

LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING

Doddini Probhuling L.

Dept. of computer science, Shivaji Plytechnic, Karad, India.

[Received-15/03/2013, Accepted-27/03/2013]

ABSTRACT:

In present paper, discussed with the cloud computing requirements for access control, migration, security, data availability, trust issues and sensitive information.

Beside this reviewed the algorithms used in the cloud computing for load balancing; weighted active monitoring load balancing algorithm, dynamic load balancing, static algorithms, dynamic algorithms, ant colony optimization algorithms, and load balancing in distributed systems

Keywords: load balancing algorithm, dynamic load balancing, static algorithms, dynamic algorithms, ant colony optimization algorithms, and load balancing in distributed systems.

INTRODUCTION:

Cloud computing is Internet based computing, whereby shared resources, software and information are provided to computers and other devices on-demand, like a public utility. [1]

As cloud computing is in its evolving stage, so there are many problems prevalent in cloud computing [2][4].

Such as:

- I.** Ensuring proper access control (authentication, authorization, and auditing)
- II.** Network level migration, so that it requires minimum cost and time to move a job
- III.** To provide proper security to the data in transit and to the data at rest.

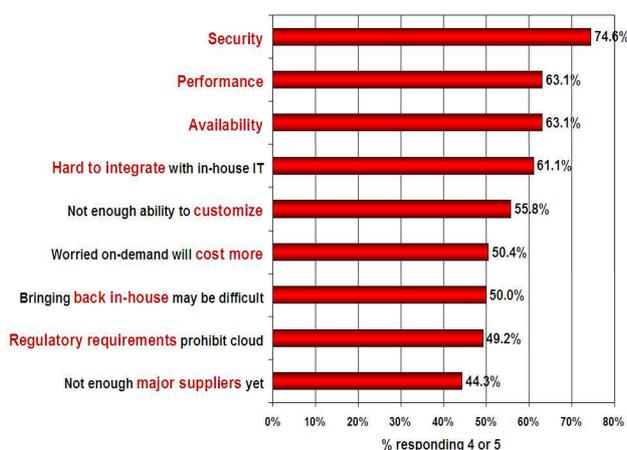
IV. Data availability issues in cloud

V. Legal quagmire and transitive trust issues

VI. Data lineage, data provenance and inadvertent disclosure of sensitive information is possible And the most prevalent problem in Cloud computing is the problem of load balancing. Further, while balancing the load, certain types of information such as the number of jobs waiting in queue, job arrival rate, CPU processing rate, and so forth at each processor, as well as at neighboring processors, may be exchanged among the processors for improving the overall performance. For this purpose various types of algorithms have been proposed and in

this paper we have tried to find the problems in the existing algorithms on the basis of some common criteria which we have termed as metrics [3]. The following figure 1 [4] shows the issues which are existing in cloud computing and we can see that the issues of performance, availability etc. are due to lack of proper load balancing algorithms.

Q: Rate the challenges/issues ascribed to the 'cloud'/on-demand model
(1=not significant, 5=very significant)



Source: IDC Enterprise Panel, August 2008 n=244

Load Balancing in Cloud Computing

Load balancing is a relatively new technique that facilitates networks and resources by providing a maximum throughput with minimum response time [6]. Dividing the traffic between servers, data can be sent and received without major delay. Different kinds of algorithms are available that helps traffic loaded

between available servers [5]. A basic example of load balancing in our daily life can be related to websites.

Without load balancing, users could experience delays, timeouts and possible long system responses. Load balancing solutions usually apply redundant servers which help a better distribution of the communication traffic so that the website availability is conclusively settled [6].

Algorithms in load balancing

1. WEIGHTED ACTIVE MONITORING LOAD BALANCING ALGORITHM

The 'Weighted Active Monitoring Load Algorithm' is implemented; modifying the Active Monitoring Load Balancer by assigning a weight to each VM as discussed in Weighted Round Robin Algorithm of cloud computing in order to achieve better response time and processing time.[7]

2. Dynamic Load Balancing Algorithms:

The three methods are:

- **SA** - Simulated Annealing: We directly minimize the above cost function by a process analogous to slow physical cooling.
- **ORB** - Orthogonal Recursive Bisection: A simple method which cuts the graph into two by a vertical cut, then cuts each half into two by a horizontal cut, then each quarter is cut vertically, and so on.
- **ERB** - Eigenvector Recursive Bisection: This method also cuts the graph in two then each half into two, and so on, but the cutting is done using an eigenvector of a matrix with the same sparsity structure as the adjacency matrix of the graph. The method is an approximation to a computational neural net. [8]

Static algorithms

Static algorithms divide the traffic equivalently between servers. By this approach the traffic on the

servers will be disdained easily and consequently it will make the situation more imperfectly. This algorithm, which divides the traffic equally, is announced as round robin algorithm. However, there were lots of problems appeared in this algorithm. Therefore, weighted round robin was defined to improve the critical challenges associated with round robin. In this algorithm each servers have been assigned a weight and according to the highest weight they received more connections. In the situation that all the weights are equal, servers will receive balanced traffic [9].

