International Innovative Research Journal of Engineering and Technology

ISSN NO: 2456-1983

International Conference on Emerging Innovation in Engineering and Technology

ICEIET-2017

A novel approach in load balancing for dynamic cloud environment using ACO

[1]A. Selvakumar., [2]Dr. G. Gunasekaran., M.E., Ph.D.
Research Scholar[1], Manonmaniam Sundaranar University, Tirunelveli,
Principal[2], Meenakshi College of Engineering,
[1]Selvakumar Mail-id: igaselva@yahoo.com

Abstract:

Cloud Computing is used to distribute and coordinate tasks in order to get optimal resource utilization and avoid overload. This proposal, present a unique method on load balancing using ant colony optimization (ACO), for workload balancing in a cloud platform. Two approaches, max-min rules and forward-backward ant mechanism are familiarized to discover the applicant nodes for load balancing. This method articulates the pheromone initialization and update affording to physical properties under the cloud environment, including pheromone vanishing, inducement, and penalty rules, etc. Combined with task execution estimation by defining the movement of ants in probability by two ways, that is, the forward movement of ant meets the backward movement of ant, or else in the adjacent node, with the goal of tracking searching processes. The proposed work provides load balancing in dynamic with high network performance.

Keywords: Load balancing; ant colony optimization; pheromone

1. Introduction

Cloud computing is progressively being accepted by large industries, as well as small and medium scale trades, for “utility computing”, which process vast potential for the forthcoming of service oriented computing. Virtualization is a major key aiding technology for cloud environments, which makes it potential to run numerous operating systems and applications on the identical hardware at the equivalent time for providing service by a virtual component. Virtualization technology is used for hardware consumption for improving costs for tragedy recovery and it can also accomplish automatic monitoring for all swarms. The main drawback for this cloud utility computing is to assign of tasks for dynamic resources for distributed computing. The problems faced by virtualization are rough allocation of resources, and a high probability of disaster in the overload nodes. In order to overcome the problems the core concept called Load balancing is used for the properties of transparent.

In latest eras, excessive development has been accomplished by load balancing with the help of swarm intelligence algorithms, such as ant colony optimization, particle swarm optimization, artificial bee colony etc. Ant colony optimization projected by Marco Dorigo in 1992, is a class of stochastic optimization algorithms based on the actions of an ant colony. By analyzing the preceding work of ACO, it is found that the ant colony optimization is proper suitable for load balancing applications in cloud computing because of the following statements

(1) The ant colony is able to edge among different nodes to search for the optimal solution in cloud computing infrastructure;

(2) The ACO is a kind of parallel appliance that can be used in distributed computing

(3) It is a self-organizing process created on the local information to make decisions

The main objective of this paper is to establish a load balancing method which uses ACO to balance the tasks among nodes in cloud environment. The targets, including overload or under load nodes, will be quickly recognized to activate the load balancing by two types of ants with their infrastructures by pheromones.

2. Related work

Load balancing:

Load balancing is used to distributing a superior processing load to smaller processing nodes for attractive the overall concert of system. In cloud environment load balancing is necessary for distributing the dynamic l workload evenly between all the nodes.

Load balancing helps in fair allocation of computing resource to achieve a high User satisfaction and proper Resource utilization .High resource processing and Proper load balancing helps in minimizing resource energy consumption. It helps in implementing fail over, scalability, and avoiding bottlenecks.

2.1. Dynamic Load Balancing Strategy in Cloud Computing

Master slave design is a mature engineering process with a solitary master server or employment tracker and a few slave servers, which has been broadly utilized as a part of distributed computing. In master slave design, an occupation is firstly submitted to a master hub by the client. At that point the employment is isolated into a few executable undertakings in the master hub and the created errands are circulated to various slave hubs. From that point onward, the assignments are executed in the slave hubs independently with the direction of the master hub, and the outcomes are come back to the master hub. At long last, the dispersed outcomes are joined together in the master hub and sent to the asking for client. Besides, the master hub is in charge of observing the entire strides and re-executing the fizzled assignments. T

This procedure, the uneven conveyance of undertakings may precipitate that some slave hubs are in light load conditions while others are in substantial load conditions. For this situation, stack adjusting operation should be completed powerfully for the cloud stage with a specific end goal to keep the stage steady and working proficiently. By breaking down the distributed computing stage, there are a few qualities in the same manner as ACO, for example, cloud hubs being practically equivalent to nourishment areas, information stores to homes, and load designation to scavenging action.

2.2 Main procedure of load balancing with ACO

(1) Ant generation. Check the cloud platform sequence, and generate ants if and only if there are existing overload nodes or underload nodes; and

(2) To find target nodes. According to searching process, the ant is searching for the target nodes which meet the conditions of load balancing in its surrounding area. The target node is also called the candidate node for load balancing.

3. Forward movement of ant -Backward movement of ant Mechanism

The ants are divided into two categories: forward movement of ant and backward movement of ant, which is the tool described in but with distinct definition. The forward movement of ant is liable to find the candidate nodes for load balancing in cloud platform and it jumps the searching activity from its made node. The candidate nodes include overload nodes and underload nodes. The backward movement of ant is in care of updating evidence pheromones for the path as that of its corresponding forward movement ant, but in the alterative direction. The backward movement of ant is created at each time when the forward ant identifies a candidate node. The forward movement of ant calculates the probability for each neighbor before it changes and then selects the main one as its next endpoint. To accelerate junction, add a special strategy to the proposed model when the forward movement of ant meets the backward movement of ant in the same node, that is, the moving probability is computed by considering both the information pheromone of the node itself and the information pheromone from all backward movement of ants in this node. Storage units are set for each node and they are used to save the information pheromone carried by backward movement of ants, with one unit for one backward ant. Two types of ants are regarded as meeting each other only when the forward movement of ant arriving at one node before the timer of one backward movement of ant running out in this node. The information pheromone carried by a backward movement of ant will be cleared when the timer getting zero. When there is more than one backward movement of ant in one node, all the influences by these backward movements of ants should be measured when processing the moving probability of forward movement of ant in the node.

