Review and Comparison of Tasks Scheduling in Cloud Computing

Yousef Tohidirad¹, Zahed Soltani aliabadi², Abdolsalam Azizi³, Siamak Abdezadeh⁴ and Mohammad Moradi⁵

¹Department of Computer Engineering, Electronic Branch, Islamic Azad University, Tehran, Iran
²,³Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
⁴Department of Computer, Boukan Branch, Islamic Azad University, Boukan, Iran
⁵Department of Computer, Baneh Branch, Islamic Azad University, Baneh, Iran

Abstract

Recently, there has been a dramatic increase in the popularity of cloud computing systems that rent computing resources on-demand, bill on a pay-as-you-go basis, and multiplex many users on the same physical infrastructure. It is a virtual pool of resources which are provided to users via Internet. It gives users virtually unlimited pay-per-use computing resources without the burden of managing the underlying infrastructure. One of the goals is to use the resources efficiently and gain maximum profit. Scheduling is a critical problem in Cloud computing, because a cloud provider has to serve many users in Cloud computing system. So scheduling is the major issue in establishing Cloud computing systems. The scheduling algorithms should order the jobs in a way where balance between improving the performance and quality of service and at the same time maintaining the efficiency and fairness among the jobs. This paper introduces and explores some of the methods provided for in cloud computing has been scheduled. Finally the waiting time and time to implement some of the proposed algorithm is evaluated.

Keywords

Cloud Computing, scheduling algorithms, Scheduling Management, Virtual Machine (VM), CloudSim.

1. Introduction

Modern communication has been termed as ‘viral exchange of information’. This follows to the effect that the contemporary world is subject to internet communication, which sees people share information in a ‘ghostly’ manner. In the actual sense, the ‘people’ who communicate during this process are the machines involved, mostly the computers and servers [1]. Users of the internet have seen continuous manipulation of data that they share. This aspect is controlled by the fact that users are many, thereby making the computers to be subject to heavy task scheduling protocols. This aims to have many people access the internet with ease and in the required bandwidth. The interconnection of these processes, despite the difference in the computer processors, is termed as Cloud Computing.

Cloud computing is a technique that uses the most elemental issues of information sharing. In this manner, there is a rightly developed infrastructure as well as the interrelating services. Different relating concepts are described within the aspects of the process. A number of computers are always connected in the network in real-time [2]. Users have the ability to reach the GUI features without knowing the applied expertise on the same. This is well laid under the principle of
visualization. Visualization is the main intention of the technology with the intention of running numerous VMs on a single machine with the required specifications of load balancing. In this way, a good task scheme is used to expand the ability for the load balancing under task scheduling within the specified distribution system. There is an ABC presence inspired by the life of the honeybee. To improve the performance of the process, there are created set of behaviours and techniques designed by the required personnel to utilize the given designs under the designs of the ABC algorithm [3].

2. MATERIALS AND METHODS

2.1. Basics of ABC

As a concern of VM scheduling, ABC algorithm has stood as the ideal measure for its success for most special implementations [4]. While being used from time to time, the algorithm has been effective for the establishment of the right value. While working under the principle of the honeybees, information sharing on machines is based on source details, destination, network quality, player of the network, threats and design tricks of the network. This can be describes by the ABC pseudo code shown in Figure 1 [1].

![ABC pseudo code/ basic algorithm of ABC aimed at having tasks scheduled](image)

Figure 1. ABC pseudo code/ basic algorithm of ABC aimed at having tasks scheduled [1]

2.2. ABC in application

The major intention of Cloud Computing is the existence of self-motivated a pool of VM resources [5]. The VM requirements are defined according to the system’s users. This is done in accordance with the expected design specifications. Using the acceptable Cloud Management policies, possibility for routing specifications is established. This aspect is done within the limits of the available servers while looking at the cloud management. Under the same specifications, it is possible have different degrees of scheduling management while considering the load levels within VMs. The balancing along this line should leave the best room for the reduction of makespan and response time.
Since the machines have the general outlook of virtual presence, it is good to represent them under a set of numbers for the resultant or overall virtuosity as shown in the equation (1) [1].

\[
VM = \{VM_1, VM_2, VM_3, \ldots , VM_N\} \quad (1)
\]

These machines are subject to individual tasks that result into an overall system task as shown in the equation (2) [1].

\[
Tack = \{tack_1, tack_2, tack_3, \ldots , tack_K\} \quad (2)
\]

The machines work in tandem (parallel) using different processors. In this way, the scheduling ability is non-preventive. As a result, figure 2 shows the diagram a flowchart of the task scheduling under the ABC instructions [1].

