International Journal of Computer Science
and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 6, Issue 4, Jun - Jul 2017; 39-48

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

A CONCEPTUAL ANALYSIS ON CLOUD COMPUTING ACO LOAD BALANCING TECHNIQUES

M. PADMAVATHI[1] & SHAIK MAHABOOB BASHA[2]

[1]Research Scholar, Department of CSE, JNTUH, Hyderabad, Telangana, India

[2] Professor, Department of CSE, Al-HABEEB College of Engineering and Technology, Chevella, Hyderabad, Telangana, India

ABSTRACT

The Drastic Change in Computing has happened due to the Cloud Computing Technology. Cloud Computing was built on Utility Computing, Networks, web Services, Virtualization, Distributed Computing and Load Balancing. The cloud Computing service includes many technical issues such as Scalability, Elasticity, Fault Tolerance and High availability. Among many technical issues, Load balancing is an important factor to consider. Many Load Balancing Algorithms developed using various technologies, now in this paper; we focused on Ant Colony Optimization (ACO) techniques. This paper gives a thorough study on Nature inspired Ant Colony Optimization Techniques towards solving Load Balancing Algorithms in Cloud Computing.

KEYWORDS: ACO, Cloud Computing, Elasticity, Load Balancing, Scalability

INTRODUCTION

Cloud Computing and Types of Services

Cloud is a pool of Heterogeneous Resources. It is a mesh of huge infrastructure. Resources are available to the users on-demand basis in pay-as-you-say manner. Cloud Computing is a Service oriented Architecture with the aim to achieve maximum utilization of Resources with higher availability at minimizing cost. Cloud Computing Services and Platforms Diagrammatical representation can be considered as follows.

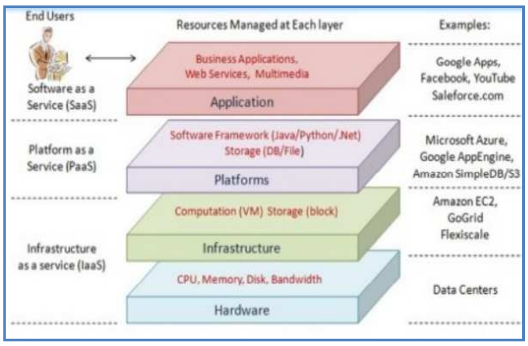

Figure 1: Cloud Computing Services

Load Balancing

In Distributed or Cloud Computing, Load Balancing plays an Essential and vital Role Load balancing techniques used to attain higher utilization of the Resource and user satisfaction by ensuring an efficient allocation of Resources.

Load Balancing; make sure of the uniform work load distribution for all the processes exists in the environment at any time.

The Goals of Load balancing can be identified as follows:

- Resource utilization Ratio.
- Increase Availability of Resources.
- Built Fault Tolerant System.
- Maintain System Stability

Two Approaches in Load Balancing Algorithms exist, namely Static and Dynamic Load Balancing Algorithms. Static Load Balancing Algorithms, Static information is used to select the node with least work Load and the other side of the coin, it is simple to implement. Static Load Balancing Algorithms can be classified as Centralized and Distributed. Static algorithm transfers only fix amount of data [14] [15]. It has no ability to fault tolerance [16].]. Radojevic proposed CLBDM (central load balancing decision model) and this is an advance form of Round Robin algorithm [11].

- Round Robin Load Balancer
- OLB + LBMM
- Max – Min
- Min – Min

Dynamic algorithm is based on different properties of the nodes, such as capabilities and network bandwidth. This need constant check of the node and is usually difficult to implement [10] [11]. In the case of Dynamic Load Balancing Algorithms, current work load of the system is considered. Whereas, Dynamic load Balancing Algorithm implementation is difficult compared to Static Load Balancing Algorithm. In dynamic load balancing, the load is distributed among the nodes throughout run-time. Its advantage comes forward, when any node is failed. Under such circumstances, it does not stop the system, but its performance is affected [13]. In dynamic load balancing, documents and data can be downloaded with the appropriate memory size [12]. To name a few, the following are the instances of Dynamic Load Balancing Algorithms.

