

LOUD COMPUTING – PARTITIONING ALGORITHM AND LOAD BALANCING ALGORITHM

Anisaara Nadaph¹ and Prof. Vikas Maral²

¹Department of Computer Engineering, K.J College of Engineering and Management
Research Pune

²Department of Computer Engineering, K.J College of Engineering and Management
Research Pune

ABSTRACT

Tremendous usage of internet has made huge data on the network, without compromising on the performance of network the end-users must obtain best service. As cloud provides different services on leasing basis many companies are migrating from their own Infrastructure to cloud, This migration should not compromise on performance of the cloud, The performance of the cloud can be improved by having excellent load balancing strategy such that the end user is satisfied. The paper reveals the method by which a cloud can be partitioned and a study of different algorithm with comparative study to balance the dynamic load. The comparative study between Ant Colony and Honey Bee algorithm gives the result which algorithm is optimal in normal load condition also the simplest round robin algorithm is applied when the partition are in Idle state

KEYWORDS

cloud , central controller system (ccs), partition status collector

1. INTRODUCTION

Due to versatile use of internet cloud computing is becoming the back bone of soft computing. When a server is overloaded the arriving job should be diverted to the server which is in normal (underloaded) state such that there will be maximum utilization of the available resources. Cloud computing has given the IT sector new direction for utilization of resources in a organized manner as a user pay for usage, Cloud computing is a combined technique from the Grid Computing , utility computing and autonomic computing.

Cloud Architecture can be alienated into 2 section

- i. Front end – Client computer or application to connect the back end
- ii. Bach End – servers, data center or data storage unit

The two are connected by the network called as Internet, There is a central manger to monitor the traffic for efficient performance of the system. The architecture for balancing load depends on whether the system is for static or dynamic as the static system doesn't store the current status of the system, it is immaterial for its design in disparity to this the dynamic system accumulate the current system information and works according to what is the current status of the system.

2. LITERATURE SURVEY

This section gives the study of different strategy to partition the cloud also for balancing the load the different algorithm available

2.1. Partitioning Techniques

There are many Partitioning techniques such as

- i. Relocation Algorithms
- ii. Probabilistic Clustering
- iii. K-medoids Methods
- iv. K-means Methods
- v. Density-Based Algorithms
- vi. Density-Based Connectivity Clustering
- vii. Density Functions Clustering.

All this technique has many drawback due to which it is not efficient for the project. In this paper we will be simply dividing the server based on geographical location in which we will consider the Region wise division of server this will simplify to check the traffic rate of server, as some part of globe have night and other have day due to this the rate of traffic is different for different region. Cloud computing is growing technology in the field of computer science. Gartner's [12], in his report that the cloud will change the scenario in IT sector. The cloud computing is changing life with various types of services. Users get service from a cloud without paying attention to the details [11]. For cloud computing NIST[5] has given a definition as a model for enabling everywhere, suitable, on-demand network access to a common pool of configurable computing assets that can be swiftly provisioned and unconfined with minimal management service provider interaction. Much and more people pay interest to cloud computing[6]. Cloud computing is resourceful and scalable but maintaining the steadiness of processing numerous jobs for cloud computing surroundings is a very multifaceted problem with load balancing receiving much consideration. As we know in dynamic system the job can arrive in a pattern which is not predictable and the abilities of each node in the cloud vary, for load balancing problem, workload control is critical to pick up system performance and maintain steadiness. The load balancing model given in this piece of writing is expected at the public cloud which has abundant nodes for geographic isolated areas. Thus, this model has number of partition in the cloud for a public cloud. These partitions make it easy for load balancing when the location is very large and compound. The cloud has a main manager that prefer the appropriate partitions for incoming jobs while for every job that is arriving the balancer decides the finest load balancing policy from Ant colony and Honey Bee load balancing algorithms[3][2].

2.2. Load Balancing

Load balancing is a new technique that improves the performance of available networks and resources by providing a highest throughput with least response time. Dividing the load between servers, data can be sent and received with negligible delay. A simple example of load balancing in our daily today life can be concerned with websites. Without load balancing, users could experience long delays, timeouts and possible delayed system responses time. Load balancing solutions usually apply redundant servers which help a better distribution of the communication traffic so that the website is availability without delay. There are different load balancing

algorithms, which can be broadly divided into two groups: static balancing [4] and dynamic load balancing.

