Global Conference on Computer Science, Software, Networks and Engineering
NOVEMBER 06– 08, 2014
Efes Sürmeli Hotel & Convention Center – İzmir, Kuşadası, Turkey

An Evolutionary Approach For Scheduling Tasks Based on Lottery Algorithm in Cloud Computing Environment

Authors : Elahe doostsadigh - Reza Asemi Tazekand - Elnaz Ashna

Abstract

Cloud computing as a pattern for distributed computing, are composed of large shrimp ask combined resources with the goal of resource sharing as a service, on the internet. Such resources as in memory, processor and services are always worth and more efficient use of this; is endless challenge. Hence the scheduling of tasks in cloud computing is very important that try to determine an efficient scheduling and source allocation.

In fact, the goal is determining a processing resource from set of resources that a task needs for process, so that can process more jobs in less time. Scheduling system controls different functions in cloud system for increasing job completion rate, resource efficiency and in consequence increasing the computing power. In this study, we provide approach base on lottery algorithm to reach those goals with minimize make-span time. Simulation of proposed method is by "CloudSim" application.

Keywords: cloud computing, task scheduling, make-span time, lottery algorithm, virtual machine

1-Introduction

In recent years, with the growing amount of information processing, need for distributed systems and parallel processing has been felt more than before. This technology uses communication infrastructures and computer networks for remote access to variety of resources. We can link heterogeneous software and hardware resources together without any processing limitation as the whole structure is seen as virtual machine, then run complex Application which needs for high power processing and huge data, on the virtual machine. Indeed, objective is to use idle processing resources for implementation of great things [1].

In fact, cloud computing as a pattern for distributed computing are composed of large shrimp ask combined resources with the goal of resource sharing as a service on the internet [2].

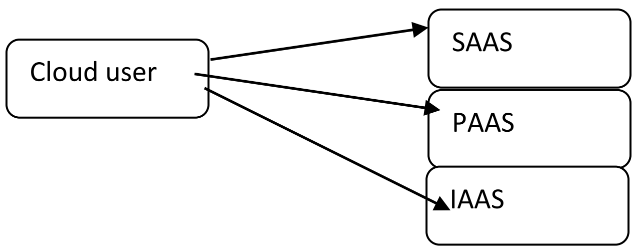


Figure1 -services in the environment

Services in cloud computing environment provide in 3 levels:
SaaS: software as a service
PaaS: platform as a service
IaaS: Infrastructure as a service

This category causes supporting for different management for each level and virtualization technology [3]. Figure1 shows the variety of cloud computing services .in cloud environment processes allocated on the pole of dynamic and virtual resources based on users need on the online mode. Task scheduling is a key process in IaaS with this adjective: running request entered into the system on resources in efficient manner with considering specificities of cloud environment.

Task scheduling, considers virtual machines as scheduling machines for dedication the physical resources running tasks. Each virtual machine is abstract unit of computing capacities and provided storage in cloud because of dynamically made and heterogeneous of cloud computing environment, discuss about scheduling tasks is considered as an NP-Complete problem. In this type of system, scheduling process must perform rapidly automatically.

2-Scheduling

The policies in cloud environment are depended to cloud environment properties. These are followings: dynamic requests of users, infrastructure changes of network as in bandwidth, resource cost, time to live, etc. If you want a summary for some of the activities that have done in task scheduling domain, first we can remind basic algorithms in cloud computing then introduce some of papers that proof one or more QOS properties.
Kinds of task scheduling in cloud computing have two categories:
1) “Batch Mode Heuristic Scheduling Algorithms” (BMHA)
2) online mode heuristic algorithms
In BMHA the jobs are logged in temporary, task scheduling and queuing and are allocated in a set as FCFS, RR, Max-Min and Min-Min.
Some another algorithms used are those that are aware of the source (RASA) which in [8] addressed in detail.
it is the result of combining two algorithms Max-Min, Min-Min and experimental results have shown the RASA is better than scheduling algorithms of distributed algorithms in large scales. One efficient model for scheduling policy of services based on cloud environment have mentioned in [31] that based on admission control scheme and priority. The main advantage is policy suggested with cloud architecture for completion rates, high service(99%) with QOS warranty.
In [12] enhanced cost-based algorithm for Task scheduler provided by the doctor Suadhai and his associates that measures both source costs and computing performance and improves rate of calculations. After summarize the works done by usual cloud computing algorithms, it turns heuristic algorithms, ACO and SA(Simulated *Annealing)* optimization methods together with the classic methods have demonstrated considerable access. In [24] a fusion algorithm is combined with the proposed two-stages:
First stage is the same genetic algorithm stops after the last iteration, where the final product was obtained as an input to the second algorithm (fusion). In [25] a dihedral algorithm from ACO and genetic was introduced under the title of TLDA. This means that the timing of the work, ACO algorithm is based on the grid does, then result of this scheduling algorithm is given to GA algorithm as sample space.

