Appraise of Load Balancing Techniques in Cloud Computing

Lanitha B, Raju M

***Abstract*— Cloud computing is definitely a new technology which uses virtual machine as a substitute for physical machine to host, store and network multiple components. The increase in web traffic and different services are increasing everyday making load balancing. As cloud computing is growing rapidly and better users are attracted towards utility computing, better and fast service ought to be provided. For better management of available good load balancing techniques are required. Better load balancing in cloud, performance is increased and user gets better services.**

***Index Terms*— Cloud Computing and Load Balancing.**

I. INTRODUCTION

Load balancing is an even division of processing work between two or more computers and/or CPUs, network links, storage devices or other devices, ultimately delivering faster service with higher efficiency. Load balancing is accomplished through software, hardware or both, and it often uses multiple servers that appear to be a single computer system [17].

Goals of Load balancing as given in [16], the goals of load balancing are:

1. To improve the performance substantially.
2. Undertake a backup plan should the system fails even partially.
3. To retain the system stability.
4. To fit future modification inside system.

II. KINDS OF LOAD BALANCE ALGORITHM

Depending on who initiated accomplishing this, load balancing algorithms can constitute three categories as given in [16]:

1. Sender Initiated: If load balancing algorithm is initialized through the sender.
2. Receiver Initiated: If load balancing algorithm is initiated through the receiver.
3. Symmetric: It is the mixture of both sender initiated and receiver initiated.

Manuscript received April, 2014.

***Lanitha B** is a Assistant Professor, Department of Computer Science at KGiSL Institute of Technology, Coimbatore, Tamilnadu, India. Her research interest include Digital Image Processing, , Data mining, Wireless Sensor Networks, Cloud Computing and Information Systems.*

***Raju M** is a PG Scholar (ME (CSE)), Department of Computer Science in CMS College of Engineering, Namakkal, Tamilnadu, India. His research areas of interest include Wireless Sensor Networks, and Cloud Computing.*

Dependent upon the current state of your system, load balancing algorithms could be divided into two categories as given in [16]:

1. Static: It doesn't depend around the current state of your system. Prior knowledge of your system should be applied.
2. Dynamic: Decisions on load balancing depend on current state of your system. No prior knowledge is needed. Then it provides multiple advances over static approach.

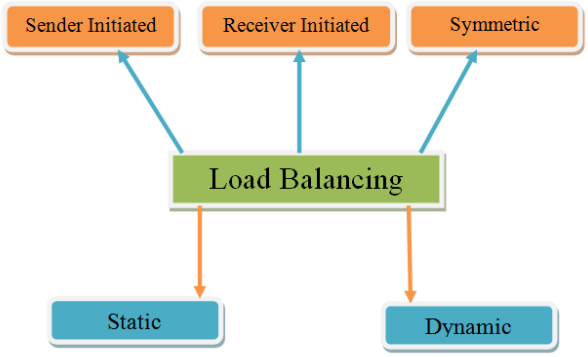

Fig. 1 Categories of load balancing algorithm

Difference between static and dynamic load balancing:

- The most significant difference between static and dynamic LB is without needing to allocate idle resources for dynamic Load balancing. Resources are allocated when asked and released when no longer required. This minimizes expensive allocation of idle resources.
- Adding new instances into static load balance configuration requires some time to testing, but with dynamic load balance everything will be automatic.
- Manual mistakes can occur in static load balance but aren't present in dynamic load balancing
- Environment change approvals are important from management to feature new instances into static environments yet not for dynamic environments.
- The challenge in developing applications is making them state agnostic. By creating applications that allow its sessions for being transferred derived from one of site to another provides for dynamic allocation.

III. LOAD BALANCING TECHNIQUES

A. Decentralized content aware [1]:

- It is just a Distributed computing environment.
- Works on the unique and special property (USP) of requests and computing nodes to help scheduler to find out the very best node for processing the requests.

ISSN: 2278 – 1323

- Uses this content information to narrow down the search.
- Advance the searching performance hence enhancing overall performance.
- Reduces idle time of the nodes.

B. LB for Internet distributed services [2]:

- It is just a Distributed web server's environment.
- Works on the protocol to limit redirection rates to avoid remote servers overloading.
- Works on the middleware to guide this protocol.
- Works on the heuristic to tolerate abrupt load Changes.
- Reduces service response times by redirecting requests to the closest servers without overloading them.
- Mean response time is 29% smaller than RR (Round Robin) and 31% smaller than SL (Smallest Latency).

C. Join-Idle-Queue [3]:

- It is just a Cloud data centers environment.
- First assigns idle processors to dispatchers for the availability of the idle processors at each dispatcher.
- Then assigns jobs to processors to reduce average queue length of jobs at each processor.
- Effectively reduces the machine load.
- Incurs no communication overhead at job arrivals.
- Doesn't increase actual response times.

D. Lock-free multiprocessing [4]:

- It is just a Multi-core environment.
- Runs multiple load-balancing processes in one load balancer.
- Improves overall performance of load balancer.

E. Scheduling strategy on LB of VM resources [5]:

- It is just a Cloud Computing environment.
- Uses Genetic algorithm, historical data and current state of system to achieve best load balancing and to reduce dynamic migration.
- Solves the problems of load imbalance and high migration cost.

F. Central LB policy for VMs [6]:

- It is really a Cloud Computing environment.
- Uses global state information to create load balancing decisions.
- Balances the strain evenly to enhance overall performance.
- Around 20% improvements in performance.
- Does not consider fault tolerance.

