Danger Theory Based Load Balancing (DTLB) Algorithm for Cloud Computing

Isha Dubey[1], Praneet Saurabh[2]

Department of Computer Science & Engineering, Technocrats Institute of Technology
RGPV University, Bhopal, M.P, India

Abstract

Cloud computing is one of the growing technology in computer science and an established way for sharing and accessing resources via internet. Load balancing is a method of reassigning the total load to the entity nodes of the collective system to create resource utilization useful and to improve the response time of the job, concurrently eliminating a condition in which some of the nodes are over loaded while some others are under loaded. There are many techniques are available for balancing the load on servers but none of the technique provide a way to generate any signal when server is overloaded.

Danger Theory Based Load Balancing (DTLB) algorithm for cloud computing is presented in this paper. Proposed algorithm provide an effective load balancing technique which properly distribute the load among servers and generate three level danger signals when any of the servers are overloaded, under loaded, not available or cross threshold value. Results provide better storage utilization, less response time, performance and better reliability with the help of this algorithm.

Keywords: *Cloud Computing, Load Balancing, Artificial Intelligence, Danger Theory.*

1. Introduction

Cloud computing is the delivery of computing services over the internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. There are lot of distributed servers in cloud computing which are providing on demand services to the consumers with reliability and scalability of data centres [1]. Load balancing is one of the main issues related to cloud computing.

It is always required to share work load among the various nodes of the distributed system to improve the resource utilization and for better performance of the system [2].Artificial immune system (AIS) is a new branch of computer science .It uses natural immune system as a metaphor for solving computational problems [3]. AIS have been used to solve a wide variety of problems that includes Computer Security, Pattern Recognition, Fault Detection, and Data Mining [4]. There are four algorithm exist in AIS which are Immune network theory, negative selection algorithm (NSA), Clonal selection, Danger theory algorithm (DTA). Each algorithm has its own application and has its own scope [5].

Danger Theory firstly proposed by Polly Matzinger has been introduced in immunology. It assumes that cells that undergo unnatural deaths may release danger signals to a small area around that cell. This area is called Danger Zone [5]. The main application of Danger Theory is generate the danger signals. So use this theory in cloud computing for generating the signals and equally distribute the load among the servers.

This paper proposes danger theory based load balancing algorithm which is inspired by danger theory undertakes task to overcome problem of load balancing.

DTLB algorithm properly distributes the load among servers and generates three level danger signals when any of the servers are overloaded, under loaded, not available or cross threshold value.

The paper is organized as follows: Section 1 describes the introduction of paper. Section 2 describes related work with cloud computing, load balancing and danger theory. In Section 3, we provide the detailed description of proposed work: model and algorithm. Section 4 gives the result analysis and in Section 5 we give conclusion to our work.

2. Related Work

Cloud computing provides on demand services over the internet. There are many applications and advantages of cloud computing. In present time load balancing is a

challenging task, with the help of this algorithm system performance is increased and future modifications are possible in the system. Danger theory is a new concept for computer science that is used for generating three level signals and provides security in terms of data. Security in terms of data means if data is not store than it generate a signal and try to store it on another server.

2.1 Cloud Computing

Cloud provides three types of services to the consumers. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS)[6]. In IaaS clusters, virtualized servers and its computational resources are delivered to consumers. Perhaps the best known example is Amazon's Elastic Compute Cloud (EC2) and Simple Storage Service (S3) which provide (managed and scalable) resources as services to the user. It involves offering hardware related services using the principles of cloud computing [6]. A platform refers to prefab software architecture upon which you can build computing solution. It provides core software functionality which would otherwise need to be engineered. Platform as a Service typically makes use of dedicated APIs to control the behaviour of a server hosting engine which executes and replicates the execution according to user requests Examples are Force.com, Google App Engine [6]. In Software as a Service (SaaS) standard application software functionality is offered within a cloud. Examples are Google Docs, SAP Business by design [6].

2.2 Load Balancing

Load balancing is the process of improving the performance of the system by shifting of workload among the processors. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine [7]. There are two types of load balancing algorithms are available.

