International Journal of Pure and Applied Mathematics
Volume 117 No. 22 2017, 1-5
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

RESOURCE SCHEDULING FOR LOAD BALANCING USING ANT COLONY OPTIMIZATION

V. Suresh Kumar
Professor, Dept of CSE, Vel Tech Multitech,
Chennai, India
sureshkumar@veltechmultitech.org

Abstract - Load Balance is among the basic important problems while dealing with cloud resources scheduling. Due to the various requirements of the numerous users, the amount of processing varies. The cloud resources scheduling process will have the rapid change in load which may be a reason for resources scheduling tilt. In this research study, a soft computing based Scheduling for load balancing has been projected using Ant colony optimization for allocation and for scheduling the load balance of incoming jobs to the servers or virtual machines (VMs). Qualitative and quantitative analysis has been done on the performance of the algorithm in cloud.
Keywords: scheduling, load balancing, cloud resource, Ant Colony optimization, virtual machines

1.Introduction

Cloud Computing has become the most popular research field adopted by both academicians and industry researchers. The major focus of Cloud technology is to distribute computing resources and services online over the network. In Cloud, store and computing services are purchased on demand by the user. Cloud resources are shared and reallocated during run time. The cloud technology handles itself the knowledge about the configuration of service delivering system and resource management and so the end user need not concern about this aspect. A number of distributed host machines are grouped in a cloud. Cloud computing system constitutes virtual machines using several servers, database centers and memory devices etc., as resources; interconnected in a dependable approach. Whenever there is any demand from any user of the cloud, then cloud system creates a virtual machine inside any one of the host machine to fulfill the clients demand in the form of resources on 'pay per use' criteria.

Virtual machines are created randomly on client's demand and hence each host machine will have variable load. Because of the change in load, few host machines will start the process with overload and some remain with less load. This overload or less load may happen in the CPU, memory, storage or sometimes network related too. In order to ensure efficient as well as effective usage of the above-mentioned resources, load balancing has to be done among running cloud services. A soft computing technique, Particle Swarm Optimization (PSO) is utilized for proper load balancing in cloud computing environment [1].

In spite of many advantages earned through cloud computing, few bottlenecks are also there which restrict the proliferation of cloud computing. Heavy Load on cloud is one of these bigger obstruction and load balancing becomes the major issue for the cloud computing. For efficient load balancing, parameters such as throughput, fault tolerance, response time, performance, scalability and resource utilization, have to be evaluated. By having a complete analysis of these parameters, the load balancing techniques ensure better resource distribution for the user demands. Measurement parameters allow us to see whether the given technique is good enough to balance the load of the traffic on the server or not [2]. Load balancing in cloud computing differs from parallel computing. Cloud computing is based on standard load balancing and parallel computing on classical load-balancing. In cloud computing the architecture and implementation of the load balancing process is different according to the use of commodity servers to perform the load balancing, which provides for new opportunities and economies of scale.

The process of reallocating the total load to the individual nodes of the collective system to improve both resource utilization and job response time is called Load Balancing. It also tries to avoid a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all nodes in the system approximately equal amount of work at any instance of time [10]. The objective of load balance is to achieve optimal resource utilization, maximize throughput, minimum response time, and avoid overload. The heterogeneous environment is considered as a major concern [11] because the heterogeneous environment consists of heterogeneous resource, so the behavior of heterogeneous cloud will be different and has different attributes and different response times for any process.

2. Literature Survey

In 2014, Stuti-Dave et-al [3] presented a "Round Robin" (RR) for load balancing at virtualized environment. In their research, they have suggested an improved Fair RR algorithm approach which provides dynamic time quantum strategy. When the request-in enters ready queue, they are processed and calculated according to time quantum and burst time computation while VM's are allocated. Thus, FRR algorithm provide fairness to larger and smaller incoming requests at executing load resulting in faster load balancing in cloud.