There are many research work of load balancing for the cloud partitioning. Load balancing in cloud computing was described in a white paper written by Adler[8] who introduced the methods used for load balancing in the cloud. Load balancing in the cloud is further a new problem that requires new architectures to adapt to many enhancement. Chaczko et al described the idea that load balancing plays in improving the performance and maintaining stability

2.3. Ant Colony Algorithm

This algorithm is based on the natural behavior of the ant in which a special substance called pheromone is laid down by the ant who goes out in search of the food and remaining ant follows the pheromone [13][10], The shortest path is selected in which maximum pheromone is there as that will be the shortest path [13][10].as the ant select the shortest path the ant colony[4] algorithm each ant will maintain a record-set and the traffic that is arriving will be diverted to the path which has the more probability as in natural ant which select a path with highest pheromone.

2.4. Honey Bee Method

A honey-bee colony consists of[9] queen, forager. Forager are of type Employed(bees with job) Unemployed(bees without job) where unemployed bee are also called as the scout bee who moves around in search of food without any knowledge and employed bees are those to whom the scout bee gives the information and based on the dance performed the patch or group of unemployed bee moves scout bee moves in all direction in search of food. And they come back and perform the waggle dance[7] on the dance floor to give the information to the remaining bee regarding the distance, direction and extent of food that is available based on the dance floor and accordingly patches(group of honey bee) moves to collect the food.

3. PROPOSED WORK

3.1. System Architecture

There is a central controller system(ccs) to which all the request arrive and then the Load balance is selected based on the balancer having minimum load, after that the nodes on which the data to be processed is selected This selection is based on Ant Colony Algorithm or Honey Bee Algorithm.

3.2. System Model

There several cloud computing methodologies in this paper we are focusing on the public cloud, and this cloud (public cloud) is based on the some standard cloud computing architecture, with service provider providing the service. A large public cloud consists of many nodes and the nodes in distinct geographical areas. To manage large cloud spread over large geographical areas cloud partitioning are used. A public cloud that consist of partition with divisions based on the geographic locations..The load balancing algorithm is based on the cloud partitioning architecture. After establishing the cloud regions, the load balancing is invoked: when a job comes at the system, with the central controller system make decision that to which cloud region the job enters the decision is based on maximum memory available if same amount of memory is available then on number of request. The partition monitoring controller load balancer then make

decision on strategy to allocate the jobs to the nodes in a particular region. According to the monitoring controller when the load status of a cloud region is normal, this partitioning can be managed local way. But if the status of cloud partition load is not normal, job should be relocating to another region

3.3. System Flow

This flow diagram explained below provides the systematic description of the entire system with Ant colony Algorithm and Honey Bee Behavior. Flow as shown in figure , in which first a request arrives to the central controller system and then depending on the free memory of the load balancer it is decided as per which balancer is selected and then the node for processing is selected by Ant colony or Honey bee algorithm, for different size job instances of request the study for both algorithm is done and comparative result will be shown.

