

Updated Adaptive Incremental Genetic Algorithm for Task Scheduling in Cloud Environments*

Updated

1. M Babar Waseem

CS

SMIU

Karachi

mbabarwaseem@gmail.com

Abstract—Updated Cloud computing is a new commercial model that enables customers to acquire large amounts of virtual resources on demand. Resources including hardware and software can be delivered as services and measured by specific usage of storage, processing, bandwidth, etc. In Cloud computing, task scheduling is a process of mapping cloud tasks to Virtual Machines (VMs). When binding the tasks to VMs, the scheduling strategy has an important influence on the efficiency of datacenter and related energy consumption. Although many traditional scheduling algorithms have been applied in various platforms, they may not work efficiently due to the large number of user requests, the variety of computation resources, and the complexity of Cloud environment. In this paper, we tackle the task scheduling problem which aims to minimize makespan by Genetic Algorithm (GA). We propose an incremental GA which has adaptive probabilities of crossover and mutation. The mutation and crossover rates change according to generations and also vary between individuals. Large numbers of tasks are randomly generated to simulate various scales of task scheduling problem in Cloud environment. Based on the instance types of Amazon EC2, we implemented virtual machines with different computing capacity on CloudSim. We compared the performance of the adaptive incremental GA with that of Standard GA, Min-Min, Max-Min, Simulated Annealing, and Artificial Bee Colony Algorithm in finding the optimal scheme. Experimental results show that the proposed algorithm can achieve feasible solutions which have acceptable makespan with less computation time.

Index Terms—Updated cloud computing, Infrastructure as a Service, genetic algorithm, Updated task scheduling

I. UPDATED INTRODUCTION

Updated As the National Institute of Standards and Technology (NIST) defines, cloud computing [1] is a new paradigm that provides configurable resource pool by network access. Resources such as networks, servers, and storage can be rapidly provided and released without service provider getting heavily involved. Since Cloud computing technologies offer scalability, are reliable and trustworthy, and offer high performance at relatively low cost, they have a variety of application domains [2], such as virtual computing labs [3], linguistic group decision-making [4], three-dimensional reconstruction [5], etc. In the Cloud model, services including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software

as a Service (SaaS) are available to consumers with various demands. Among these three service models, IaaS has the capability to provide Virtual Machines (VMs) and enables customers to control the computational resources or networking components without caring about the location, maintenance or management of the physical devices. In general, IaaS can meet the requirement of performing complex tasks or large applications and measure the service according to the Cloud pricing schemes. However, the process of mapping tasks to VMs, namely task scheduling, becomes more complex due to the large number of physical servers, the complex Cloud environment, the variety of consumer requirements, and different Service Level Agreements (SLA) compared with traditional distributed or heterogeneous computing environment.

The task Scheduling problem in Cloud, which is known to be NP-hard, is assigning different tasks to corresponding resource node under the Quality of Services (QoS) constraints. Cloud task scheduling is an NP-hard problem. For an NP-hard algorithm, it is generally believed that no algorithm exists that solves each instance in polynomial time [6]. Some traditional task scheduling algorithms have been applied in heterogeneous computing environments such as Min-Min [7], Max-Min [8], etc. Min-Min algorithm allocates tasks with shortest completion time to the corresponding machine. By contrast, Max-Min algorithm prefers to choose larger tasks which can cause smaller task delays for long time. Although these algorithms are simple and can be easily transplanted to the Cloud environment, they may lead to lower efficiency when the number of tasks is very large.

TABLE I
SAMPLE CAPTION

	Hello		Hi	Apna	Bata	Bhai	Apna
Naam	Bata	Bata	Bhai	Apna	Bata	Bhai	Apna
	Sho	Mjaa	Maa				
	Hello				Hi		
Naam	Bata	Bhai	Apna				
	Sho	Mjaa	Maa				

Updated Identify applicable funding agency here. If none, delete this.

II. UPDATED ADAPTIVE INCREMENTAL GENETIC ALGORITHM

Updated Genetic Algorithm (GA) is a bio-inspired algorithm which belongs to Evolutionary Algorithms (EA). Genetic algorithm is based on the idea of Darwin's Theory of Evolution and Mendelian Genetics. Darwin's Theory of Evolution indicates that, during the process of biological evolution, individuals who are more able to adapt to the environment have a higher probability of survival; Mendel's genetic theory demonstrates that genes are kept by chromosome in the form of genes, and gene mutation or chromosome crossover can generate new features for individuals. Those genetic structures with high adaptability are easier to be retained.

The main operators of GA include selection, crossover, and mutation. In GA, the select operator imitates natural selection. The purpose of selection is saving "good" individuals. The larger the fitness value is, the higher the probability that allows the individual to enter into the next generation. On the contrary, individuals with smaller fitness value will be less likely to be kept. Selection strategies affect the direction of evolution and convergence rate of the population. Crossover is the process of gene recombination. Ideally, the offspring produced by the crossover of two parents should inherit the excellent features of their parents. Mutation operation produces new individuals by changing a few genes. It contributes to the diversity of the population and also plays an important role in approaching the optimal solution.

When using GA to optimize different problems, each chromosome represents a candidate solution. Fitness is defined to evaluate the solution. A series of genetic operations (crossover, recombination, mutation, etc.) are used to generate new chromosomes, while select operators will choose better solutions to retain. After a number of generations, the group converges to the individuals who are most adaptive to the environment, that is, the optimal solution.

To address the task scheduling problem, the chromosome can be encoded with a real number. The chromosome represents a feasible solution. Considering the complex cloud environment, the datacenter schedules a great number of tasks in the request list. Standard GA can cost a long time to search for the optimal solution because evaluation, crossover, and mutation operators involve more computing. In this paper, an incremental method is proposed to tackle this problem. The tasks are divided into some independent sets, and then every time some tasks are chosen to schedule. Details about the proposed adaptive incremental GA are as follows.

My Favorite Fruits

- Apple
- Banana
- Cherry
- Date
- Fig

A. Updated Encoding

Updated Encoding strategies such as binary encoding, real number encoding, and symbol encoding have been applied on



Fig. 1. Geeks for Geeks

different problems. Binary coding maps solution space to a bit string, which is a simple and common encoding method. Similarly, a real-valued chromosome is represented by a real number in the decision space. In the task scheduling problem, the chromosome is encoded by a real number. The length of a chromosome equals the number of tasks. The gene's value represents to which VM the task is allocated.

B. Updated Fitness Function

Updated The fitness function is used to evaluate the quality of the candidate solution, which is an important index for selecting the best individual. Usually, the value of the objective function can be directly used for fitness measurement, or define the fitness function according to the specific problem. As given in Equation (3), the fitness is calculated by the makespan of a solution. The maximum finish time of among all the VMs is the fitness value.

C. Updated Select

Updated There are several selection strategies such as roulette, tournament, and sorting selection. The roulette method firstly computes the probability of selection according to each individual's fitness (the proportion of the individual's fitness to the sum of all individual fitness values). Then, the disk is divided into N copies, each with a central angle which is proportional to the probability of selection. Choosing an individual depends on a randomly generated number falling into which region. Tournament selects a certain number of individuals, and the individual with highest fitness will be chosen. This process continues until the number of individuals reaches a pre-set size. Sorting selection firstly calculates each individual's fitness value. After sorting by fitness, the solution will be assigned a probability of selection. Then, the selection is determined by the order of distribution.

D. Updated Crossover

Updated There are some crossover strategies such as single-point crossover, two-point crossover, uniform crossover and so on. An example of single-point crossover is shown in Figure 1.