Dynamic load balancing in distributed memory systems

Davit Siradeghyan
SEUA
Yerevan, Armenia
e-mail: davitsiradeghyan@gmail.com

Gevorg Margarov
SEUA
Yerevan, Armenia
e-mail: gmargarov@gmail.com

ABSTRACT

Cloud computing is a model which can be used in any domain expanding from large scale enterprise to startups to educational institutions [1]. Memory group in distributed systems connected by network can be treated as a big memory system for educational systems. One of the key advantages in cloud technologies is a fact that computational resources can be allocated and de-allocated dynamically [2].
We are using cloud technologies in our system to implement memory model. The objective of this paper is a load balancing algorithm which can be used in distributed memory systems.

Keywords

Cloud computing, load balancing, distributed systems, distributed memory systems.

1. INTRODUCTION

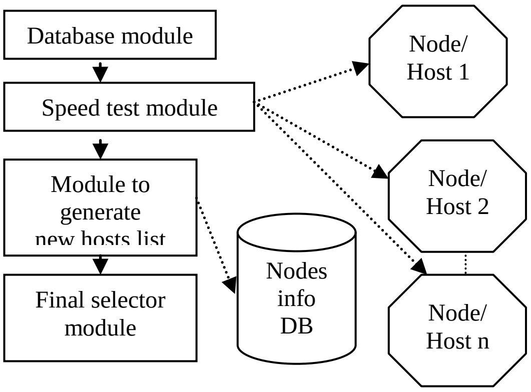
Often cloud memory storages are implemented as a complex and multi-level systems, which are formed as a group of independent machines and they are very good solution for “e-commerce” systems [3]. As memory consists from many pieces of independent memories there can be situations when some modules are overloaded when others not. The purpose of load balancing [4], [5]. [6] is to avoid such kind of situations. This paper represents load balancing algorithm and scheme for cloud technologies with practical results of usage.
We are suggesting the following solution for dynamic load balancing for the load balancer module developed by us:

1. Select load balancing principle (by the smallest time, by the most appropriate storage available, mix of time and storage).
2. Get list of memory nodes from database.
3. Do speed test between information source and memory nodes.
4. Make new table of nodes based on speed test results. At the top we will have nodes which have the smallest speed test time with the source.
5. Select nodes from new table which have enough memory to store necessary information.
6. From the last selected list choose a node which has minimal speed test time and maximum appropriate free space.

2. LOAD BALANCING ALGORITHM

2.1. Speed test between file nodes and information source

Speed test will be implemented based on “ping” principle. Scheme below represents a solution to choose a node for load balancing.



As we can see from the scheme above node selection is based on few parameters values. The first parameter is a speed between host and information source, the second parameter is a capacity. The best server will be the one which has maximum appropriate free space and at the same time good speed test results. For example, if we want to post a file with 5Mb size and we have 2 servers with 6 and 10 Mb available spaces, the best solution will be the first one as; using it we will “loose” only 1Mb, so less fragmentation more space to use. Speed between 2 hosts will be calculated as an average value for 8 sequential ping results.

2.2. The best node selection from the selected short list

To explain this process we will use sample data. Two tables below will contain information about average speed and available free space. The first table contains an average speed between the node and information source, the second table contains free spaces available on each node.

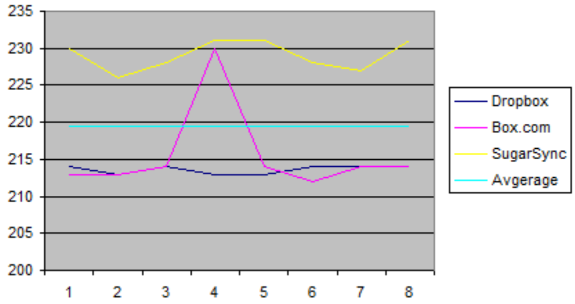
Node name or address	Ping average value
host1.example.com	215ms
host2.example.com	229ms
host3.example.com	213ms

Node name or address	Free space
host1.example.com	10Mb
host2.example.com	2Mb
host3.example.com	15Mb

Imagine we need to upload 8Mb file from the source to the node. The first table will show an average speed between the source and nodes. As we can see host3 has the best speed between source and node. Next we need to check free spaces. As we can see host2 hasn’t enough free space so at the end we will have host1 and host3 selected. As host1 has the best matched free space and its ping time is acceptable for us, not the worst in the table load balancer will choose host1.

3. PRACTICAL TEST RESULTS

We did several tests and used dropbox, box, sugarsync cloud services as host services. Our research shows that we have best results if use combined privileges for load balancing (equal privileges for speed and capacity). During the tests we got the following results.



4. CONCLUSION

It is very important to consider different parameters for load balancing. As a result we will have a very flexible dynamic load balancing module, which can make decisions based on various parameters and even mix them based on requirements. This load balancing module also very flexible as it can be easily integrated with not only local memory systems, but also can be configured to use favorite cloud file storages like dropbox, box.com, etc.

REFERENCES

[1] Akula Sai Kiran from Guru Nanak, “Cloud hosting with load balancing: An effective way to serve the web”, www.academia.edu.

[2]. Menno Dobber, Ger Koole, Rob van der Mei , “Dynamic Load Balancing Experiments in a Grid”, Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium, pp. 1063 - 1070 Vol. 2, 2005.

[3]. Resat Umit Payli, Kayhan Erciyes and Orhan Dagdeviren, “A cluster-based dynamic load balancing protocol for grids”, Third International Conferences, WiMo 2011 and CoNeCo 2011, pp. 315-324, 2011.

[4]. Karen D. Devine, Erik G. Boman, Robert T. Heaphy, Bruce A. Hendrickson, “New Challenges in Dynamic Load Balancing”, ADAPT '03: Conference on Adaptive Methods for Partial Differential Equations and Large-Scale Computation, pp. 133-152, 2005.

[5]. James Dinan, Stephen Olivier, Gerald Sabin, Jan Prins, P. Sadayappan, and Chau-Wen Tseng, “Dynamic Load Balancing of Unbalanced Computations Using Message Passing”, International Parallel and Distributed Processing Symposium/International Parallel Processing Symposium - IPDPS(IPPS), pp. 1-8, 2007.

[6]. Abbas Karimi, Faraneh Zarafshan, Adznan b. Jantan, A.R. Ramli, M. Iqbal b.Saripan, “A New Fuzzy Approach for Dynamic Load Balancing Algorithm”, International Journal of Computer Science and Information Security, IJCSIS, Vol. 6, No. 1, pp. 1-5, 2009.