![Flowchart of task scheduling](image)

Figure 2. ABC instruction and the scheduling modality [1]
In the second process, evaluation of the population’s fitness is a calculation based on the formula (3) [1].

$$\text{fit}_j = \frac{\sum_{i=1}^{n} \text{task}_i \cdot \text{length}_i}{\text{evaluation of the capacity of VM}_j}$$

In this case, the fitness of the process numbers are shown by fit within the capacity of VM within the tasks of descriptions task length. To obtain the overall capacity, equation (4) is applicable [1].

$$\text{capacity}_j = \text{pe}_j \cdot \text{num}_j \cdot \text{pe}_j \cdot \text{mips}_j + \text{vm}_j \cdot \text{bw}_j$$

In this case, the number of processors in the system is indicated by pe_numj within the instruction limits shown by pe_mipsj in the required bandwidth of vm_bwj as per the ability of the virtual machines. The identification and the allocation of these elements lead to the fitness selection of the required neighbourhood VMs. When the process recruitment reaches, the bounding element is definable by the formula (5) [1].

$$\text{fit}_j = \frac{\sum_{i=1}^{n} \text{task}_i \cdot \text{length}_i + \text{inputfile}_i \cdot \text{length}_i}{\text{evaluation of the capacity of VM}_j}$$

In this case, the input File length becomes the size of the task with pending execution details. Once this is done, the overshadowing fitness is chosen in line with the assign task VM. This allows the calculation of load balance within the workload VMs as per the information obtained from the datacentre [6]. It becomes easier to get the standard deviation with respect to the actual load VMs as formula (6) [1].

$$\text{S.D.} = \sqrt{\frac{1}{n} \sum_{j=0}^{n} (X_j - \bar{X})^2}$$

Any VM now has a processing time indicated by the equation (7) [1].

$$X_j = \frac{\sum_{i=1}^{k} \text{task}_i \cdot \text{length}_i}{\text{capacity}_j}$$

The mean processing time for all the VMs is also indicated as formula (8) [1].

$$X = \frac{\sum_{j=1}^{n} X_j}{n}$$

From the above conclusive equations, it can be stated that whenever S.D. of the full VM becomes less or equal to mean, a balanced state of the system is experienced. In case S.D. is higher than the mean, the imbalance state prevails [7].
2.3. Results and discussion

By the use of a CloudSim tool (any version from 3.0.1); the simulation platform is subject to the table 1 [1].

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datacenter</td>
<td>Number of Datacenter</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Number of Host</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Type of Manager</td>
<td>Space_shared,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time_shared</td>
</tr>
<tr>
<td></td>
<td>Number of PE per Host</td>
<td>2 - 4</td>
</tr>
<tr>
<td></td>
<td>Bandwidth</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td>Host Memory (MB)</td>
<td>2048 - 10240</td>
</tr>
<tr>
<td></td>
<td>Datacenter Cost (The cost of using processing in this resource)</td>
<td>10</td>
</tr>
<tr>
<td>Virtual Machine (VM)</td>
<td>Total number of VMs</td>
<td>30 - 210</td>
</tr>
<tr>
<td></td>
<td>MIPS of PE</td>
<td>1000 - 2000</td>
</tr>
<tr>
<td></td>
<td>Number of PE per VM</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>VM Memory (MB)</td>
<td>512 - 2048</td>
</tr>
<tr>
<td></td>
<td>Bandwidth (Bit)</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>Type of Manager</td>
<td>Time_shared</td>
</tr>
<tr>
<td>Task</td>
<td>Total number of Tasks</td>
<td>100 - 700</td>
</tr>
<tr>
<td></td>
<td>Length of Task (MiB)</td>
<td>5000 - 20000</td>
</tr>
<tr>
<td></td>
<td>Number of PE per requirement</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Type of Manager</td>
<td>Space_shared</td>
</tr>
</tbody>
</table>

For setting the ABC algorithm, the parameters are shown in TABLE 2 [1].

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Number of scout bees</td>
<td>1000</td>
</tr>
<tr>
<td>m</td>
<td>Number of sites selected out of n visited site</td>
<td>5</td>
</tr>
<tr>
<td>e</td>
<td>Number of best site out of m select site</td>
<td>1</td>
</tr>
<tr>
<td>np</td>
<td>Number of bees recruited for best e site</td>
<td>800</td>
</tr>
<tr>
<td>nsp</td>
<td>Number of bees recruited for other (m-e) selected sites</td>
<td>200</td>
</tr>
</tbody>
</table>
As possible to establish, the assessment of the procedure is describable under three algorithmic instructions. There is the presence of First Come First Served (FCFS) that works on the chronological reach of the given task. There is the Shortest Job First (SJF) that considers the sequencing by making the selection of the first job in short terms stated [8]. The Longest Job First (LJF) ensures that the heavy job is selected first. The results of the first and the second experiments are shown in Figures 3 and 4[1, 5].

![Figure 3. Comparison of makespan and the tasks](image)

![Figure 4. Comparison of makespan and the number of VMs](image)
As shown in figure 3, there are a number of increasing requests. As a result, the average numbers of requests being made are progressing under the same design specifications [9, 10, 11, and 12]. Consequently, ABC_LJF can be said to be effective within the required scheduling on the full system [12, 13]. As for the second experiment, there is an exponential decay of makespan with the number of VMs. In the same design show, the previous conclusion still prevails.

3. CONCLUSION

As the paper indicates, ABC controlled algorithm is seen to be suitable for solutions surrounding VM scheduling management so that the tasks are scheduled in the right manner and needs. This goes under the adjustable features in Cloud Computing. As these parameters are optimized, the stability of the system still prevails. This allows the system to operate with optimum utility. It is then imperative to state that ABC algorithm is good for Cloud Computing and its environment under the utility of the required parameters in a manner that the tasks are handled with utmost utility. This stability makes the system to works without crashing in the manners of ABC_LJF needs and designs.

REFERENCES