In 2013, Elina Pacini [4] discussed a cloud VM scheduler based on "Particle swarm optimization" (PSO). In this algorithm all hosts of cloud are regarded as swarm and each host in cloud is particle in the swarm. In each iteration,

searching of host is performed and velocity difference is compared with neighbouring host. If the nearby host has a lower load than the original host, then the VM is moved to the neighbour host with a greater velocity. Additionally, keeping information that the particles move through hosts of their neighbourhood in search of a host with the lower load and reaches up quickly. Therefore, each particle makes a move to one of its neighbours, which has the minimum load among all. If its entire neighbourhood is busier than the host itself, the VM is not moved from the current host. All particles move to the minimum load and eventually at the end particle delivered associated VM to the host with low load among neighbouring host and task end. Since each move that a particle performs, involves moving through the network, to minimize the number of moves: every time a particle moves to a neighbouring host with no allocated VM. The particle allocates its associated VM to it directly without performing further steps. The number of messages sent over the network by a particle to their neighbours hosts to obtain information regarding their availability load is accumulated in the network messages variable.

In 2013, Kousik Dasgupta [5] proposed "Genetic algorithm" (GA) for efficient utilisation of resources and also guarantees QOS. In this, randomly populated processing unit is initialized first and encoded them into binary strings. Then fitness value of each population is evaluated in crossover step followed by mutation where small value is picked as mutation probability and this GA process is repeated till either the fittest chromosome (optimal solution) is found or the termination condition (maximum number of iteration) is exceeded. Researchers compared GA with three commonly used scheduling algorithms First come first serve (FCFS), Round robin (RR), Scholastic hill climbing (SHC). The merit of developed strategy has linear search capability to larger extend and is applicable to complex objective function and can avoid being trapping into local optimal solution. The complexity analysis of any algorithm includes computation time complexity analysis and space complexity analysis. Thus it is robust as compared with other three algorithms.

In 2012, O.M Elzeki [6] suggested improved "Max-Min" algorithm to rise Max- Min efficiency by concurrent execution of task as resources and focuses on selecting task with maximum completion time. The algorithm calculates the expected completion time of the submitted tasks on each resource. Then the expected execution time is assigned to a resource that has the minimum overall completion time. Finally, this scheduled task is removed from meta-tasks and all calculated times are updated and the processing is repeated until all submitted tasks are executed. The algorithm focuses on minimizing the total make span which is the total complete time in large distributed environment. The proposed algorithm encourages mapping schema similar to RASA in such concurrency executing tasks and minimization of total completion time required to finish all tasks. Although time complexity of developed algorithm is same as the previous one O(MN^ 2) and same execution time but produces better make span with more reliable scheduling allows concurrent execution of tasks.

Shridhar G. Damanal et al. introduced a modified throttled algorithm for load balancing in cloud computing [7]. This algorithm concerns with the fact that how incoming jobs are assigned to the available virtual machines effectively and efficiently. Al-Jaroodi et al. proposed a Dual Direction Downloading algorithm from FTP servers (DDFTP) for cloud computing load balancing [8]. DDFTP algorithm divides a file of size m into m/2 partitions. Now each server node can work independently on these two partitions, one in the incremental order while other in a decrement order. Along with load balancing, this algorithm minimizes the extent of network communication needed between the clients and nodes resulting in reduced network overhead. It also works on other parameters such as network load, node load, network speed etc.

Mesbahi et al. (2014) has proposed a new cloud light weight model to balance the cloud load. In this algorithm, CloudSim cloud system simulator is used for the validation of algorithm. This algorithm balances the system load among all processing nodes in a cloud datacenter. Using this algorithm in our simulation, we balanced the cloud so that all its nodes have approximately the same weight in terms of distributing system workload. The main advantage of using algorithm is that it not only balances the cloud load but also gives assurance for the Quality of Services (QoS) for end users. It also reduces the migration time during execution and number of VM (Virtual machine) migration processes [9].

Wen et al. (2015) has proposed VM migration strategy based on Ant Colony Optimization for cloud computing load balancing. In this approach, local migration ants monitor the resource utilization and adapt two different traversing strategies to identify the near optimal mapping between the virtual machines and physical machines. For the experimental evaluation of the load balancing, author has used the CloudSim toolkit package and shows the outperform migration results for the proposed ACO-VMM [12].