1- with reviewing previous works, we realized scheduling issue and efficient use of time in processes, more attention challenged day after day. All seals are increasing studies confirm this.

3- The proposed method

As noted earlier in task timing issues in cloud computing, the output is a proper mapping of tasks to resources. So that parameters such as response time, make-span time and performance of datacenters, are optimized. In this report, we present a new algorithm based on lottery algorithm. The proposed algorithm has evolutionary view and most prominent characteristic of it is being agile. Stages of it are as follows:

First step: number of answers are created randomly, which either is discussed as a participant in proposed method.
Second step: the propriety of each participant is measured with 3-3 and 3-4

$$F_i = \left(T^{len}/R^{w}\right) + N^{cc} \qquad (3\text{-}3)$$

$$Fitness_{total} = \left(\sum_{t0}^{n} F_i\right)/1 \qquad (3\text{-}4)$$

$F_{i=}$value obtained for task i

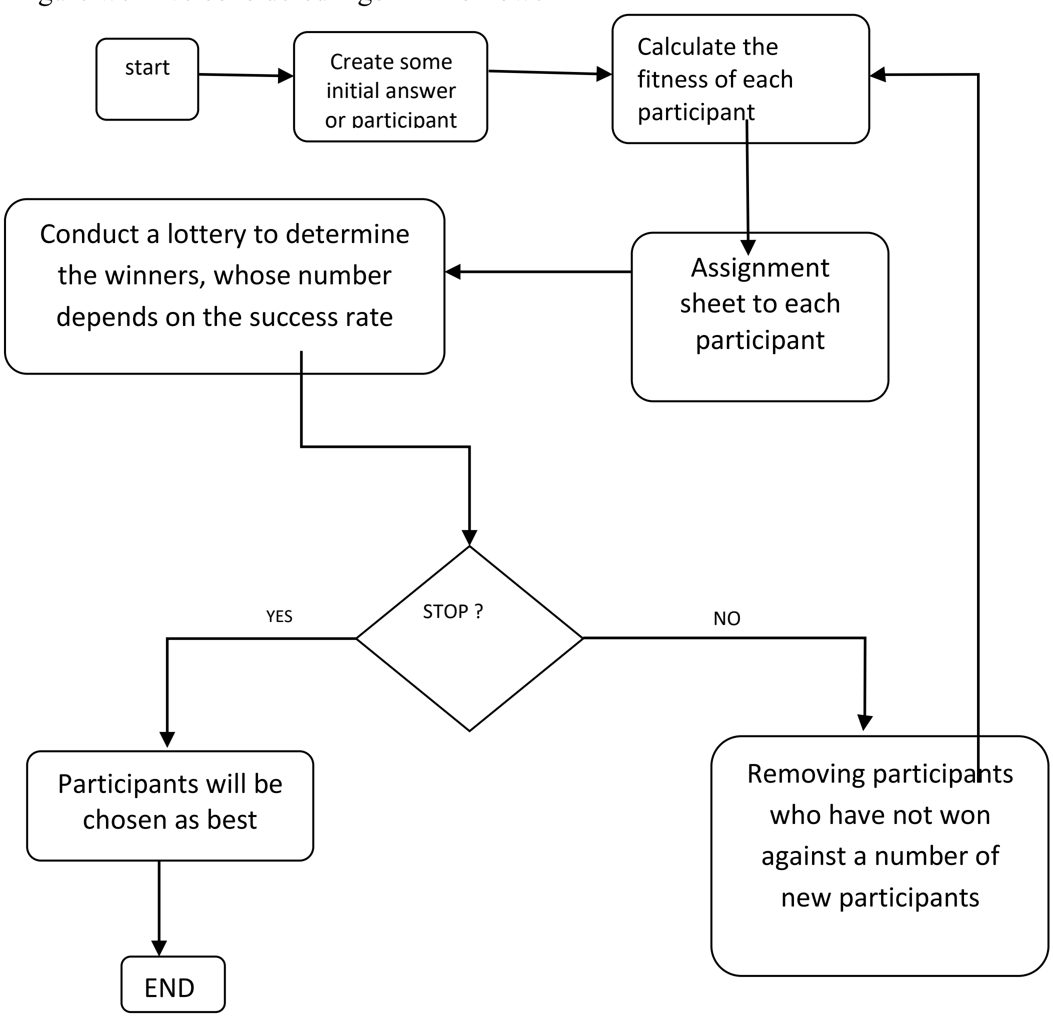
T_{len}=length of i_{th} task
R_w=workloads that are already on the CPU which i_{th} task is allocated for it
NCC=communication cost for the selected virtual machine for i_{th} task or databroker
Fitness $_{total}$=the answer fitness

Third step: each participant assigned a fitness based on the number of sheets. Indeed, participants who had more points will have more sheets.