- Ant colony algorithm
- Honey bee foraging algorithm
- Throttled load balancing

Work Flow of Load Balancing Algorithm:

- Different Cloud users from different Geographical locations may send request to the cloud let.
- Cloud let is a Networking of small no of Systems and it stores all the user Requests and it is are Connected to the Data Controller (DCC).
- DCC finds out the Virtual Machine (VM) for assigning the jobs, with minimum work load as identified.

- Load Balancer can maintain a table, which stores Virtual Machine Id.

Four Load Balancing Algorithms Designed by using the following four Nature Inspired Intelligent Approaches.

GA'S (Genetic Algorithms): In computer science and operations research, a genetic algorithm (GA) is a Met heuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. By using crossover or mutation, weak generation can be demolished and stronger population proves the Existence [1].

ABC (Artificial Bee Colony): Artificial bee colony (ABC) algorithm is an optimization technique that simulates the foraging behavior of honey bees, and has been successfully applied to various practical problems. ABC belongs to the group of swarm intelligence algorithms and was proposed by Karaboga in 2005. ABC Algorithm consists of 3 types of bees (employed bees, onlooker bees, and scout bees) looking for the best food source which is the solution to the optimization problem. From these 3 types of bees formulas are derived [3].

PSO (Particle Swarm Optimization): In computer science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions, like dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. The PSO Algorithm is widely researched for its simplicity and good performance in reaching solutions. Its work mechanism is to calculate particle movements to new possible solutions by relying hugely on the differential equation of velocity [2].

ACO (Ant Colony Optimization): In the early 1990s, Ant Colony Optimization (ACO) was introduced by M. Dorigo and colleagues. The foraging behavior of ant colonies can be replicated in the simulation and inspires a class of ant algorithms known as ''Ant Colony Optimization'' (ACO). An Ant colony specifies self-organized group of ants that follows very intelligent approach for finding the shortest route between source to the Destination by Stigmergy and pheromone techniques.

ACO is novel nature-inspired Meta heuristic for the solution of hard combinatorial optimization (CO) problems and it belongs to the class of Meta heuristics [4] which are approximate algorithms used to obtain good enough solutions to hard CO problems in a reasonable amount of computation time. This characteristic of real ant colonies is exploited in artificial ant colonies in order to solve CO problems.

Various Parameters to be considered in Load Balancing: The Existing Load Balancing Algorithms have to focus on the following issues [6].

Parameter Name	Description	Value to be
Overhead Associated	It is composed of overhead due to movement of tasks, inter-processor and inter-process communication.	minimized
Throughput	No. of Tasks whose Execution has been Completed.	maximized
Resource Utilization	Utilization of Resources.	maximized
Scalability	Ability of an Algorithm to perform load balancing for a System with any finite Number of Nodes.	Improved
Response Time	Amount of time taken to Respond by a particular Load Balancing Algorithm in a Distributed System.	minimized
Fault Tolerance	An algorithm to perform uniform load balancing in spite of arbitrary node or link failure	maximized
Migration time	Time to Migrate the jobs or resources from one node to other	minimized

Figure 2: Issues to be considered in Load Balancing

ACS (ANT COLONY SYSTEMS) BASIC CONCEPTS

Rather than implementing the exact behavior of Biological ants, Artificial ants are with some omitted features and added features to solve the problems.

Pheromone

Pheromone is chemicals produced by the ants to communicate among individual ants or between individual units and the environment.

Pheromone trails foragers can follow the path to food discovered by other ants. This collective trail-laying and trail-following behavior, whereby, an ant is influenced by a chemical trail left by other ants is the inspiring source of ACO.

The central component of an ACO algorithm is a parameterized probabilistic model, which is called the pheromone model. The pheromone model consists of a vector of model parameters τ called pheromone trail parameters. The pheromone trail parameters $\tau_i \in \tau$, which is usually associated with components of solutions, has values τ_i, called pheromone values.

Pheromone Evaporation

Pheromone effect can be controlled by using Evaporation. Evaporation factor plays very crucial in directing the ants' movements. By using evaporation factor, previous Experience can be reduced. For each iteration, Evaporation can be reduced by the Evaporation factor. The Pheromone values τ_{ij} can be reduced by the factor ρ such that

$\tau_{ij} \longrightarrow \tau_{ij} X$ (1- ρ). As the time increases Pheromone value decreases.