3. Research Methodology

In this proposed system, ant colony optimization for scheduling the load balancing is used. Resource Scheduling has become one of the major key functionality in cloud research. There are a number of soft computing techniques available for solving complex problems. This paper basically deals with Ant Colony Optimization techniques to solve various problems in scheduling for load balancing in cloud computing environment.

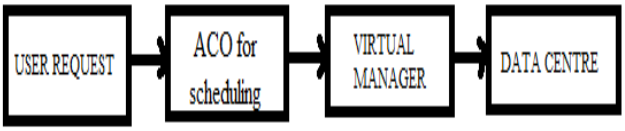

Figure 1 Flow of proposed system

Ant Colony Optimization

Ants basically are simple beings, they jointly form an ant colony which do important tasks including shortest path traversal to find food source and information sharing with other

ants by generating 'pheromone'. In the field of ant colony optimization, models of collective intelligence of ants are transformed into useful optimization techniques which is very useful in computer networking. In this approach, local migration ants monitor the resource utilization and adapt two different traversing strategies to find the near optimal mapping between the virtual machines and physical machines.

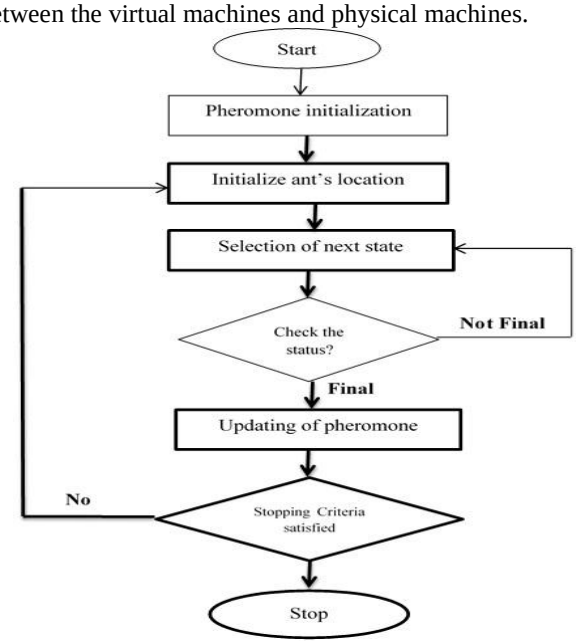

Figure 2 Ant Colony Optimization

The Basic Steps in Ant Colony Optimization are briefed for our understanding. It begins with Pheromone Initialization. The Location of the ant is initialized as an entry state. After this, "Next" state will be selected. If the 'next' state is final state, then update the Pheromone. If the 'next' state is not the final one, repeat the process from initialization. Pheromone updating includes deposit, daemon and evaporate of pheromone.

The following metric table has to be analyzed for load balancing.

Metrics	Description
Nature	determines the **behaviour of algorithms**; either static or dynamic
Overhead	determines the **implementation details of algorithm** like inter-process communication, migration of tasks etc. Also, this should be minimized so that algorithm can work efficiently.
Throughput	is **number of process that is executed per time unit** which should be maximized for better performance.
Process migration	determines **the migration of tasks or resources** from one node to another which should be minimized as it enhances the performance of the system
Response time	is **time to compute or execute any task** which should be minimized for better efficiency.
Resource utilization	The **proportion of the available time** (expressed usually as a percentage) that a system or resources is operating which should be optimized for better performance.
Fault tolerant	The **ability of a system to respond** gracefully to an unexpected hardware or software failure.
Waiting time	is **the amount of time process takes** while in ready queue. It should be minimized for system for better performance.
Scalability	is **the capability of a system network or a process to handle growing amount of work** in a capable manner or its ability to be enlarged to accommodate that growth.
Performance	The **completion of a given task measured against accuracy, completeness cost, speed etc. to check efficiency** of any system.

Table 2 Load Balancing - Metric Table

Advantages of ant colony optimization in Scheduling

It has number of advantages with some critical issues that to be determined in order to enhance reliability of the cloud system. Such problems are associated with the fault acceptance, load balancing, and variety of security related issues in cloud computing system. The main concern of this study is load balancing in cloud computing environment. The load can be memory capacity, CPU load, network load, delay in network, etc.