Forth step: lottery is done. There is one win rate equal to 0.8. indeed, what it means is that in each iteration 80% of current stage's participants are transported to new stage. 20% of the initial population consist of participants. For lottery, we have used randomly numbers with uniform dispatch.

Fifth step: we check end condition in this stage. In our proposed method, the end condition is specific number of repetition, but we can consider the end condition near an optimal condition.

In figure we have considered algorithm's flowchart



4-Simulation

In this paper we used CloudSim simulator for our implementations. This application is a java based toolkit, that is composed some classes. For complete evolution, we did algorithm's simulation in five different situations which table 4-1 shows number of tasks and virtual machines in 5 mode as A,B,C,D,E.

In table 2-4 each column represents the conditions. The row with name ”CPU(mips)” shows power of CPU in mips. The row with name “RAM(mips)” shows the amount od main memory, the row about NCC shows amount of memory for communication with no unit of measurement but for each NCC unit, one hundredth of bandwidth is low.

Here NCC can be any cost to transfer. Characteristics of scenarios selected according to [f1,f2] studies, and has been tired in various aspects of processing algorithms to be tested.

Table 1-4 numbers of tasks and VMs

E	*D*	*C*	*B*	*A*	
1000	*500*	*60*	*60*	*20*	*Number of tasks*
50	*50*	*12*	*7*	*7*	*Number of VM_s*

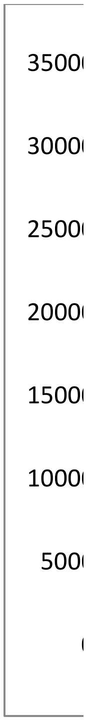
Table 2-4 virtual machines and task features

					Virtual machines properties and tasks
E	*D*	*C*	*B*	*A*	
200_3000	*10_500*	*10_500*	*10_500*	*10_500*	*Cpu(Mips)*
2000_20000	*100_2000*	*10_5000*	*10_5000*	*10_5000*	*Ram(MB)*
10_1000	*100_500*	*10_500*	*10_500*	*10_500*	*Ncc*
100_30000	*500_3000*	*1000_100000*	*1000_100000*	*1000_100000*	*Task(MIPS)*

Experimental results

Implementation of this research has been done, are according to the configuration described in section IV. In this section, we have established suggested algorithm about Make-span time, average of response time which will show this result.

Figure 1-4 Make-Span Time diagram which got from implementation



In figure 1-4, the horizontal axis illustrates scenarios whom has detailed in tables 2-4 and 1-4. Vertical axis says make-span time in millisecond that of course is much better than. As can be seen, our proposed algorithm had better results.

Table 4-6 presents the recovery percentage of proposed algorithm raised to the three algorithms of this study, in terms of make-span time. It is gotten from this:

$$Improvment = \frac{(\text{Result} - \text{ Propsal Approach }) * 100}{\text{Result}}$$

In the above equation, “result” is the make-span time of algorithm that we want it to show improvement over the proposed method. Figure 3-4 illustrate response time average, which it has gotten in various scenarios in each algorithm. As is clear from figure 3-4, suggested algorithm has better answer than two other algorithms GA and PSO of response time. PSO algorithm is more than locally efficient has caught and could not get efficient response.

6-conclusion

As mentioned in the previous section, in this research, discussed methods examined in two aspects make-span and response time were studied. PSO algorithm had most make-span time and this is due to PSO algorithm has been caught in local optimal and has failed to search the problem space. Genetic algorithm had fewer make-span time than PSO, has searched problem space better than PSO it has less involved in local optimal and proposed algorithm, which was less involved in local optimal, better than two other methods.

The advantage of proposed method is agility and simplicity, which they are deficiencies of other algorithms and we have special attention to it.

Second point which previous section’s tables and diagrams show, there is a direct relationship between make-span time and average of response time. Using other numerical distribution for lottery stage, can be efficient.