FNC a New Method to Optimize Task Scheduling Based on ACO in Cloud Computing

Hamid Mehdi

Department of Computer Engineering, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran

Hamidmehdi@Gmail.com

Abstract

Nowadays Cloud computing has to provide necessary facilities in order to provide reliable and truthful service to users. Based on mentioned point task scheduling is one of the fundamental issues in cloud computing environment. In fact one of the main features of task schedule is its scheduling strategy to change environment and the types of tasks which leads to load balancing. Many different resources exist in cloud computing including CPU, RAM, and Bandwidth which are engaged by virtual machines. Since the number of users require different types of virtual machines and all of resources are associated with virtual machines, task scheduling algorithms help to load balancing more efficient. In this paper, it has been tried to present a new method based on bandwidth manager via input as well as output data finally calculating load factor by formula for every Virtual machines. Results show the presented method is performed and good for task scheduling in cloud computing environments.

Keywords: Cloud Computing, Job Scheduling, Scheduling Algorithm, Cloudsim, Cloud Analyst.

I. Introduction

According to this matter which a cloud provider has to serve many users in Cloud computing system (Jinhua et al,2010), Therefore, Task scheduling is the one of matters which plays a vital role in establishing cloud computing systems. The important aim of job scheduling is to extend the resource utilization and minimize processing time of the tasks (Kun et al,2011). Ant colony optimization (ACO) has proposed as a method of optimization technique as practical. ACO takes from the searching for food behavior of real ants (Dorigo et al,2006). Job Scheduling is an important point in Cloud computing, the reason why its feature. The cloud provides a verity platform by creating a VM that assists users in completion their jobs within a logical time and cost effectively without sacrificing the QoS (Yanget al,2011). The most of task scheduling methods are developed to complete two aims. The first one is to promote the QoS in executing the jobs and provide the expected output on time. The Next one is to assert performance and justice for all jobs. There are many proposed task scheduling algorithms which many of these proposed algorithms are especially for doing jobs in a cloud. ACO algorithm is a random finding algorithm. It takes the behavior of real ant colonies in nature to find the nutrients and connect to each other by pheromone set down on path. Many researchers used ACO to solve optimization problem (Kun et al,2011). Load balancing is one of the main cloud mechanisms which distribute the dynamic workload among nodes. It achieves the high user pleasure and resource utilization. The Article proposed the “ACO optimization for job scheduling in cloud computing”. The head promote of task scheduling is to achieve a high performance computing and the best system throughput. Scheduling manages availability of CPU memory and the best scheduling policy gives the highest utilization of resource. Nowadays, when it comes to cloud computing a set of facilities comes to mind including data center, virtual machine. Any data centers have many servers as well as any servers have many VMs. Cloud has three various functional layers including, infrastructure as a service, platform as a service, and software as services. Cloud
Computing is still under its development stage and has many issues and challenges out of the different issues in cloud task scheduling plays a vital role in determining the effective execution. Scheduling refers to the set of policies to control the order of work to be performed by a system. There have been different sorts of scheduling methods existing in distributed computing systems, and task scheduling is one of them.