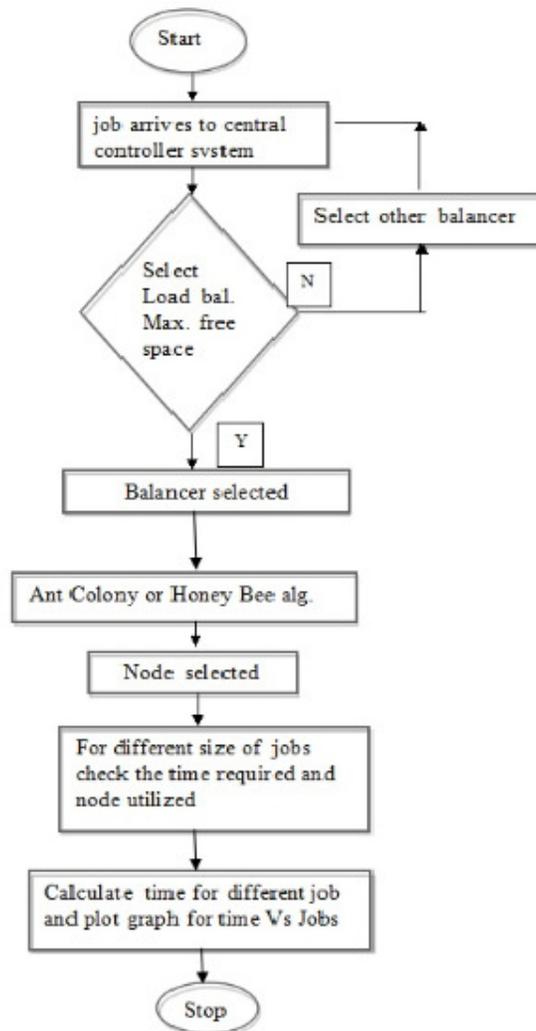


Figure1 Flow Diagram for Load Balancing using Ant Colony Algorithm or Honey Bee Algorithm

4. MATHEMATICAL MODEL

This fragment of paper describes the mathematical model for the entire system which includes selection of Load balancer by the central controller system and then selection of Nodes by the balancer for processing by Ant Colony Algorithm and Honey Bee Behaviour

4.1 Selection of balancer

The function used for selection of the load balancer[1]

$$F = m_utli(B) = \sum_{n=0}^n Memory(n) \quad (1)$$

Where Memory(n) is memory utilized by node, m_utli(B) is the memory utilized by Load Balancer.

- a. If m_utli(B) = 0 then the Balancer is in Idle state.
- b. If m_utli(B) > 0 and <= Threshold Load, Balancer is in Normal state.
- c. If m_utli(B) > threshold, Balancer is in Overloaded state.

4.2 Selection of node in Idle state

The selection of node is done in Round Robin fashion as shown in equation(2)

$$\forall \text{ node}(i) : \text{empty}(\text{node}(i)) \rightarrow \text{select}(\text{node}(i)) \quad (2)$$

where i takes the value from a to d for balancer A and a to d for balancer B where a to d are the nodes attached with the balancer.

4.3. Selection of node in Normal state

4.3.1 Load Balancing by Ant Colony Algorithm[4]

Let the system be defined by S for choice of node using Ant Colony Algorithm on Cloud Partition. The 5 parameters for the system S selection is given as Input, Output, Function, Success, Failure, hence it is defined as

$$S = \{I, O, F, S, Fi\}$$

where I= input to the system, O= output of the system, F = function used in the system, S= condition for success, Fi= condition for failure of the system.

For the system I = set of jobs that arrived from the central controller, the output obtained for the system will be (O)= selection of the node, the system is in success state if the node is obtained and the system is in failure state if node is not obtained for storage of job. selection of Node using Ant Colony Algorithm the function F2 [10] is computed as

- a. Initially a randomized selection is done is memory space is available then select the node

- b. If the randomly selected node does not have sufficient space then the node having maximum entry in the forward Pheromone table is selected as shown in equation (3) fp is the table having forward pheromone or value for selection of the next node and max is a variable used to check for maximum pheromone and it is initialized to 0

$$\sum_{i=1}^n fp(i) > max \left\{ \begin{array}{l} Max= fp(i) \\ Next\ node=i \end{array} \right. \quad (3)$$

4.3.2 Load Balancing by Honey Bee Behavior[2]

Let S1 be System for choice of node using Honey Bee Behavior .The system S1 has 5 parameters in it Input, Output, Function, Success, Failure given as S1= {I, O, F, S, E}.

The input to the Balancer for selection of node by honey bee is job of varying size, Oupput will be node selected. The function applied for node selection is in equation(3)

$$\forall \text{ node}(i) : M_Free_{max}(i) \rightarrow \text{select}(\text{node}(i)) \quad (4)$$

where i takes the value from a to d for balancer A and a to d for balancer B where a to d are the nodes attached with the balancer.

- Preemptive scheduling is needed when the load balancer is selected, but the server nodes do not have sufficient space for execution of job then the jobs with same priority are checked and the node which has maximum number of same priority job are selected.

$$T_n = \sum_{i=1}^m P_i \quad (5)$$

In equation (5) used T_n is the total number of job of same priority, the T_n value is calculated for each server for different priority which ranges from 1 to m, and the node having more number of same priority job as arrived job is assigned for processing.