Performance Analysis

The proposed an efficient algorithm by updating actual Ant Colony Optimization (ACO) algorithm in their own way for scheduling based load balancing of nodes in cloud environment. The Performance of proposed algorithm is shown in table 1 and comparison with other algorithm is shown in the table 2.

Factors	ACO algorithm
Network overhead	Moderate
Replication of files	Full
Resource utilization	High
Implementation complexity	Low
Response time	Fast
Fault tolerance	Yes
Data Center with No. of Virtual Machines	Moderate

Table 2 performance analysis of Proposed algorithm

Factors	PSO	ABC	ACO

	algorithm	algorithm	algorithm
Network overhead	Moderate	High	Moderate
Replication of files	Full	full	Full
Resource utilization	Low	Low	High
Implementation complexity	High	High	Low
Response time	Medium	Fast	Fast
Fault tolerance	No	Yes	Yes
Data Center with No. of Virtual Machines	Moderate	Moderate	Moderate

Table 2 Comparison with other algorithms

4.Conclusion

In the cloud environment the resources scheduling process, if load changes suddenly, this may cause changes the resources in scheduling. This paper represents a soft computing based Scheduling for load balancing has been proposed. Ant colony optimization approach is used soft computing for allocation and scheduling the load that balance the incoming request or load to the servers or virtual machines (VMs). Performance of the Ant colony optimization algorithm in cloud is analyzed by both qualitatively and quantitatively.

Reference

[1] Akhil Goyal, Bharti," *A Study of Load Balancing in Cloud Computing using Soft Computing Techniques*", International Journal of Computer Applications (0975 – 8887) Volume 92 – No.9, April 2014.

[2] Sapna, Pooja Nagpal, "Comparative *Analysis of Soft Computing Based Load Balancing Techniques in Cloud Environment: A Review*" International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 5 Issue 10 Oct. 2016, Page No. 18244-18248.

[3] Stuti Dave, Prashant Mehta "*Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Computing*" IJAC (0975-8887) Volume 94-No.4, May 2014.

[4] Elina Pacini, Cristian Mateos and Carlos Garc´ıa Garino , "*Dynamic Scheduling based on Particle Swarm Optimization for Cloud-based Scientific Experiments*" HPCLatAm VI Latin American Symposium on High Performance Computing, 2013.

[5] Kousik Dasgupta et al., "*A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing*" in Proc. of Elsevier, Procedia Technology, Volume 10, Pages 340-347, 2013.

[6] O.M. Elzeki, M.Z. Reshad, M.A. Elsoud "*Improved Max- Min Algorithm in Cloud Computing*" IJCA (0975-8887) Volume 50-No. 12 July 2012.

[7] Shridhar G. Domanal, G. Ram Mohana Reddy, "*Load Balancing in Cloud Computing Using Modified Throttled Algorithm*," in proc. International Conference on Cloud Computing in Emerging Markets (CCEM), IEEE, pp. 1-7, October 2013.

[8] Al-Jaroodi, J. and N. Mohamad, "*DDFTP: Dual-Direction FTP*," In proc. 11th IEEE/ACM International symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp. 504-513, May 2011.

[9] Mesbahi, Mehran, Amir Masoud Rahmani, and Anthony Theodore Chronopoulos. "*Cloud light weight: A new solution for load balancing in cloud computing*." In Data Science & Engineering (ICDSE), 2014 International Conference on, pp. 44-50. IEEE, 2014.

[10] Kim, Sung-Soo, Ji-Hwan Byeon, Hongbo Liu, Ajith Abraham, and Sean McLoone. "Optimal job scheduling in grid computing using efficient binary artificial bee colony optimization." *soft computing* 17, no. 5 (2013): 867-882.

[11] Nuaimi, Klaithem Al, Nader Mohamed, Mariam Al Nuaimi, and Jameela Al-Jaroodi. "*A survey of load balancing in cloud computing: challenges and algorithms*." In Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on, pp. 137-142. IEEE, 2012.

[12] Wen, Wei-Tao, Chang-Dong Wang, De-Shen Wu, and Ying-Yan Xie. "*An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment*." In Frontier of Computer Science and Technology (FCST), 2015 Ninth International Conference on, pp. 364-369. IEEE, 2015.