Aging

As ant age increases, the amount of pheromone deposited is decreasing, known as aging. Aging can be used to reduce the previous Experience. In aging, an ant deposits lesser and lesser pheromone as it moves from node to node. Like that ants are less successful in finding the path between source to destination. Aging and Evaporation used to avoid the old non –optimal paths and to discover new optimal paths.

Pheromone Smoothing

The amount of pheromone can be reduced by placing maximum limits (τ_{max}) on every edge and non optimal paths can be reduced. By using Pheromone smoothing approach, pheromone on each edge (i, j) can be Reduced as follows.

$\tau_{i,j}(t') = \tau(t) + \delta \times (\tau_{max} - \tau_{ij}(t))$; δ is a constant between 0 and 1;

Both Pheromone evaporation and Pheromone smoothing are used to prevent the situation of a rising Dominant Paths. Pheromone Smoothing reduces the pheromone concentration by considering optimal path.

Stigmergy

Stigmergy is a process of indirect coordination between the ants towards solving the complex problems like optimization and Control Algorithms by laying a chemical substance, called pheromone. Pheromone–based communication is one of the most effective ways of communication, which is widely observed in nature. This collective trail-laying and trail-following behavior, whereby, an ant is influenced by a chemical trail left by other ants is the inspiring source of ACO.

ACO META HEURISTIC

Heuristic is a technique used to design for solving a problem more quickly, when classic methods are too slow, or for finding an approximate solution, when classic methods fail to find any exact solution. Heuristic techniques can be used in computer science, artificial intelligence, and mathematical optimization. A Heuristic function, also called simply a heuristic.

The objective of a heuristic is to produce a solution in a reasonable time frame that is good enough for solving the problem at hand. This solution may not be the best of all the actual solutions to this problem, or it may simply approximate the exact solution. But it is still valuable because finding it does not require a prohibitively long time.

A meta heuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. The use of meta heuristics has significantly increased the ability of finding very high quality solutions to hard, practically relevant combinatorial optimization problems in a reasonable time.

A particularly successful meta heuristic is inspired by the behavior of real ants.

Starting with Ant System, a number of algorithmic approaches based on the very same ideas were developed and applied with considerable success to a variety of combinatorial optimization problems from academic as well as from real-world applications

Combinatorial Optimization

Combinatorial optimization problems involve finding values for discrete variables such that the optimal solution with respect to a given objective function is found. Many optimization problems of practical and theoretical importance are of combinatorial Nature.Ex: Shortest-Path Problem, Scheduling and Load Balancing etc.

A Combinatorial optimization problem is either maximization or a minimization problem which has associated a set of problem instances. The term problem refers to the general question to be answered, usually having several parameters or variables with unspecified values. More formally, an instance of a combinatorial optimization problem π is a triple (S, f, Ω), where S is the set of candidate solutions, f is the objective function which assigns an objective function value f(s) to each candidate solutions ϵ S, and Ω is a set of constraints. The solutions belonging to the set $\tilde{S} \subseteq S$ of candidate solutions that satisfy the constraints Ω are called feasible solutions. The goal is to find a globally optimal feasible solution S*.For minimization problems this consists in finding a solution $s^* \in S$ with minimum cost, that is, a solution such that $f(s^*) \leq f(s)$ for all $s \in \tilde{S}$; for maximization problems one searches for a solution with maximum objective value, that is, a solution with $f(s^*) \geq f(s)$ for all $s \in \tilde{S}$.

Note that in the following, we focus on minimization problems and that the obvious adaptations have to be made if one considers maximization problems. An important theory that characterizes the difficulty of combinatorial problems is that of NP-***completeness.*** This theory classifies combinatorial problems in two main classes: those that are known to be solvable in polynomial time, and those that are not. The first are said to be tractable, the latter intractable.

We say that a problem is **NP-hard**, if every other problem in NP can be transformed to it by a polynomial time reduction. Therefore, an NP-hard problem is at least as hard as any of the other problems in NP. However, NP-hard problems do not necessarily belong to NP

Two classes of algorithms are available for the solution **of combinatorial optimization problems: Exact and Approximate** algorithms. Exact algorithms are guaranteed to find the optimal solution and to prove its optimality for every finite size instance of a combinatorial optimization problem within an instance-dependent run time. In the case of NP-hard problems, exact algorithms need, in the worst case, exponential time to find the optimum.