5. EXPERIMENTAL RESULT

The overall memory of each balancer is calculated my adding the memory space of each server. Initally all server are in Idle state and hence a Round Robin algorithm are used whereas when all server reaches the normal state the Ant Colony optimization or Honey Bee behaviour algorithm are used. Job is assigned to the central Controller (ccs) then the ccs decides (based on the balancer with maximum space). the job is transferred to the balancer for this either the Ant Colony or Honey Bee algorithm is used and for each job with Ant Colony and Honey Bee the time required to assign the node is calculated for the same the result obtained are as follows. The figure 2 and figure 3 gives the experimental result when the maximum threshold of the server is set as 100KB whereas figure 4 and figure 5 gives the result when maximum threshold is set to 300KB.

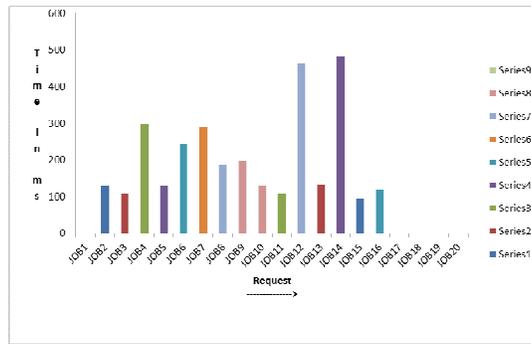


Figure 2 – Ant colony optimization (upper threshold -100)

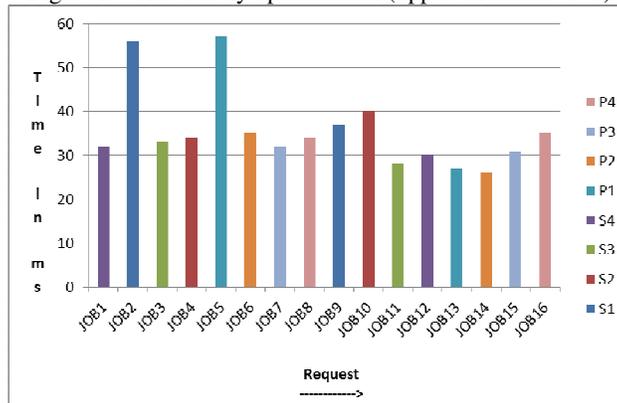


Figure 3 – Honey Bee behavior (upper threshold -100)

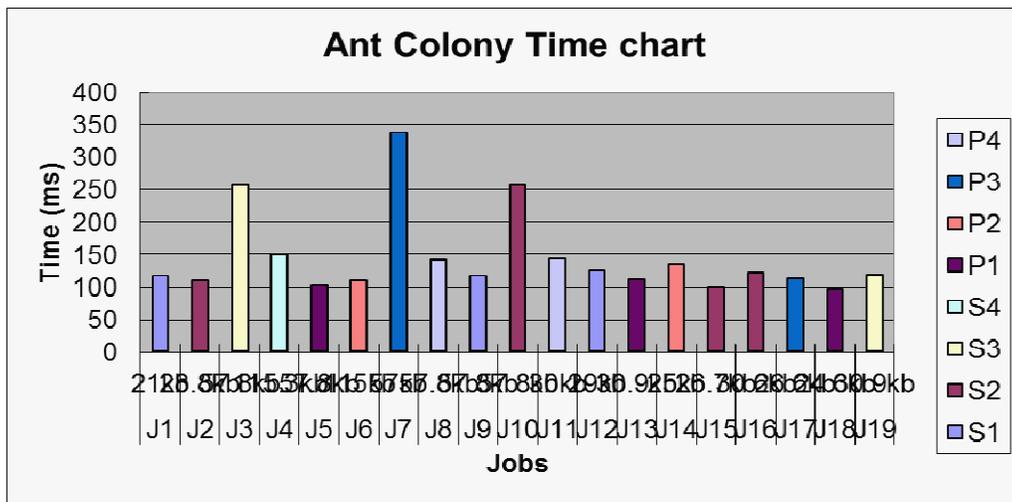


Figure 4 – Ant colony optimization (upper threshold -300)