G. LBVS: LB technique for Virtual Storage [7]:

- It is really a Cloud Storage environment.
- Uses Fair-Share Replication strategy to reach Replica Load balancing module which in turn controls the access load balancing.
- Uses writing balancing algorithm to regulate data writing load balancing.
- Enhances flexibility and robustness.
- Provides large scale net data storage and storage as a service.

H. Task Scheduling Centered on LB [8]:

- It is really a Cloud Computing environment.
- First maps tasks to virtual machines and then virtual machines to host resources.
- Improves task response time.
- Improves resource utilization.

I. Honeybee Foraging Behaviour [9]:

- It is really a large scale Cloud Systems environment.
- Achieves global load balancing through local serve actions.
- Performs well as system diversity increases.
- Does not increase throughput as system size increases.

J. Biased Random Sampling [9]:

- It is really a large scale Cloud Systems environment.
- Achieves load balancing across all system nodes using random sampling of the system domain.
- Performs better with high and similar population of resources.
- Degrades as population diversity increases.

K. Active Clustering [9]:

- It is really a large scale Cloud systems environment.
- Optimizes job assignment by connecting similar services by local re-wiring.
- Performs better with high resources.
- Utilizes the increased system resources to improve throughput.
- Degrades as system diversity increases.

L. ACCLB (Ant Colony and Complex Network Theory) [10]:

- It is definitely an Open Cloud Computing Federation environment.
- Uses small-world and scale-free characteristics of complex network to achieve better load balancing.
- Overcomes heterogeneity.
- Adaptive to dynamic environments.
- Excellent in fault tolerance.
- Good scalability.

M. Two-phase scheduling (OLB + LBMM) [11]:

- It is just a Three-level Cloud Computing Network environment.
- Uses OLB (Opportunistic Load Balancing) to keep each node busy and uses LBMM (Load Balance Min-Min) to achieve the minimum execution time of every task.
- Efficient usage of resources.
- Enhances work efficiency.

N. Event-driven [12]:

- It is just a Massively Multiplayer Online Games environment.
- Uses complete capacity event as input, analyzes its components and generates the overall game session load balancing actions.
- Capable of scaling up and down a game title session on multiple resources based on the variable user load.
- Occasional QoS breaches only 0.66%.

O. Carton (LB + DRL) [13]:

- It is just a Unifying framework for cloud control environment.
- Uses Load balancing to minimize the associated cost and uses Distributed Rate Limiting for fair allocation of resources.
- Simple.
- Easy to implement.
- Really low computation and communication overhead.

P. Compare and Balance [14]:

- It is definitely an Intra-Cloud environment.
- Predicated on sampling.
- Uses adaptive live migration of virtual machines.
- Balances load amongst servers.

- Reaches equilibrium fast.
- Assures migration of VMs from high-cost physical hosts to low-cost host.
- Assumption of having enough memory with each physical host.

Q. VectorDot [25]:

- It is just a Datacenters with integrated server and storage virtualization environment.
- Uses dot product to tell apart node based on the item requirement.
- Handles hierarchical and multidimensional resource constraints.
- Removes overloads on server, switch and storage.

IV. CONCLUSION

An important purpose of load balancing is to fulfill the customer requirement by distributing load dynamically one of the nodes and to bring about maximum resource utilization by reassigning the entire load to individual node. We required of efficient load balancing algorithm for efficient by using resources. Future work specializes in the numerous load balancing algorithms and their applicability in cloud computing environment.

REFERENCES

[1] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), February 2011, pages 370-375.

[2] A. M. Nakai, E. Madeira, and L. E. Buzato, "Load Balancing for Internet Distributed Services Using Limited Redirection Rates", 5th IEEE Latin-American Symposium on Dependable Computing (LADC), 2011, pages 156-165.

[3] Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb, and A. Greenber, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", An international Journal on Performance evaluation, In Press, Accepted Manuscript, Available online 3 August 2011.

[4] Xi. Liu, Lei. Pan, Chong-Jun. Wang, and Jun-Yuan. Xie, "A Lock-Free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), 2011, pages 1-4.

[5] J. Hu, J. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), 2010, pages 89-96.

[6] A. Bhadani, and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE), January 2010.

[7] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, "LBVS: A Load Balancing Strategy for Virtual Storage", International Conference on Service Sciences (ICSS), IEEE, 2010, pages 257-262.

[8] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[9] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, April 2010, pages 551-556.

[10] 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.

[11] S. Wang, K. Yan, W. Liao, and 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 (ICCSIT), Chengdu, China, September 2010, pages 108-113.

[12] V. Nae, R. Prodan, and T. Fahringer, "Cost-Efficient Hosting and Load Balancing of Massively Multiplayer Online Games", Proceedings of the 11th IEEE/ACM International Conference on Grid Computing (Grid), IEEE Computer Society, October 2010, pages 9-17.

[13] R. Stanojevic, and R. Shorten, "Load balancing vs. distributed rate limiting: a unifying framework for cloud control", Proceedings of IEEE ICC, Dresden, Germany, August 2009, pages 1-6,.

[14] Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea, August 2009, pages 170-175.

[15] A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers", Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.

[16] 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.

[17] www.techopedia.com-what is Load Balancing? - Definition from Techopedia

Lanitha B is a Associate Professor, Department of Computer Science in KGiSL Institute of Technology, Coimbatore, Tamilnadu, India. She teaches course for BE Computer Science and Engineering. Her research areas of interest include Digital Image Processing, Biometrics, Data mining, Cryptography, Wireless Sensor Networks, Cloud Computing and Information Systems.

Raju M is a PG Scholar (ME (CSE)), Department of Computer Science in CMS College of Engineering, Namakkal, Tamilnadu, India. His research areas of interest include Wireless Sensor Networks, and Cloud Computing.

ISSN: 2278 – 1323