In static load balancing algorithms jobs are assigned to the systems based upon the capability of the system to process new requests. The process is based on previous knowledge of the system property and capability. These algorithms don't consider dynamic changes in attributes at run time. In addition these algorithms cannot adopt to load changes during run time [8].

Dynamic load balancing algorithms obtain from the different attributes of the system capabilities and network bandwidth. These algorithms rely on grouping of knowledge based on previously gathered information about the systems in the cloud and its run-time properties together as the selected systems process the jobs components. On the basis of attributes gathered and calculated this algorithm allots jobs and reassign them to the system accordingly. Dynamic algorithms require stable monitoring of the systems and job progress and are generally harder to implement. However, Dynamic algorithms are more accurate and its results are more efficient [9].

2.3 Danger Theory

Danger theory is artificial immune system algorithm. It is mainly used for generating danger signals and provides the security of data [10]. It used in various field like computer and network security and detecting some abnormal conditions in systems.

3. Proposed Work

Proposed algorithm equally distributes the load among servers and generates three level danger signals when any of the servers are overloaded, under loaded, not available or cross threshold value.

Cloud sim simulator is used for simulation and firstly storage devices are declared and each have some capacity. When simulation is started and user request is come for storing data then, if primary storage is available so store the data and if it is not available then DTLB generate a danger signals. Algorithm generates three level danger signals. Next time when request is come for storing new data then check the availability of storage. If it is not then generate the signals this process is done until all available storage is full.

Firstly declare and initialize all virtual machine (VM), data centers, bandwidth, processors, processing speed, users and user requests, storage and memory. Danger TheoryVmLoadBalancer maintains an index table of VMs, associated the number of requests currently allocated to the VM. At initiate all VM's have 0 allocations. When a request to allocate a new VM from the DataCenterController arrives, it parses the table and identifies the availability of storage. Allocate load according to the primary availability of the VM's in data center.

Danger signal generated by DTLB, if storage has not enough space or it is not available. It generates three levels of danger signals. If available storage devices cross the threshold then generate the low signal which is called signal 0, if primary storage has not enough space and load is shifted on another storage so generating urgent signal which is represented by signal 1 and if all storage is full or not any available space is exist then generate high signal which is represented by signal 2 and its manage the load if it is possible by transferring the load from one server to another on the bases of data size.

Results provide better storage utilization and performance of the system and minimize response time and data center processing time.

3.1 Proposed Model and Algorithm

Figure 1 show the working of DTLB algorithm.

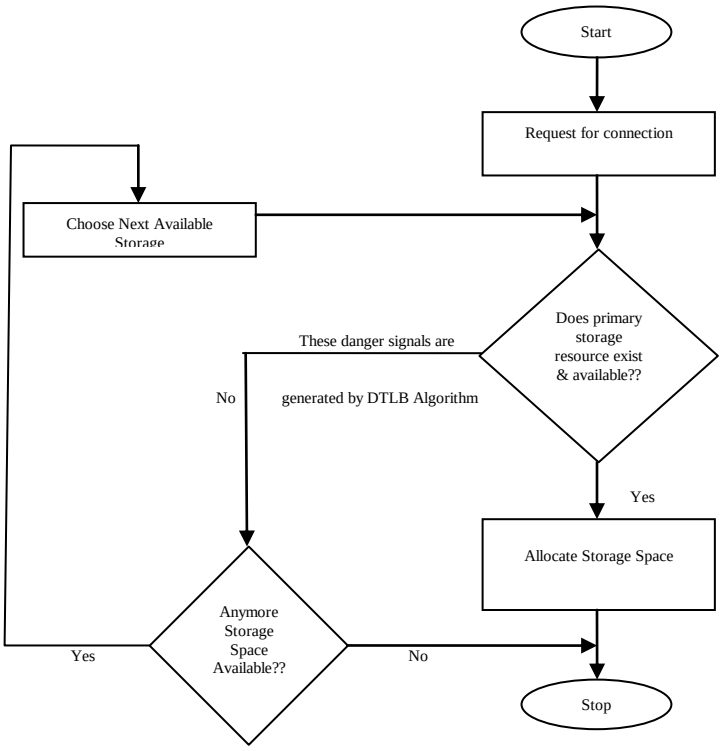


Fig. 1 Proposed Model

DTLB Algorithm:

Step1: Start .