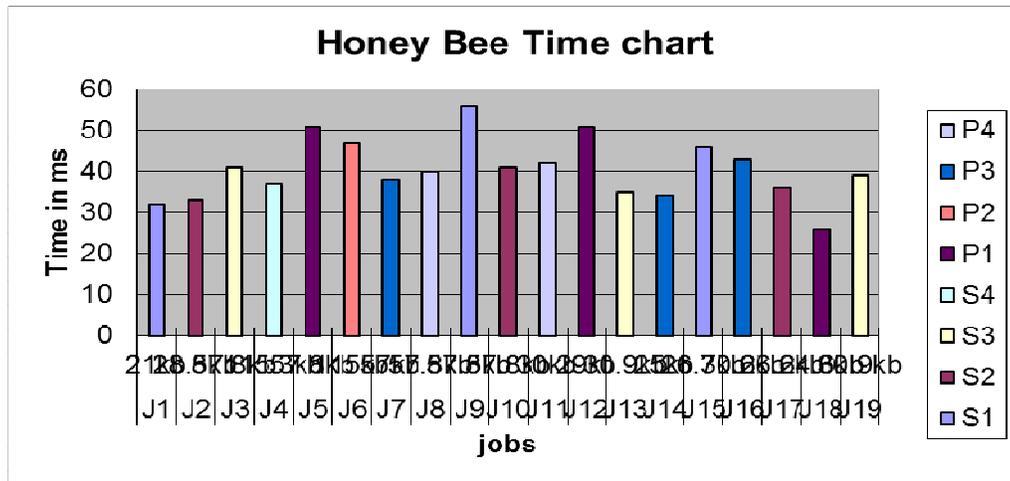


Figure 5 – Honey Bee behavior (upper threshold -300)

6. CONCLUSIONS

This paper presented a analysis of Honey Bee and ant Colony load-balancing algorithms for Cloud partitioning. It was observed that centralized allocation will not efficient for load across all nodes in a system. So a partitioning approach is required that balances load among network. As such, Honey Bee and Ant colony algorithms were consider for this comparative analysis of study. Both (Honey bee foraging and Ant behavior) based on a swarm intelligence. When resource and system architecture are considered, new issues will arise. Both Ant Colony and Honey Bee appear better as the number of processing requests is increased. As such, enhanced future work required for further study, It is found that there are number of algorithms available for load balancing. It is not known how to select appropriate balancing techniques for given applications that will provide a suitable configuration for the application. The experiment gives the conclusion that Honey Bee gives better result than Ant colony optimization techniques

REFERENCES

- [1] GaochaoXu, Junjie Pang, and Xiaodong Fu "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud" IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013
- [2] DhineshBabu L.D. a, P. VenkataKrishnab," Honey bee behavior inspired load balancing of tasks in cloud computing environments", www.elsevier.com/locate/asoc,2013.
- [3] LizheWang,Jie Tao, Marcel Kunze "Scientific Cloud Computing: Early Definition and Experience" The 10th IEEE International
- [4] Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta and KunwarPratap Singh, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization", 2012 14th International Conference on Modelling and Simulation
- [5] P. Mell and T. Grance, The NIST definition of cloud computing, "http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf", 2012.
- [6] Microsoft Academic Research, Cloud computing, "http:// libra.msra.cn/Keyword/6051/cloud-computing?query= cloud%20computing", 2012.
- [7] Nyree LemmensCoMo, VrijeUniversiteit van Brussel, Belgium Steven de Jong, "A bee algorithm for multi-agent systems: Recruitment and navigation combined"
- [8] Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale. com/info center/white-papers/Load-Balancing-in-the-Cloud.pdf", 2012

- [9] Brian R. Johnson James C. Nieh, " Modeling the Adaptive Role of Negative Signaling in Honey Bee Intraspecific Competition", Springerlink.com, 2010
- [10] Ratan Mishral and AnantJaiswal "Ant colony Optimization:"A Solution of Load balancing in Cloud", International Journal of Web Semantic Technology Vol.3, No.2, April 2012
- [11] Rouse, Public cloud, "<http://searchcloudcomputing.techtarget.com/definition/public-cloud>", 2012.
- [12] A. Bhadani and S. Chaudhary,"Performance evaluation of web servers using central load balancing policy over virtual machine on cloud", Proceeding of the Third Annual ACM Bangalore Conference , January 2010
- [13] <http://mute-et.sourceforge.net/howAnts.shtml>