II. Related works

Many various methods exist associated with the ACO algorithm. One of the main purposes of using the ACO method is load balancing which has good efficacy on datacenters. As the first algorithm, load balancing ACO (Kun et al., 2011) was proposed which is used to find the optimal resource allocation for each task by dynamically. This algorithm compared with Basic ACO algorithm and FCFS algorithm, result nodes are balanced dynamically. In addition to this, tasks are mutually independent i.e., there is no superiority restriction between task and computationally intensive which is not realistic for cloud. Main disadvantage of this method can be the heterogeneity of system. The problem of this kind of system is removed in a method using Load Balancing mechanism based on ACO and complex network theory. As, the second algorithm states for three level cloud computing network (Wang et al., 2010), works efficiently and achieve better utilization of resources but, gives more overhead during run time. (Zhang et al., 2010). It provides the suitable scalability of nodes and better fault tolerant system. The overhead increases during run time and poor response time. An Ant encounters dead state at the end due to absence of synchronization of Ants. In Cloud initiative using modified ACO framework (Yanget al., 2011) algorithm it minimizes the make span. In his method modification is involved in basic pheromone updating formula. This modification gives the better utilization of resource but does not give the good fault tolerant system. Load balancing methods have primary goal the better utilization of resources. Load balancing of nodes using ACO optimization (Rastogi et al., 2012). It is the best case for effective load balancing. Because in previous work can move in just one direction (Zhang et al., 2010). But in this algorithm an ant can move both direction that is forward direction and backward direction. Such as an ant finds the food is called forward direction and return to the nest is called backward direction. This is beneficial for balanced the node quickly. This algorithm works fast because an ant can move concurrently in both directions for balanced the node. This algorithm gives the better utilization of resources. But this algorithm (Rastogi et al., 2012) works on limited parameter such as speed, network overhead and fault tolerance. This algorithm is more power consuming and it is not consider the energy related issues. Several operational cost does not consider which may result in poor performance.

III. ACO Algorithm

Dorigo M. introduced the ant algorithm based on the behavior of real ants in 1996 (Dorigo et al., 2006) (Kun et al., 2011), it is a new heuristic algorithm for the solution of combinatorial optimization problems. Investigations show that: Ant has the ability of finding an optimal path from nest to food. On the way of ants moving, they lay some pheromone on the ground; while an isolated ant encounters a previously laid trail, this ant can detect it and decide with high probability to follow it. Hence, the trail is reinforced with its own pheromone. The probability of ant chooses a way is proportion to the concentration of a way’s pheromone. To a way, the more ants choose, the way has denser pheromone, and the denser pheromone attracts more ants. Through this positive feedback mechanism, ant can find an optimal way finally (Dorigo et al., 2006). ACO is inspired from the ant colonies that work together in foraging behavior. In fact
the real ants have inspired many researchers for their work, and the ants approach has been used by many researchers for problem solving in various areas. This approach is called on the name of its inspiration ACO. The ants work together in search of new sources of food and simultaneously use the existing food sources to shift the food back to the nest (Rastogi et al, 2012).

Figure 1. System Structure - Forward movement

IV. Ant Colony Optimization Algorithm

Based on prior works Ant begins from the head node. All of ants cross the width and length of the network in such a way that they know whole location of under loaded node and overloaded node. Network has been traversed and updated the pheromone table by ants which the information of utilization of each node is going to be stored. In the prior work movement of ant two ways:

1. Forward movement-The ants continuously move in the forward direction in the cloud encountering overloaded node or under loaded node

2. Backward movement-If an ant encounters an overloaded node in its movement when it has previously encountered an under loaded node then it will go backward to the under loaded node to check if the node is still under loaded or not and if it finds it still under loaded then it will redistribute the work to the under loaded node. The vice-versa is also feasible and possible. (Rastogi et al, 2012)

The ant use two types of pheromone for its movement these are:

A. Foraging Pheromone

Generally ACO uses foraging pheromones to explore new food sources. In our algorithm the ant would lay down foraging pheromone after encountering under loaded nodes to search overloaded nodes. Therefore, after an ant comes up to an under loaded node it will try to find the next path through pheromone.

B. Trailing Pheromone

In a typical ACO the ant uses trailing pheromone to discover its path back to the nest. However, in our algorithm the ants would use this to find its path to the under loaded node after encountering overloaded
Node. Therefore, after an ant encounters an overloaded node it will try to trace back. The main aim of the two types of pheromone updating is to classify the ants according to the types of nodes they are currently searching for. The ants after originating from the head node, by default follow the Foraging pheromone, and in the process, they update the FP trails. After coming upon an overloaded node they follow the Trailing Pheromones and simultaneously update the TP trails of the path. After reaching an under loaded node of the same type they update the data structure so as to move a particular amount of data from the overloaded node to under loaded node. Ants then select a random neighbor of this node, and if they encounter an under loaded node they start following the FP to trace an overloaded node, therefore they repeat the same set of tasks repeatedly in a network to improve the network performance. (Rastogi et al., 2012)

![Figure 2. System Structure - Backward movement](image)

V. Proposed Algorithm

The characteristics of Ant algorithm has been utilized in the Article. In the other side other SALB algorithm inherits the basic idea of ACO. It considers the loading of different nodes. This method called FNC (Factor Node Calculation).