Step2: Client request for storing the data.

Step3: Loop until primary storage resources are exists and available.

Step4: Allocate storage space and go to the step 9.

Step5: If primary storage resource not exists, it is not available or cross threshold value then Danger Signal generated by Danger Theory algorithm then go to step 6.

Step6: Loop until anymore storage space available.

Step7: If storage space available then choose next storage And go to step 3.

Step8: If storage space not available then go to step 9.

Step9: Stop.

DTLB find the expected response time and generating a danger signal with the help of danger theory algorithm when server is overloaded or not available, the expected response time can be found with the help of the following formulas:

With the help of following formula calculate the response time of algorithm [11].

Response Time=Finish time– Arrival time + Transmission delay ….. ….. (1)

Where, Arrival time is the arrival time of user request and Finish time is the finish time of user request and the transmission delay can be determined using the following formulas.

$$\text{Transmission delay} = T_{latency} + T_{transfer} \ldots\ldots\ldots\ldots (2)$$

Where, $T_{latency}$ is the network latency and $T_{transfer}$ is the time taken to transfer the size of data of a single request (D) from source location to destination.

$$T_{transfer} = \text{single request} / Bw_{\text{per user}} \ldots\ldots\ldots\ldots\ldots (3)$$
$$Bw_{peruser} = Bw_{total} / N_r \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Where, $Bw_{peruser}$ is band width per user, Bw_{total} is the total available bandwidth and Nr is the number of user requests currently in transmission.

4. Result Analysis

Results are analysis thoroughly by linux data center operating system, service proximity based routing, five regions, one data center which is located in 0 region, five user base, five VM's, five physical H/w units, 1024 Mb storage, 512 Mb memory and 1000000 Mb bandwidth for each device and five processors are input parameters. Simulation time is set at 60 sec. DTLB response time and data center processing time are calculated by its formula.

Experiment is performed on DTLB algorithm with one data center and five virtual machines. Five physical host and server proximity based routing policy is used for simulation. Table (1-2) shows the parameter values. Each physical host has 5 numbers of VM's and unique id for load balancer. These ids are used by balancer to find out the availability of VM's. In all the experiment five processors and time shared VM policy is used, while simulation time is set at 60 sec.

Id	Memory (Mb)	Storage (Mb)	Available BW (Mb)
0	512	1024	1000000
1	512	1024	1000000
2	512	1024	1000000
3	512	1024	1000000
4	512	1024	1000000

Table 1: Parameter Values

No. Of processors	Processor Speed (MIPS)	VM Policy
1	10000	TIME-SHARED
2	20000	TIME-SHARED
3	30000	TIME-SHARED
4	40000	TIME-SHARED
5	50000	TIME-SHARED

Table 2: Parameter Values

The above parameters are set with the values indicated in table 1, 2. Primary and secondary output s are determined based on these parameters.

PRIMARY OUTPUT:

i). Response time of the simulated application.
ii). Overall average, minimum and maximum response time of all user requests simulated in tabular format.
iii). How many users use the application at what time from different regions of the world, and the overall result of that usage on the data centers hosting the application?

SECONDARY OUTPUT:

iv). The time taken by data center to service a user request.
v). The overall request processing time for the entire simulation.
vi). The average, minimum and maximum request processing time by data center.

4.1 DTLB Overall Response Time Results

This experiment is carried with the intention to determine response time for DTLB when response time is varied from 50.062 milliseconds to 498.888 milliseconds. Primary inputs such as VM's, hosts, storage, memory, bandwidth are kept constant and against these constant parameters response time of DTLB is calculated. Proximity based routing policy is used for this experiment. Output results include Minimum, Average and Maximum response time along with overall response time and data center processing time. Table 3 shows the DTLB over all response time.