4. Max-Min Rules

Two particular standards, named max-min guidelines, to trigger the forward development of ant with the ingenuity of diminishing the ideal opportunity for seeking nodes as beneath. Rule 1: Maximum esteem trigger. A forward development of ant is produced from a slave hub when the heap in this hub is bigger than a specific limit. It means that the hub has been running near or past its most extreme load, which needs to disperse the tasks without moving hubs in order to accomplish ideal asset use.

Rule 2: Minimum esteem trigger. A forward development of insect is produced from a slave hub when the heap in this hub is littler than a specific edge. It shows that the hub is running in the light load state, which can acknowledge a scope of new undertakings, keeping in mind the end goal to share its asset to the over-burden hubs.

5. Process of Load equalization
Based on the on top of ways, the core steps of load equalization square measure delineate as follows.
(1) Figure moving likelihood for all of its neighbors and choose the most important one as its next destination;
(2) Move to a replacement node and choose whether or not it's a candidate node. If yes, generate a backward pismire and initialize this backward pismire. For forward pismire, head to step 1;
(3) The backward pismire goes back to the place to begin of its forward pismire, on the trail of its forward pismire with the other direction. Update the knowledge secretion of every node the backward pismire passes by, and delete the backward pismire once reaching the beginning point;
(4) Calculate the add resources of the candidate nodes, and stop the method if they're ready to meet the demand ofload equalization and
(5)Perform the load equalization operation.
These steps square measure an equivalent for max-min rules except the thanks to calculate the moving likelihood.

6. Dynamic Load equalization modeling with ACO

With relevancy the strategy conferred on top of, we tend to additional analyze the main points of the moving chance with pismire colony improvement during this section. To explore each load allocation potency and network performance, 2 vital problems should be self-addressed. First, secretion low-level formatting ought to be affordable within the cloud computing surroundings, to satisfy the specified QoS. Second, the secretion update must meet the dynamic demand of the employment variability, with the aim of fast the convergence.

6.1Initialization of secretion

In cloud computing, the physical resources allocated to each virtual node are not the same and usually changing dynamically. Due to of this characteristic, we use the physical resources of virtual machines to measure a node's initial pheromone. Five physical resources are involved in pheromone initialization here, that is, CPU (number of cores and MIPS for each core), internal and external storage, I/O interface.

The CPU capability can be calculated by:

$$p_{CPU} = n \times p$$

6.2 Pheromone Update

The goal of pheromone update in our advance is to increase the pheromone values for slave nodes related with good conditions and decrease those associated with bad ones. Three factors that impacts the pheromone update are measured in our strategy, namely pheromone evaporation, update by task, and encouragement for successful tasks.

6.3 Pheromone vanishing

The pheromone in the node is decreasing over time due to evaporation. We use the local update approach to modify the pheromone on slave nodes where the pheromone is not zero. The pheromone update by evaporation is defined as:

$$\tau_i(t+1) = (1-\rho)\times\tau_i(t), \quad 0<\rho<1$$

6.4 Inducement and penalty Rules

We outline 2 rules for task execution in a slave, that is, incentive rule and penalization rule. The previous means the secretion is raised during this node if the tasks are performed with success. The latter denotes that the secretion is shriveled during this node if the tasks are done unsuccessfully.

$$\begin{array}{ll} \tau_i(t+1) = (1+\theta)\times\tau_i(t), & \\ if(success) & 0<\theta<1 \\ else & -1<\theta<0 \end{array}$$

6.5 Task Execution Prediction

Task execution prediction is to guage the execution rate for a slave node that reflects the aptitude and performance of virtual resources in cloud computing. Usually speaking, the potency of the cloud platform will be improved once allocating the tasks to the slave nodes with higher performance. It tends to style a prediction model that evaluates the execution rate of a slave node for succeeding timeframe by accumulating the previous records. Through this employment and therefore the employment performed last time, we will predict the speed of a slave node for succeeding timeframe by

$$EV_i^{a_{k+1}}(k+1)=\frac{a_{k+1}}{a_k}((1-\omega)EV_i^{a_k}$$

Conclusion

Cloud computing is seen to transform the major services and their resources are availed on the internet by users. Techniques of load balancing which exist having been examined mostly focusing on enhancing quality services and offering expected results on time. Hence, there is the requirement to establish the technique of load balancing that may enhance the cloud figuring performance together with maximum resource consumption. The suggested technique of a load balancing basing on ACO maximization provides optimal resource exploitation.

References

1. Gao, R., & Wu, J. (2015). Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization. Future Internet, 7(4), 465-483. doi:10.3390/fi7040465

2. Gupta, E., & Deshpande, V. (2014). A Technique Based on Ant Colony Optimization for Load Balancing in Cloud Data Center. 2014 International Conference on Information Technology. doi:10.1109/icit.2014.54

3. Ragmani, A., & Moussaid, K. (2016). A performed load balancing algorithm for public Cloud computing using ant colony optimization. 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech). doi:10.1109/cloudtech.2016.7847703

4. YAO, J., & HE, J. (2013). Load balancing strategy of cloud computing based on adaptive artificial bee colony algorithm. Journal of Computer Applications, 32(9), 2448-2450.doi:10.3724/sp.j.1087.2012.02448

5. Zhao, G., Dong, Y., & Wang, D. (2011). Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization. 2011 Sixth Annual Chinagrid Conference. doi:10.1109/chinagrid.2011.17