NP-complete problem [2,3]. Cloud computing not just help a decent variety of uses, yet in addition give a virtualized condition for the applications to keep running in an efficient and minimal effort way [4]. A cloud data center usually consists of a large group of servers connected to the Internet. A task scheduler is needed in a cloud data center to arrange task executions. The task scheduler has to efficiently utilize the resources of the cloud data center to execute tasks. The performance issues of the scheduling algorithm includethe makespan and energy consumption. A good scheduler can use fewer resources and times to accomplish tasks execution. Using fewer resources implies that less energy is consumed. The minimization of energy consumption and makespan is one of the major issues for building large-scale clouds. There are different prospects of cloud computing has been studied to exploit the diversity of it viz. designing and implementing scheduling strategies and algorithms for specific tasks fault-tolerant tasks with real-time deadlines or energy efficient tasks such as dependent or independent. There is certain inherent problem associated with resource provisioning and task scheduling. The optimization goals, once set at the design time, will be statically built into the task scheduling and resource provisioning algorithm and implementation as the monolithic system component, thus lacking flexibility and adaptability in the presence of changing workload characterization, resource provisioning and cloud execution environment. Lots of task scheduling and resource provisioning strategy and algorithms, though intended with varied different optimization objectives, often contribute to some widespread functional mechanism and employ comparable software engineering framework for execution. However, adding new scheduling competence needs to be done for each scheduling algorithm one at a time, which is not only monotonous but also costly and leads to error. Natural selection tends to eliminate animals with poor foraging strategies through methods for locating, handling, and ingesting food and favors the propagation of genes of those animals that have successful foraging strategies, since they are more likely to obtain reproductive success. After many generations, poor foraging strategies are either eliminated or re-structured into good ones. Since a foraging organism/animal takes actions to maximize the energy utilized per unit time spent foraging, considering all the constraints presented by its own physiology, such as sensing and cognitive capabilities and environmental parameters (e.g., density of prey, risks from predators, physical characteristics of the Cloud computing environment by considering different factors like completion time, the total cost for executing all users' tasks, utilization of the resource, power consumption, and fault tolerance. The problem of finding the right compromise between the resolution time and the energy consumed by a precedence-constrained parallel application is a bi-objective optimization problem. The solution to this problem is a set of Pareto points. Pareto solutions are those for which improvement in one objective can only occur with the worsening of at least one other objective. Thus, instead of a unique solution to the problem, the solution to a bi-objective problem is a (possibly infinite) set of Pareto points. Task scheduling has been proved as an NP-complete problem [2,3]. Cloud computing not just help a decent variety of uses, yet in addition give a virtualized condition for the applications to keep running in an efficient and minimal effort way [4]. A cloud data center usually consists of a large group of servers connected to the Internet. A task scheduler is needed in a cloud data center to arrange task executions. The task scheduler has to efficiently utilize the resources of the cloud data center to execute tasks. The performance issues of the scheduling algorithm includethe makespan and energy consumption. A good scheduler can use fewer resources and times to accomplish tasks execution. Using fewer resources implies that less energy is consumed. The minimization of energy consumption and makespan is one of the major issues for building large-scale clouds. There are different prospects of cloud computing has been studied to exploit the diversity of it viz. designing and implementing scheduling strategies and algorithms for specific tasks fault-tolerant tasks with real-time deadlines or energy efficient tasks such as dependent or independent. There is certain inherent problem associated with resource provisioning and task scheduling. The optimization goals, once set at the design time, will be statically built into the task scheduling and resource provisioning algorithm and implementation as the monolithic system component, thus lacking flexibility and adaptability in the presence of changing workload characterization, resource provisioning and cloud execution environment. Lots of task scheduling and resource provisioning strategy and algorithms, though intended with varied different optimization objectives, often contribute to some widespread functional mechanism and employ comparable software engineering framework for execution. However, adding new scheduling competence needs to be done for each scheduling algorithm one at a time, which is not only monotonous but also costly and leads to error. Natural selection tends to eliminate animals with poor foraging strategies through methods for locating, handling, and ingesting food and favors the propagation of genes of those animals that have successful foraging strategies, since they are more likely to obtain reproductive success. After many generations, poor foraging strategies are either eliminated or re-structured into good ones. Since a foraging organism/animal takes actions to maximize the energy utilized per unit time spent foraging, considering all the constraints presented by its own physiology, such as sensing and cognitive capabilities and environmental parameters (e.g., density of prey, risks from predators, physical characteristics