Response Time by Region	
User bases	**Average Response Time (ms)**
User Base-1	50.062(ms)
User Base-2	199.614(ms)
User Base-3	299.48(ms)
User Base-4	500.349(ms)
User Base-5	498.888(ms)
Overall Response Time	**310.05(ms) or 310050 microsecond**

Table 3: DTLB Response Time Result

Response time can be measured as the time interval between sending a request and receiving its response. Average response time of each user base is calculated by response time formula. User base 1 has 50.062 milliseconds, user base 2 has 199.614 milliseconds, user base 3 has 299.48 milliseconds, user base 4 has 500.349 and user base 5 has 498.888 millisecond average response time. The overall response time of DTLB is 310.05 millisecond or 310050 microseconds.

4.2 Comparative Results for DTLB/Round Robin/Throttled Load Balancing Algorithm

In proposed work DTLB algorithm compare with existing Round Robin (RR) and Throttled load balancing algorithms. In RR algorithm datacenter controller assigns the requests to a list of VMs on a rotating basis. First request is allocated to a VM- picked randomly from the group and then the datacenter controller assigns the subsequent requests in a circular order. Once the VM is

assigned the request, the VM is moved to the end of the list. Simulation is performed on same DTLB parameters for RR algorithm. Its response time is 310.12 ms (millisecond) and data center processing time is 0.285ms.

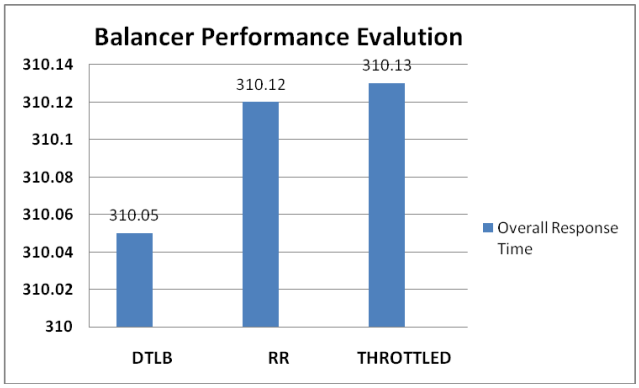
Throttled algorithm maintains record of the state of each virtual machine (busy/idle). If a request arrived concerning the allocation of virtual machine, the algorithm balancer sends the ID of ideal virtual machine to the data center controller and data center controller allocates the ideal virtual machine. Simulation is performed on same DTLB parameters for throttled algorithm. Its response time is 310.13 ms (millisecond) and data center processing time 0.250ms. Table 4 shows the comparative results. DTLB provides better storage utilization, performance and three level danger signals.

S. No	Cloud Application Service Broker Policy	Danger Theory based Load Balancing policy (Proposed DTLB)	Round Robin(RR) (Existing)	Throttled (Existing)
1.	Overall Response Time	310.05 (millisecond)	310.12 (millisecond)	310.13 (millisecond)

Table 4: Performance Effectiveness: DTLB/RR/Throttled load balancing policy Vs Service proximity based routing

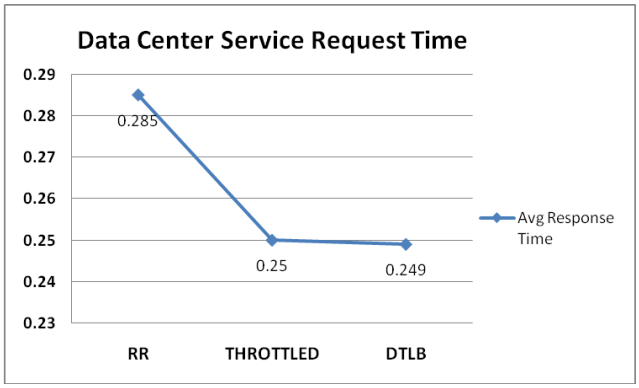
DTLB, RR and Throttled over all response time are compared with respect to the service broker policy 'Service Proximity Based Routing' as condition with constant values of datacenter, virtual machine and simulation configuration. DTLB shows optimal performance ie.310050 microseconds as against RR-310120 and Throttled -310130 microseconds with danger signal for cloud application.

