

CLOUD COMPUTING IN AIRCRAFT NETWORK

P.Shakthikumaran¹, ²T.Nalini, ³J.Rama

¹ student, B.Tech (C.S.E), Bharath University, Chennai – 73.

^{2,3} Professor, Bharath University, Chennai -73.

Shanpan85@gmail.com

¹Rama_j1@yahoo.com

Abstract - Load Balancing is a method of distributing workload across many servers in a network. Typical datacenter implementations rely on large, powerful (and expensive) computing hardware and network infrastructure, which are subject to the usual risks associated with any physical device, including hardware failure, power and/or network interruptions, and resource limitations in times of high demand. Load balancing in the cloud differs from classical thinking on load-balancing architecture and implementation by using commodity servers to perform the load balancing. This provides for new opportunities and economies-of-scale, as well as presenting its own unique set of challenges. Load balancing is used to make sure that none of your existing resources are idle while others are being utilized. To balance load distribution, you can migrate the load from the source nodes (which have surplus workload) to the comparatively lightly loaded destination nodes. When load balancing is applied during runtime, it is called dynamic load balancing — this can be realized both in a direct or iterative manner according to the execution node selection: In the iterative methods, the final destination nodes are determined through several iteration steps. In the direct methods, the final destination node is selected in one step. Another kind of Load Balancing method can be used i.e. the Randomized Hydrodynamic Load Balancing method, a hybrid method that takes advantage of both direct and iterative methods.

Keywords - Cloud Software as a service, Elastic Compute Cloud, ant colony optimization, Amazon Machine Image, mobile agents, Cloud Infrastructure as a Service

I INTRODUCTION

In this paper, we proposed an algorithm for load distribution of workloads among nodes of a cloud by the use of Ant Colony Optimization (ACO). This is a modified approach of ant colony optimization that has been applied from the perspective of cloud or grid network systems with the main aim of load balancing of nodes. This modified algorithm has an edge over the original approach in which each ant build their own individual result set and it is later on built into a complete solution. However, in our approach the ants continuously update a single result set rather than

updating their own result set. Further, as we know that a cloud is the collection of many nodes, which can support various types of application that is used by the clients on a basis of pay per use. Therefore, the system, which is incurring a cost for the user should function smoothly and should have algorithms that can continue the proper system functioning even at peak usage hours.

II OBJECTIVE

In non-distributed type, either one node or a group of nodes do the task of load balancing. Non distributed dynamic load balancing algorithms can take two forms: centralized and semi distributed. In the first form, the load balancing algorithm is executed only by a single node in the whole system: the central node.

III EXISTING SYSTEM

Many researchers have invented different techniques for load balancing and distributing the workload in a cloud environment. The servers are grouped under virtual servers (VS), each VS having its own virtual service queues. Each server processing a request from its queue calculates a profit or reward GrenchMark is a framework for generating and submitting synthetic or real workloads to grid computing environments. Over the seven three years, GrenchMark has been used in over 25 testing scenarios in grids (e.g., Globus based), in peer-to-peer systems (e.g., BitTorrent-based), and in heterogeneous computing environments (e.g., Condor-based).

IV PROPOSED SYSTEM

The cloud computing is the development of distributed computing, parallel computing and grid computing, or defined as the commercial implementation of these computer science concepts. One of the fundamental issues in this environment is related to task scheduling. Cloud task scheduling is an NP-hard optimization problem, and many meta-heuristic algorithms have been proposed to solve it. A

good task scheduler should adapt its scheduling strategy to the changing environment and the types of tasks. This paper proposes a cloud task scheduling policy based on Load Balancing Ant Colony Optimization (LBACO) algorithm. The main contribution of our work is to balance the entire system load while trying to minimizing the make span of a given tasks set. The new scheduling strategy was simulated using the CloudSim toolkit package. Experiments results showed the proposed LBACO algorithm outperformed FCFS (First Come First Serve) and the basic ACO (Ant Colony Optimization). The major advantages of this system is sharing a computer system among multiple users, isolating users from each other and from the control program and Emulating hardware on another machine