A. Task Scheduling Algorithm

In presented method two different movement exist including Forward and backward direction. Figures 1 and 2 show the proposed algorithm.

- Forward direction - In this movement ant finds overloaded node. Figure 1
- Backward direction - In this movement the ant replaced the VM with other. Figure 2

Proposed Algorithm includes of three steps which are as follows:
- Find Overloaded node in Datacenter (DC).
- Finding overloaded node by Formula 1
- replaced the VM with other

The main parameter in proposed method calculation is based on usage bandwidth. Before anything it must be calculated as follow formula:

\[
\text{Factor Node (FN)} = \frac{P}{B} \quad (1)
\]

\(P = \) Usage BW of Host – This is input bandwidth of any hosts  
\(B = \) Total Input BW - This is input bandwidth of Datacenter

This formula assists to find factor node in any hosts according to requests to host. Therefore this formula will reach to a number which show traffic amount. While input traffic to hosts becomes more, the factor node decreases.
The first step, both bandwidth of any hosts must be calculated by VMs which become $P$ and input bandwidth of datacenter which is specific besides becomes $B$. After calculating $P$ plus $B$ must be divided which leads to find factor node. This trend, ant finds the overloaded node for doing this ant moves the forward direction. Initially ant assign the threshold value to all the node and then ant calculate the BW utilization of each node, forward ant also maintain the pheromone table. Pheromone table contains the threshold of each node and BW utilization of each node. If the BW utilization is greater than the threshold then node is overloaded. Function I return the overloaded Node id to the function II. Then, selection of VM is based on minimum migration time. The migration time is calculated as the amount of user request.

In this module, Ant move in backward direction, Ant select the particular VM. In the third step Ant determine the under loaded host and replace VM.

VI. Simulation Result and Analysis

This paper simulate proposed algorithm based on Requirements parameters as follow: around 10 datacenters 500 hosts, type of manager time shared VMs 40, MIPS 2500, RAM 4096 MB, bandwidth 1 Gb/s, Cloudlet around 100. In this experiment Figure1 shows that requests are go to host 1 and the simulation time when the first host utilization is greater than the threshold then node is overloaded. Figure 2 shows that simulation time when host 0 and host 1 are balanced.

<table>
<thead>
<tr>
<th>Table 1. Pseudo Code Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
</tr>
<tr>
<td>Begin</td>
</tr>
<tr>
<td>for Node 1 To Node N</td>
</tr>
<tr>
<td>enter nodeid</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>then</td>
</tr>
<tr>
<td>select nodeid</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

| **Step 2** | SelOverloaded(Number of VM,nodeid) |
| Begin | Select overload nodeid |
| for vm1 to m | calculate migration time |
| end | select vm=minimum migration time |
| end |

| **Step 3** | ChangeVM (vmid, underload node id) |
| Begin | Select underload node from pheromone table(unid) |
| redirect the cloudlet of vmid to the unid | if (node is balanced) |
| then | update pheromone table |
| end | end |
VII. Conclusion

It has been presented a modification and promotion ACO method of ant colony which around cloud computing environment. The main aim of the proposed algorithm is efficient in finding the overloaded node and load balancing in lowest time. The proposed method associated with concept of ACO promotion in connection with movement of the ant which is in both forward and backward direction. The way ants are created pheromone table that contains the data around all nodes and its corresponding load. The aim is to balance the node with efficiency & maximum utilization of resource and a better performance is the
need of our algorithm. It improves the performance by achieving the best result in terms of throughput, response time.


iii. Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang “Cloud Task scheduling based on Load Balancing Ant Colony Optimization” 2011 IEEE.