2 Framework of task scheduling algorithm

To design and provide scheduling management framework for engineering implementation in IaaS Clouds, in this section, we introduce one of the important aspects in cloud computing resource management, i.e., task scheduling. The goal of cloud computing is to provide an optimal scheduling of the tasks, to provide the users, and the entire cloud system with optimal operation time, improved QoS at the same time and load balancing. Load balancing and task scheduling are closely related with each other in the cloud environment. Task scheduling is for the optimal matching of tasks and resources [14]. To design and provide scheduling management framework for engineering implementation in IaaS Clouds, in this section, we introduce one of the important aspects in cloud computing resource management, i.e., task scheduling. The cloud is mainly to provide users with a Quality of Service (QoS). The main aim of task scheduling algorithms is to achieve two main objectives namely, task scheduling helps to minimize the makespan and energy. Now we briefly depicts the entire energy aware task scheduling framework. This framework is composed of four main components: user portal, information service, task scheduler, and cloud data center with physical machines (PM) Fig. 1. The user portal provides an interface for users to submit task unit. The task unit further divided into small tasks to be executed in PMs. Information Service keeps the details of resource utilization and other log information to help scheduler to schedule a task to a PM in a data center

3 System models and definition

The model of the cloud system to be considered in this work can be described as follows [15]. We describe different models and definitions associated with problems formulated in this paper. In this paper, we have assumed that the cloud scheduling environment is highly heterogeneous and with the Physical machines have uncertain utilization information. We have designed multiple objectives of minimizing energy consumption and makespan. Under the favorable condition we find the Pareto set of multi-objective optimization.

3.1 Definitions

The cloud service provider keeps the detail information about the arrival of user requests and the available utilization of PMs in the data center. The complete scenario can be represented by using a direct acyclic graph where the user requests are presented. Here properties of tasks, task unit relationship and task unit arrived are captured.

Definition 1. User request (TD): This is the set of user request that consists of 1/n task units.

Definition 2. Task type (TS): It is the task type of each single task unit from 1/m, where Tm denotes maximum number of task in a task unit. Form example if we have three task unit fTD1; TD2; TD3g, then each task unit may have task type TD1 , TD2 .

Td1 TD2

TD3 TD4

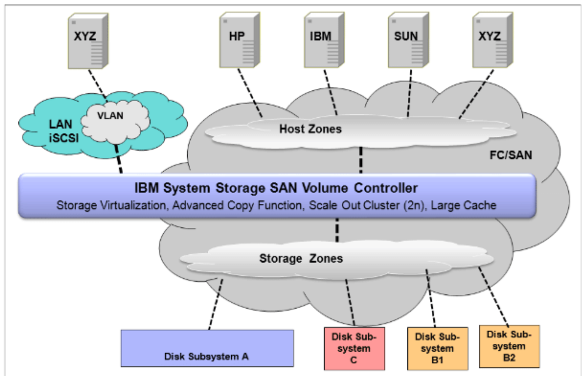

Figure 1: cloud computing

TS1	TS2	TS3 height
1	2	3
2	3	8

4 Problem formulation

From the framework of task scheduling algorithm, the following subsection first analyzes the relationship between make span and energy consumption then the objectives formulated. 4.1. A priori analysis The cloud operation comprises of five stages: the arrival of requests from users, resource exploration that facilitates the resource requirement, scheduling of resource this schedules the resources to execute users requests, service and process monitoring and the last one feedback submission. Out of these steps, the third one plays a vital role in the quality of service and total cost during the user request execution life cycle. The major factor that affects this stage is the makespan and energy consumption. To explore both makespan and energy efficiency, three crucial issues must be addressed. First, excessive power cycling of a server could reduce its

reliability. Second, turning resources off in a dynamic envi- ronment is risky from the QoS perspective. Due to the vari- ability of the workload and aggressive consolidation, some

PMs may not obtain required resources under peak load, and fail to meet the desired QoS. Third, ensuring SLAs brings challenges to accurate application performance management in cloud computing environments. All these issues require effective scheduling policies that can minimize energy

consumption and make span without compromising the user- specified QoS requirements. Currently, task scheduling in a

Cloud data center aims to provide high performance while

meeting SLAs, without focusing on allocating PMs to mini- mize energy consumption and makespan. Task scheduling in cloud computing is an NP-complete problem [18]. Authors in Ref. [17] propose an energy efficient resource scheduling algorithm that reduces operating costs and provides quality of service. Energy saving and resource utilization are achieved through scheduling of virtual machines. The QoS is modeled by the amount of resource needed for the CPU capacity measured in Millions Instructions Per Second (MIPS), the amount of primary memory (RAM), and by the network bandwidth rate. Insufficient available of these resource leads to SLA violation. In Ref. [19] authors proposes a load sharing optimization problem between a remote and a local Cloud service. Their

multi-objective approach defines to optimize energy con- sumption per job and response time using a Poisson arrivals

jobs rate.