V PROBLEM DEFINITION

Cloud providers are generally not aware of a specific organization's security and privacy needs. Adjustments to the cloud computing environment may be warranted to meet an organization's requirements. Organizations should require that any selected public cloud computing solution is configured, deployed, and managed to meet their security and privacy requirements. Every industry standard states that there should be security and privacy policies. Security policies are the top-level set of documents of a company. They document company's decision on the protection, sharing and use of information. A complete and appropriate set of policies will help to avoid liability and compliance problems. Cloud computing has been defined by NIST as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or cloud provider interaction.

VI METHODOLOGIES

Methodologies are the process of retrieving exact information what the party looking for. Here is two step frame work is used to retrieve the content.

- a) Client-Server connectivity
- b) Resource Allocation
- c) Job Allocation
- d) Implement Ant Colony Optimization
- e) Design Pheromone table

a) Client-Server connectivity

This is the first module of our project. In this module a client was going to choose one server. The client chosen server in that time connection established between client and server. Particular server maintains the client's information. Here we are design the web application for user interaction.

Input:

- Client name
- Password

Output:

- Client connection page

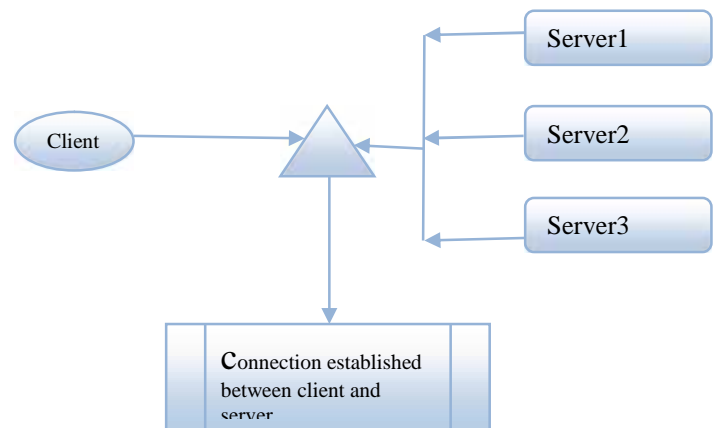


Fig. 1 Client-Server connectivity

b) Resource Allocation:

This is second module of our project. Here a client interact with server allocate resources like upload the files to the server and download the files from server. In particular, for decisions about transitioning organizational data, applications, and other resources to a public cloud computing environment. That can be rapidly provisioned and released with minimal management effort or cloud provider interaction.

Input:

- Upload the file to the selected server
- Download the files from selected server

Output:

- Get the acknowledgment message from server

c) Job Allocation

This is the third module of our project. Here a client interact with server allocates jobs like text to pdf conversion from server and video compression from server. The part of the dynamic load balancing algorithm which selects a job for transferring from a local node to a remote node is referred to as Transfer policy or Transfer strategy. A client sends text file to the server that file converted as pdf file as resend to the client as well as client sends the video file to

the server that file compressed their size and resend to the client.

Input:

- Client sending the text file to server

Output:

- The text file converted as pdf file

d) *Load balancing*

This is fourth module of our project. In this module we are going to balancing the cloud servers depend upon their load. Load means like conversion or compression, a client send the text or video file to the server if a server was maintain heavy load then load will be forward to the another server means neighbor server Individual client's loads are behaviorally much unsophisticated insects. They have a very limited memory and exhibit individual behavior that appears to have a large random component. Acting as a collective however, client's loads manage to perform a variety of complicated tasks with great reliability and consistency.

Input:

- Server has send to client's load their neighbor server

Output:

- Neighbor server sends the response to the client.