Ex: Quadratic Assignment Problem,

In addition to the exponential worst-case complexity, the application of exact algorithms to NP-hard problems in practice also suffers from a strong rise in computation time when the problem size increases, and often their use quickly becomes infeasible.

Approximate algorithms, often also loosely called heuristic methods or simple heuristics, seek to obtain good, that is, near-optimal solutions at relatively low computational cost without being able to guarantee the optimality of solutions.

These can be classified as constructive or local search methods. Constructive algorithms build a solution to a combinatorial optimization problem in an incremental way. Step by step and without backtracking, they add solution components until a complete solution is generated. Although the order in which to add components can be random, typically some kind of heuristic rule is employed.

Local search is a general approach for finding high-quality solutions to hard combinatorial optimization problems in reasonable time. It is based on the iterative exploration of neighborhoods of solutions trying to improve the current solution by local changes. The types of local changes that may be applied to a solution are defined by a neighborhood structure.

The solution found by a local search algorithm may only be guaranteed to be optimal with respect to local changes and, in general, will not be a globally optimal solution.

Def1: A neighborhood structure is a function $N: S \rightarrow 2^{s}$ that assigns a set of neighbors $N(s) \subseteq S$ to every $s \in S$ .N(s) is also called the neighborhood of S.

Def2: A local optimum for a minimization problem (a local minimum) is a solution such that $\forall\, s' \in N(s)$: $f(s) \leq f(s')$. Similarly, a local optimum for a maximization problem (a local maximum) is a solution s such that $\forall\, s^1 \epsilon\, N(s)$: $f(s) \geq f(s')$.

A local search algorithm also requires the definition of a neighborhood examination scheme that determines how the neighborhood is searched and which neighbor solutions are accepted. While the neighborhood can be searched in many different ways, in the great majority of cases the accepted rule is either the *best-improvement rule*, which chooses the neighbor solution, giving the largest improvement of the objective function, or the *first-improvement rule*, which accepts the first improved solution found.

Local search starts from some initial solution and repeatedly tries to improve the Current solution by local changes.

Steps Involved in Local Search

step1: The definition of a neighborhood structure over the set of candidate solutions. Neighborhood structure: it defines for each current solution the set of possible Solutions to which the local search algorithms can move.

Ex: iterative improvement, hill climbing, gradient-descent for maximization and minimization problems, respectively.

Step 2: The local search algorithm searches for an improved solution within the Neighborhood of the current solution.

Step 3: If an improving solution is found, it replaces the current solution and the local search is continued.

These steps are repeated until no improving solution is found in the neighborhood and the algorithm terminates in a local optimum.

A disadvantage of iterative improvement is that the algorithm may stop at very poor-quality local optima.

Comparative Analysis of various Load Balancing Algorithms:

Table 1: Comparative Analysis of ACOLB

Title	Concept	Pros	Cons
Cloud task Scheduling based on Load balancing ACO [5]	1. This Algorithm finds the optimal resource allocation for each task in the dynamic cloud system. 2. It works on minimization of the task completion time that are distributed among virtual machine and Compared with the basic ACO algorithm and FCFS algorithm.	1. Nodes are balanced dynamically 2. Can handle all type of condition.	1. Independent tasks and no precedence constraints between task. 2. It is not heterogeneity of the system.
A Load Balancing A mechanism based on Ant colony & complex network theory in open cloud computing federation[6]	1.It uses small-world & Scale free characteristics of a complex network to support better load balancing 2. This overcomes the heterogeneity.	1. Good Scalability 2. It has excellent fault tolerance 3.Dynamic in Nature	1. Overhead and migration time increases and poor response time 2. at the end ant Encounter dead state. 3. Lack of Synchronization of Ants.