Graph 1 represents the response time comparison of DTLB, RR and Throttled algorithms. DTLB provides minimum response time i.e. 310.05 millisecond as compare to RR and Throttled.



Graph 1: Response time comparison

Y –axis represent over all response time in milliseconds and X- axis represent DTLB, RR and Throttled algorithms. Graph shows the balancer performance with server proximity based routing policy.



Graph 2: Data center request processing time comparison

Time taken by data centers to service a user request is called data center request processing time. Graph 2 shows the data center request processing time. Y-axis represent data center processing time in millisecond and X-axis represent DTLB, RR and Throttled algorithms. DTLB provides minimum data center request processing time i.e. 0.249 millisecond as compare to RR and Throttled.

5. Conclusions

This paper presents "Danger Theory Based Load balancing" (DTLB) algorithm in cloud computing environment. Proposed algorithm implement a new system inspired by Danger Theory Algorithm to overcome existing limitations of load balancing in cloud computing. It generate danger signals to identify vulnerabilities in order to provide reliability and trust. Results provide less response time, less data center processing time, increase storage utilization, reliability and

performance. For future scope various soft computing self learning techniques can be including into this system. So, best part of self learning i.e. learn from previous experience can be acquire into the system.

References

[1] 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.

[2] Nayandeep Sran and Navdeep kaur "Comparative Analysis of Existing Load balancing techniques in cloud computing", International Journal of Engineering Science Invention, Vol-2 Issue-1 2013 .

[3] B. Radojevic and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments," in MIPRO, 2011 Proceedings of the 34th International Convention. IEEE, 2011, pp. 416–420.

[4] Z. Sui and S. Pallickara, "A survey of load balancing techniques for data intensive computing," in Handbook of Data Intensive Computing, B. Furht and A. Escalante, Eds. Springer New York, 2011, pp. 157–168.

[5] Hai-Dong Fu and Gui-Feng Li" Three-level Anomaly Disposal System Model Based on Danger Theory " 978-1-4244-2108-4/08/$25.00© 2008 IEEE.

[6] H. Wen, Z. Hai-ying, L. Chuang, and Y. Yang, "Effective load balancing for cloud-based multimedia system," in Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, vol. 1. IEEE, 2011, pp. 165–168.

[7] Sandeep Sharma, Sarabjeet Singh and Meenaksshi Sharma, "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology, 2008.

[8] Saroj Hiranwal and Dr. K.C. Roy, "Adaptive Round Robin Scheduling Using Shortest Burst Approach Based On Smart Time Slice" International Journal Of Computer Science And Communication July-December 2011 ,Vol. 2, No. 2 , Pp. 319-323

[9] R.Yamini "Power Management in Cloud Computing Using Green Algorithm", IEEE-International Conference On Advances In Engineering, Science And Management, March 2012.

[10] Reza Azmi "Secure hypervisor based anomaly detection using danger theory " Computer & Security Elsevier journal ,19 july 2013 .

[11]Jasmin james and Dr. Bhupendra verma "efficient vm load balancing algorithm for a cloud computing environment " international journal on computer science and engineering (ijcse) ISSN : 0975-3397 Vol. 4 No. 09 Sep 2012.

First Author
M/s. Isha Dubey, MCA, R.G.T.U., Bhopal, M.P., India. Presently pursuing dissertation of M.Tech Software Systems, under the supervision of Prof. Praneet Saurabh, Technocrats Institute of Technology, Bhopal (M.P.) India.

Second Author

Prof. Praneet Saurabh, M.Tech in Computer Science and Engineering .Presently working as Asst. Professor in Computer Science & Engineering Department, Technocrats Institute of Technology, Bhopal(M.P.) India.