Dynamic algorithms

Dynamic algorithms designated proper weights on servers and by searching in whole network a lightest

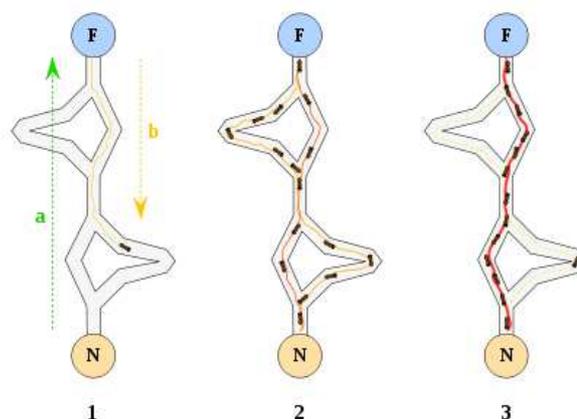
server preferred to balance the traffic. However, selecting an appropriate server needed real time communication with the networks, which will lead to extra traffic added on system.

In comparison between these two algorithms, although round robin algorithms based on simple rule, more loads conceived on servers and thus imbalanced traffic discovered as a result [9].

Ant colony optimization algorithms

Genetic Algorithms (GA) have been used to evolve computer programs for specific tasks, and to design other computational structures. The recent resurgence of interest in AP with GA has been spurred by the work on Genetic Programming (GP). GP paradigm provides a way to do program induction by searching the space of possible computer programs for an individual computer program that is highly fit in solving or approximately

solving the problem at hand[10,11]. The genetic programming paradigm permits the evolution of computer programs which can perform alternative computations conditioned on the outcome of intermediate calculations, which can perform computations on variables of many different types, which can perform iterations and recursions to achieve the desired result, which can define and subsequently use computed values and subprograms, and whose size, shape, and complexity is not specified in advance. GP use relatively low-level primitives, which are defined separately rather than combined a priori into high-level primitives, since such mechanism generate hierarchical structures that would facilitate the creation of new high-level primitives from built-in low-level primitives [12-14].



1. The first ant finds the food source (F), via any way (a), then returns to the nest (N), leaving behind a trail pheromone (b)
2. Ants indiscriminately follow four possible ways, but the strengthening of the runway makes it more attractive as the shortest route.
3. Ants take the shortest route; long portions of other ways lose their trail pheromones.

Load Balancing in Distributed Systems

Distributed system load balancing is still an active area of research in which load balancer attempts to improve the performance of a distributed system by using the processing power of the entire system to smooth out periods of high congestion at individual nodes [15,16], this is done by transferring some of the workload of heavily loaded nodes to other nodes for processing. Decisions on how to balance loads among the nodes are either static [17-20] or dynamic [21-26]

CONCLUSIONS:

In present paper, discussed with the cloud computing requirements for access control, migration, security, data availability, trust issues and sensitive information. The algorithms used in the cloud computing for load balancing, this information might be useful in the research associated with cloud computing

REFERENCES:

1. Manisha B. Jadhav, Vishnu J. Gaikwad, C. V. Patil, G. S. Deshpande, 2010. CLOUD COMPUTING APPLICATIONS IN COMPUTATIONAL SCIENCE, International Journal of Advanced Computer and Mathematical Sciences.. Vol 1, Issue 1. Pp 1-6
2. T.R.V. Anandharajan, Dr. M.A. Bhagyaveni” Co-operative Scheduled Energy Aware Load-Balancing technique for an Efficient Computational Cloud” IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.
3. Ruchir Shah, Bhardwaj Veeravalli, Senior Member, IEEE, and Manoj Misra, Member, IEEE “On the Design of Adaptive and Decentralized Load-Balancing Algorithms with Load Estimation for Computational Grid Environments” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 18, NO. 12, DECEMBER 2007.
4. Wayne Jansen Timothy Grance” Guidelines on Security and Privacy in Public Cloud Computing” NIST Draft Special Publication 800-144.
5. A. Brian. Load Balancing in the Cloud: Tools, Tips, and Techniques. A Technical white paper in Solutions Architect, Right Scale.
6. R. Shimonski. *Windows 2000 & Windows Server 2003 Clustering and Load Balancing*. Emeryville. McGraw-Hill Professional Publishing, CA, USA (2003), p 2, 2003.
7. Jasmin James and Bhupendra Verma. 2012, EFFICIENT VM LOAD BALANCING ALGORITHM FOR A CLOUD COMPUTING ENVIRONMENT, International Journal on Computer Science and Engineering, Vol. 4 No. 09, pp1658-1663
8. Roy D. William, 199, Performance of Dynamic Load Balancing Algorithms for Unstructured Mesh Calculations. C3P 913. pp1-20
9. R. X. T. and X. F. Z.. A Load Balancing Strategy Based on the Combination of Static and Dynamic, in Database Technology and Applications (DBTA), 2010 2nd International Workshop (2010), pp. 1-4.
10. A. Abraham et al.: Evolutionary Computation: from Genetic Algorithms to Genetic Programming, Studies in Computational Intelligence (SCI) 13, 1–20 (2006), www.springerlink.com c _ Springer-Verlag Berlin Heidelberg 2006.
11. C. Grosan and A. Abraham: Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews, Studies in Computational Intelligence (SCI) 75, 1–17 (2007), www.springerlink.com c_Springer-Verlag Berlin Heidelberg 2007.
12. Héctor A Montes and Jeremy L Wyatt ,”Cartesian Genetic Programming for Image Processing Tasks”,
13. John R. Koza, Margaret Jacks Hall, “ SURVEY OF GENETIC ALGORITHMS AND GENETIC PROGRAMMING”, <http://www-csfaculty.stanford.edu/~koza/>.
14. Riccardo Poli, William B. Langdon , Nicholas F. McPhee, John R. Koza, “Genetic Programming :AnIntroductory Tutorial and a Survey of Techniques and pplications” , Technical Report CES-475 ISSN: 1744-8050 October 2007. essex.ac.uk/dces/research/publications/.../2007/ces475.pdf Appendix
15. Ferrari, D. and S. Zhou, 1987. An empirical investigation of load indices for load balancing applications. Proceeding of the 12th International Symposium on Computer Performance Modeling Measurement and Evaluation, Dec. 7-9, North-Holland Publishing Co., Amsterdam, The Netherlands, pp: 515-528. <http://portal.acm.org/citation.cfm?id=725013>
16. Zhou, S. and D. Ferrari, 1987. An experimental study of load balancing performance. University of California, Berkeley, CA., pp: 26.
17. Rao, G.S., H.S. Stone and T.C. Hu, 1979. Assignment of tasks in a distributed processor system with limited memory. IEEE Trans. Comput., C-28: 291-299. <http://portal.acm.org/citation.cfm?id=1310399>
18. Tantawi, A.N. and D. Towsley, 1985. Optimal static load balancing in distributed computer systems. J. Assoc. Comput. Mach., 32: 445-465.
19. Pinter, S.S. and Y. Woltstahl, 1987. On mapping processes to processor in distributed systems. Int. J. Parall. Programm., 16: 1-15.
20. Chen, M.S. and K.G. Shin, 1990. Subcube allocation and task migration in hypercube multiprocessor. IEEE Trans. Comput., 39: 1146-1155.

21. Ni, L.M., C.W. Xu and T.B. Gendreau, 1985. A distributed drafting algorithm for load balancing. *IEEE Trans. Software Eng.*, SE-11: 1153-1161.
22. Eager, D.L., E.D. Lazowska and J. Zahorjan, 1986. Adaptive load sharing in homogeneous distributed systems. *IEEE Trans. Software Eng.*, 12: 662-673.
23. Shin, K.G. and Y.C. Chang, 1989. Load sharing in distributed real-time systems with state-change broadcasts. *IEEE Trans. Comput.*, 3: 1124-1142.
24. Xu, J. and K. Hwang, 1993. Heuristic methods for dynamic load balancing in a message-passing multicomputer. *J. Paral. Distribut. Comput.*, 18: 1-13.
25. Ali, A.D., 2000. A robust load balancer for heterogeneous distributed systems type-A. *Proceeding of the Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 16-20, Vancouver, British Columbia, Canada, p: 1-1. <http://www.scs.org/confernc/scsc/scsc00/text/spec2.html>
26. Ali, A.D., 2001. A dynamic cluster constructor for load balancing in big heterogeneous distributed systems. *Proceeding of the Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 15-19, Orlando, Florida, USA., pp: 47-55.