Table 1: Contd.,

Title	Concept	Pros	Cons
Cloud computing initiative using modified ACO framework [7]	1.it minimizes the make Span supports heterogeneous systems 2. It is heuristics algorithm Modified the basic ACO by modifying basic pheromone updation formula.	1.Increased utilization of resources 2.Dynamic in nature	1. Fault tolerance is not good and overhead increases
Load balancing of node in cloud using Ant colony optimization (ACO)[8]	1.The ant updated the single result set Rather than result set. It is compiled only once updating their own duration of movement of ant 2. It is best case that finds the node is under loaded or not. 3. Methodology to collect information by Ant. Forward and backward movement is possible. 4. Two types movement of ant that is forward movement and backward.	1. It gives optimum Solution . of load and centralized 2. No Deadlock Situation. 3. Better synchronization of ants	

CONCLUSIONS AND FUTURE ENHANCEMENT

In this paper, we surveyed multiple ACO algorithms for load balancing in cloud computing environment. We discussed the factors should consider for performance Evaluation of Load balancing Algorithm. We Discussed different Nature inspired technologies and Ant Colony System (ACS) terminology. This paper focused on different ACO Load Balancing Algorithms with their pros and cons. Based on the study, an algorithm should be developed that it meets the minimum requirements.

REFERENCES

1. K. Dasgupta, B. Mandai, P. Dutta, J. K. Mandai, and S. Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing," International Conference on Computational Intelligence Modeling Techniques and Applications, vol. 10, pp. 340-347,2013.
2. F. Ramezani, J. Lu, and F. K. Hussain, "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization," Int J Parallel Prog International Journal of Parallel Programming, vol. 42, no. 5, pp. 739-754,2013.
3. B. K ruekaew and W. Kim pan, "Virtual Machine Scheduling Management on Cloud Computing Using Artificial Bee Colony," Proceedings of the International Multi Conference of Engineers and Computer Scientists, vol. 1, 2014.
4. M. Dorigo, C. Blum / Theoretical Computer Science 344 (2005) 243 – 278.
5. Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang, “Cloud Task scheduling based on Load Balancing Ant Colony Optimization”, 2011 Sixth Annual China Grid Conference, 978-0-7695- 4472-4/11, 2011 IEEE.
6. Z. Zhang, and X. Zhang, “A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation”, Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pages 240-243.

7. S. Banerjee, I. Mukherjee and P.K. Mahanti, “Cloud Computing Initiative uses Modified Ant Colony Framework”, World Academy of Science and Technology, 56, pp. 221-224, 2009.

8. R. Rastogi Kumar Nishant, Pratik Sharma, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization." Proceedings of the 14th International Conference on Computer Modeling and Simulation (UKSim), IEEE, pp: 3-8, March 2012.

9. S.C. Wang, K.Q. Yan, W.P. Liao and S.S. Wang, “Towards a Load Balancing in a Three-level Cloud Computing Network”, Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, pp. 108-113, 2010.

10. V. Sakthivelmurugan, A. Saraswathi, and R. Shahana, “Enhanced load balancing technique in a public cloud,” IJREAT International Journal of Research in Engineering & Advanced Technology vol. 2, no. 2, pp. 1–4, 2014.

11. R. Somani and J. Ojha, “A hybrid approach for vm load balancing in the cloud using cloudsim,” International Journal of Science, Engineering and Technology Research (IJSETR), vol. 3, no. 6, pp. 1734–1739, 2014.

12. K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, and J. Al-Jaroodi, “A survey of load balancing in cloud computing: challenges and algorithms,” Second Symp. Netw. Cloud Comput. Appl., pp. 137–142, Dec. 2012.

13. I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas, “A Flexible Accounting Model for Cloud Computing,” in Annual SRII Global Conference, pp. 277– 284, 2011.

AUTHOR DETAILS

Mrs. M. Padmavathi working as Asst. Professor in Computer Science and Technology Dept. She Registered her Ph. D from JNTU (H) in the Academic Year 2012-13.Her Research Area is Cloud Computing .Her Specialization is Resource Allocation in Iaas. Up to now Published 8 Articles in International Journals, s2 in National Journals and two in IEEE-Explore

Prof.MD. Mahaboob Basha Received an Engineering Degree in Electronics from Bangalore Univerisy, M. Tech (DS&CE) from JNTU and Ph. D in Image Processing From S.K. University. Currently He is Professor in Al Habeeb College of Engineering & Technology, Hyderabad. His research areas include Cloud Computing, Network Security, Image and Video Processing