Authors in Refs. [20e22] propose DVFS enabled tech- niques to minimize power consumption in a distributed system

ignoring performance. From the above study, we observe that it is a trade-off to minimize makespan and energy. So, minimization of both

these conflicting parameters at topological sorting to mini- mize both makespan and energy can be better realized as the

multi-objective optimization problem which is discussed below. 4.2. Problem objective In this section, the overall problem is defined and named as Multi objective optimization of task scheduling algorithm (MOOTS) problem. 4.3. Objective functions Based on Definition and Eqs. (1)e(8) given in previous section, our objectives are defined as follows. Objective (1): Minimization of makespan CTf Min CTn ebLB 9 Objective (2): Minimization of energy (TE) MinTE 10 Fitness function m 1/4 a

CTn ebLB

bTE 11 where 0 a¡ 1 and 0 b ¡1 are weights to prioritize components of the fitness function. 4.4. Constraints The following constraints are considered. Constraint 1. This constraint confirms that each task unit can only select one physical machine from the resource pool available in cloud data center. This is given as follows.

5 6. Principle of HBFA

The genetic algorithm (GA) has lack of local search capability and excellent global search. The ability of Bacteria Foraging (BF) global search is poor and very high local search capability [32e38]. When we combine these two algorithms

through the selective combination of certain favorable func- tion, this could give rise a solution that may contain best local

and global search capability and faster convergence time. The HBF posses all the merged properties of GA and BF [30,39,40]. The GBF appears to be an implementation of BF than GA, but it combines the features of both the algorithm.

Literature in reviewed that BF is hybridized with other algo- rithms other than GA and theoretically verified the effective- ness of the developed algorithm.

In all these literature it

has been observed that the HBF has maintained general val- idity and optimized features can be applied in many other

applications. The HBF inherits both swarming, and elimination and dispersal from BF. The objective is to make BF more global concerning search capability so we need to keep these features with HBF and change those function which does not support global search capability. The elimination and dispersal and swarming are the procedure which is critical in globalization search procedure hence they have been kept in the procedure. The other two functions, i.e., chemotaxis and reproduction are in focus to convert into global searching capability. The concept of genetic algorithm was introduced by Holland in 1973, which is inspired by biological evolution. It is a random search algorithm achieved by simulating natural

selection and genetic mechanisms. Genetic algorithm simu- lates the basic process of biological evolution with a string

of digital to analog the individual chromosome organisms,

and the basic processes of biological evolution through se- lection, crossover and mutation operators. The fitness func- tion represents the quality of the solutions, the average

fitness of each generation can be increased through the continuous upgrading of the population, the direction of the evolution of the population can be guided through fitness

function and on this basis we can make the solution repre- sented by the optimal individual approximate the global

optimal solution. 7. Proposed MHBFA for scheduling problem This section provides the proposed MHBFA bacteria foraging optimization algorithm to solve task scheduling in cloud computing. The evaluation of the objective function is proceeded in each iterative steps which lead to obtain better solution to the multi objective optimization problem. The position (coordinate) of the bacteria represents the parameters to be optimized. In a simple sentence, we can say that a bacteria represents the solution for our task scheduling problem. We have generated many bacteria for the algorithm input. The bacteria are evaluated against the objective function to obtain minimum makespan and energy. The parameters are discretized in the desirable range, each set of these discrete values represent a point in the space coordinates. These parameters are discretized in the desirable range. Each of these discrete values describes a point on a space coordinate. In the proposed MHBFA algorithm, at each iteration, all bacteria are evaluated according to a measure of solution quality. Our primary objective is to reduce makespan and energy consumption, which is bacteria position. The parameters used in algorithm's are as follows: p: Dimension of the search

6 Simulation results

In this section, we present the simulation result and discuss the performance of the proposed MHBFA solving MOO problem to verify its effectiveness by

	TAS	TAG	TDF height
6		6	9
5		89	90

Figure 2: Caption

comparing the results with other existing heuristic algorithms namely PSO [42], GA [15], BFA [27]. We have presented the simulation result using

[15,43,44] simulation with the Matlab R2013a software plat- form. Our simulation process is of two fold. In the first phase

we present experimental results and in second phase we make statistical result comparison. The parameter settings to conduct simulation for the algorithms are given in Table 5. We have considered three examples and simulated these using our simulation environment. This example consists of 5, 10 and 15 task unit with each task unit different tasks number and five physical resources (PM) for task scheduling and resource allocation on cloud computing. The detail parameter and values and the characteristics of PMs and task unit, that we have used for all our experiments is depicted in Table 6. We have the five-task unit with 23 tasks, ten task unit with 100 tasks and fifteen task unit with 200 tasks. Table 7 Task unit and its tasks details. TD1 TD2 TD3 TD4 TD5 TS11 TS21 TS31 TS41 TS51 TS12 TS22 TS32 TS42 TS52 TS13 TS23 TS33 TS43 TS53 TS14 TS24 TS34 TS44 TS54 TS15 TS35 TS45