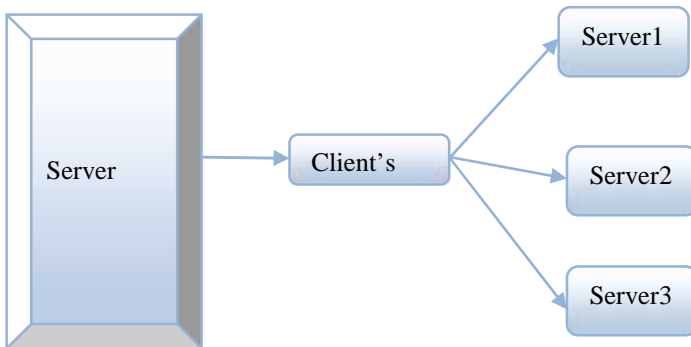


Fig. 2 Load Balancing

e) *Design Pheromone table*

This is last module of our project. To design pheromone table depend upon server's status We replaced the routing tables in the network nodes by tables of probabilities, which we will call 'pheromone tables', as the pheromone strengths are represented by these probabilities. Every node has a pheromone table for every possible destination in the network, and each table has an entry for every neighbor. The entries in the tables are the probabilities which

influence the client's load selection of the next server on the way to their destination server. Here destination server will choose the client's load at dynamically

Input:

- Server maintains all the server details in the cloud.

Output:

- Shows the status of servers

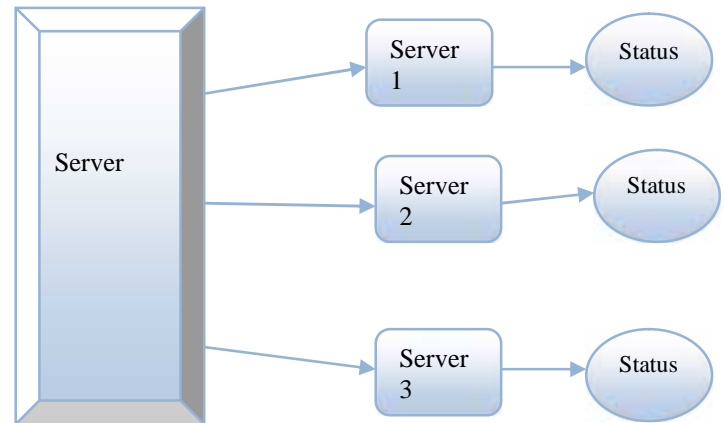


Fig 3 Design Pheromone table

VII TECHNIQUE OR ALGORITHM

a) *Load balancing*

The problem-solving paradigm of ACO, this example illustrates the use of two sets of mobile agents (that act as routing packets) for establishing call connections in a circuit-switched network. To establish connections between gateways 1 and 3, the two groups of mobile agents construct, manipulate and consult their own routing tables. In ACO, each group of mobile agents corresponds to a colony of ants, and the routing table of each group corresponds to a pheromone table of each colony. Even though the two groups of mobile agents (MAG1 and MAG2) may have their own routing preferences, they also take into consideration the routing preferences of the other group.

This increases the chance of distributing data traffic between gateways 1 and 3 between the two connections and the same destination Gateway3, MAG1 is more likely to move along whereas MAG2 is more likely to move along. By adopting the MACO approach, it may be possible to reduce the likelihood that all mobile agents establish connections using only the optimal path. If MAG2 selects the optimal path, the idea of repulsion may increase the

probability that MAG1 will select an alternative to. The advantage of using MACO in circuit-switched routing is that it is more likely to establish connections through multiple paths to help balance the load but does not increase the routing overhead. An on-going work implements some of the ideas of ACO in a test bed, and preliminary empirical results seem to suggest that using the same number of mobile agents (routing packets), it is more likely that ACO can establish connections through multiple paths while traditional

VIII IMPLEMENTATION

a) Resource Allocation:



Fig 4 screen shot for resource allocation

b) Conversion Text To Pdf

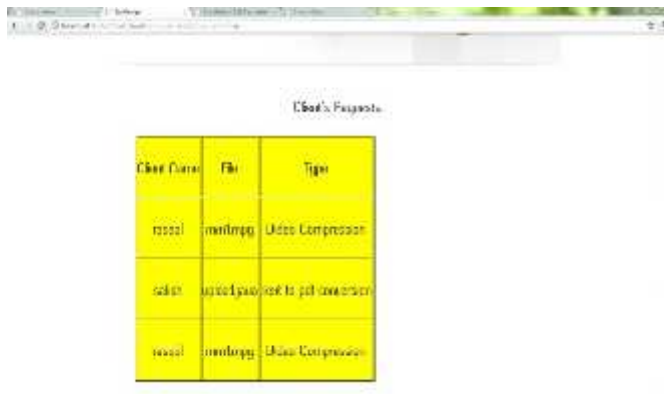


Fig 5 screen shot for conversion text to pdf

c) Video Compression Request:



Fig 6 Screen shot video compression request

IX APPLICATION

Here mining the user information at the registration process and then detecting the user behaviors based upon his/her profile and then satisfying user needs at the right time. Then detecting masquerade's and secures user profiling. An optimal redistribution algorithm, which minimizes the Euclidean norm of the migrating load, is derived. The problem is further studied by modeling with the unsteady heat conduction equation. Relationship between this algorithm and other dynamic load balancing algorithms is discussed. Convergence of the algorithm for special graphs is studied.

X FUTURE ENHANCEMENTS

The pheromone update mechanism has been proved as a efficient and effective tool to balance the load. This modification supports to minimize the make span of the cloud computing based services and portability of servicing the request also has been converged using the ant colony optimization technique. This technique does not consider the fault tolerance issues. Researchers can proceed to include the fault tolerance issues in their future researches.

XI CONCLUSION

Till now we have discussed on basic concepts of Cloud Computing and Load balancing. In addition to that, the load balancing technique that is based on Swarm intelligence has been discussed. We have discussed how the mobile agents can balance the load of a cloud using the concept of Ant colony Optimization. The limitation of this technique is that it will be more efficient if we form cluster in our cloud .So, the research work can be proceeded to implement the total solution of load balancing in a complete cloud environment.

XII REFERENCE & BILOGRAPHY

1. Dancho Danchev, "Building and Implementing a successful Information Security Policy" windowsecurity.com- Windows Security Resources for IT admins.
2. David Escalante and Andrew J. Korty, Cloud Services: Policy and Assessment, EDUCAUSE Review, vol. 46, no. 4 (July/August 2011)
3. Richard N. Katz, "Looking at Clouds from All Sides Now", EDUCAUSE Review, vol. 45, no. 3 (May/June 2010): 32-45
4. Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, Cloud Computing A Practical Approach, TATA McGRAW-HILL Edition 2010.
5. Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.
6. Mladen A. Vouk, Cloud Computing Issues, Research and Implementations, Proceedings of the ITI 2008 30th Int. Conf. on Information Technology Interfaces, 2008, June 23-26.
7. Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.
8. ibm.com/press/us/en/pressrelease/22613.wss
9. <http://www.amazon.com/gp/browse.html?node=201590011>
10. Martin Randles, EnasOdat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.
11. Peter S. Pacheco, "Parallel Programming with MPI", Morgan Kaufmann Publishers Edition 2008
12. MequanintMoges, Thomas G.Robertazzi, "Wireless Sensor Networks: Scheduling for Measurement and Data Reporting", August 31, 2005
13. Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.
14. Martin Randles, EnasOdat, David Lamb, Osama Abu- Rahmeh and A. Taleb-Bendiab, "A Comparative Experiment in Distributed Load Balancing", 2009 Second International Conference on Developments in eSystems Engineering.
15. Fourth International Conference on Semantics, Knowledge and Grid" Load Balancing in Nondedicated Grids Using Ant Colony Optimization".
16. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid