REQUEST ROUTING MECHANISM FOR SIMILARITY SERVICE BASED CLUSTER WEB SERVERS

T.N. Anitha[1] , Dr. Balakrishna .R[2]

[1]Research Scholar, Rajarajeswari College of Engineering, Bangalore (India)
[2]Professor & HOD Dept of ISE, Rajarajeswari College of Engineering, Bangalore (India)

ABSTRACT

Because of the busy work schedule most of the people are not getting the time to do simple personal works like phone recharging, paying bills, shopping etc. So they are depending on the internet services. These days a huge number of services are available on internet. Only the thing that if the services are popular then hit rate on that service increases that causes declining quality of service on internet. To improve the quality of services, we are proposing a content based request routing and load balancing.

Keywords: Clusters ,Web Servers, Service Wise Request Routing

I INTRODUCTION

We are all huge consumers of various services the internet supports. Web Servers have gained an immense popularity these days due to its nature to cater huge number of user requests were any number of users can get service from web based applications. As it comes with global accessibility, servers hosting popular website tends to get massive load which deteriorates its efficiency to provide quality service[11]. Imagine a situation where you are accessing your bank account to transfer funds to a friend's account for an emergency. And the site simply says, ' Sorry, the service is unavailable!'. As end users, we would hate to be in this situation. That you get to call the customer service. This despite having invested in a cluster for a reliable service!

Most businesses happen over the Internet and Enterprises prefer clusters for the reliable services. Critical servers and applications such as database servers, exchange servers etc are hosted in clustered environment for high availability. Further, load balancers are plugged into the network to distribute the load on servers to address the primary need of un-interrupted availability at all times[8]. There are three essential principles driving cluster & load balancing today:

- Send data as efficiently as possible
- Send data as infrequently as possible
- Send as little data as possible

Clustering allows us to run an application on several parallel servers . The load is distributed across different servers, and even if any of the servers fails, the application is still accessible via other cluster nodes. Clustering is crucial for scalable enterprise applications, as you can improve performance by simply adding more nodes to the cluster. Clustering solutions usually provide Scalability, High Availability & Load Balancing.

Cluster has two types of Failovers: Firstly Request Level Failover in which If one of the servers in the cluster goes down, all requests should get redirected to the remaining servers in the cluster. Secondly Session Level Failover in which if one of the servers in the cluster goes down, then some other server in the cluster should be able to carry on with the sessions that were being handled by it, with minimal loss of continuity.

The objective of load balancing that it should distribute the load among the servers in the cluster to provide the best possible response time to the end user. Here we are focusing to perform load balancing based on the content type of service request.

The paper includes following sections of information: Section II provides Related work, Section III: provides Request Routing section IV Experimental Results & Section V conclusion

II RELATED WORK

While content-aware distribution getting more popular in cluster-based web systems. Most of the researchers using different techniques on clusters to improve the service accessibility. Zhiyong Xu et al[1], propose a novel and efficient content-aware dispatching algorithm. Their approach eliminates the potential bottleneck and the single point of failure problems completely by using totally decentralized P2P architecture. It is scalable, increases throughput linearly with the increased number of servers and it does not introduce heavy communication overhead among back-end servers.

- DU Zeng-Kui et.al[2] they proposed a architecture with dispatching policy named DWARD to achieve scalable server performance. With DWARD, all the server nodes participate in request dispatching on the basis of local access pattern. They developed this policy using HASH function on homogeneous servers . They implemented the test on linux kernel and compare the results with WARD and LARD algorithms. The performance results shows that this method achieves good throughput.
- YunFeng Li, et al[3], considering heterogeneity server performance the authors propose a novel scheduling algorithm for Web switch operating at layer-7 of the OSI protocol stack. A scheduling policy classifies the requests according to their contents on Web servers and estimate the load state of each server. Using a L7 scheduling algorithm,allow the switcher to select randomly two servers from the server pool and calculating the load of the two servers, the request will be dispatched to the lowly loaded server. This method improves the performance but there is a difficulty in classifying the client requests.
- Chun-Wei Tseng,et.al,[4] the authors propose a mechanism that can effectively shield off the effect of code 304 in the server cluster. these authors used two-level dispatching scheme to cooperate for distributing requests. In this method they place a *workload divider* in front of the two clusters to map different clients to different distributors.Implement the algorithms such that it evenly divide the workload to the two clusters. Workload divider can preserve the possible reference locality in the user session because they divide workload on a per client basis. But the time consumption to carry out this method is more.

III SERVICE WISE REQUEST ROUTING

An explosion in the number and variety of devices is dramatically changing the world of internet computing. Recently With the increasing availability of the internet on a wide range of these devices the web is emerging as the proffered data delivery mechanism for a number of application scenarios.

There is a rapid movement towards the use of powerful processors. These processors have constraints on the available Processing Power, Network and Storage. Such constraints affect the quality of services on web for the end users. Accessing the services through the cellular network is limited upto 19.2 KB/Secs. The processing capacity of the system varies based on their configurations and memory capacity. These constraints failure to provide effective service expectations to the user.

Cluster based web server system becomes one of the most prevailing mechanisms to satisfy this need. As shown in the figure below

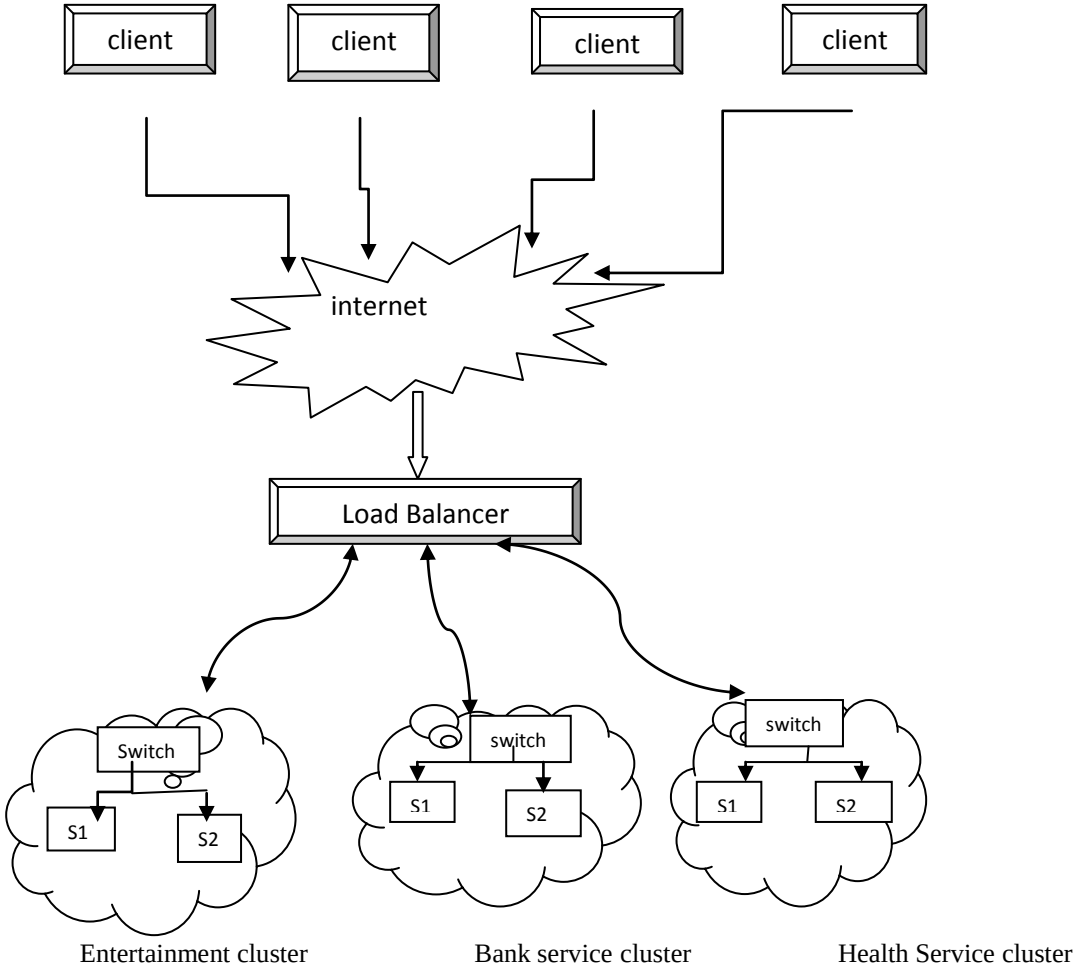

Fig 1: Proposed Cluster Architecture

The cluster are formed based on their services using a mathematical model based on Random walk algorithms concepts. Distribution of the users requests over a group of servers is a strategy to achieve scalability . A HTTP is a stateless protocol, So each request may be directed to any host in a distributed server cluster. This may creates un accessibility of service from the servers.

So we are proposing the service based request dispatching method using mapping table . The load balancer mapping table contains the IP address of the cluster heads , type of service and load state of a cluster . The IP address of the Cluster Head is given by DNS server connected to an Load balancer.

Load of each server LSi= Cpu load +mem load +bandwidth usage

Average Load on cluster $L_{avg} = \sum_{i=1}^{n} LSi / M$ where M= number of servers present in an cluster. we are consider two load states

Ln < Underload*Lavg= underload

Ln > Upperload*Lavg= overload

Load on the servers are identified using following algorithm

```
if(matching(request type)=True) then
{
For (j=1; j<m; j++)
For( i=1; i<n;  i++)
If ( cj.si= underloaded)
Allocate the request to si
Elseif(c.s1 =  overloaded)
Search another server in cluster j
}
If( all servers in cj are overloaded) then
Find another nearest cluster providing the same type of service
Allocate request to that cluster
}
```

The request can be dispatched in an two methods :Non Content approach and Content-aware approach.

In an non content based method , the load balancer randomly dispatches the request to one of the cluster server but in case of content based method the requests are dispatched based on request category. We are using content aware request serving concept.

IV RESULT

To carry out the proposed work we setup a Hardware and Software configurations as stated below.

Hardware Configuration:

We formed cluster with minimum two Pentium processors ranging from 1GHZ to 2 GHZ with minimum of 512Mb &DDR RAM. Also used different machines with variable processor speeds are as clients.

Software Configuration:

All the system in the cluster runs windows xp 2007 operating system; we developed experimental test on our proposed model using Jmeter. Jmeter generates load on cluster servers. We vary the loads on the cluster by varying the number of concurrent clients. The performance of our work is measured using response time , latency and throughput calculations. We compare our results with non content based concepts.

The following figure 2 indicates the response time of content based and non content based, in which content based method reduces the response time by 7.9 msecs compare to non content based method.

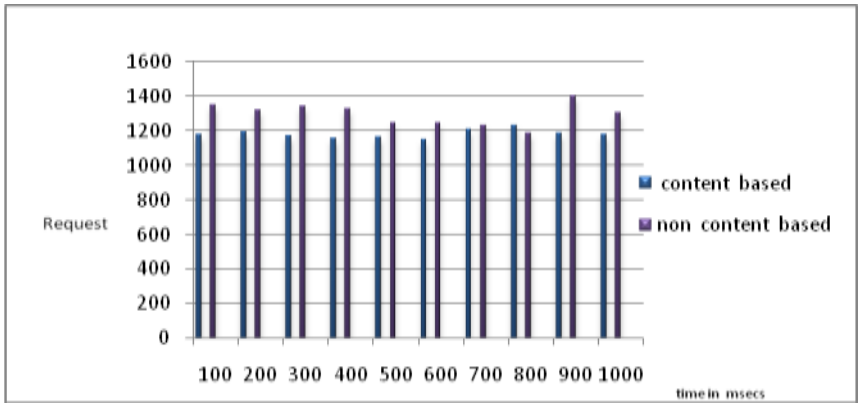

Fig2: Response Time Comparison

Below figure 3 shows latency comparison where it is shown that the non content based system servers serves only less number of requests per minute compare to the content based system. From this we may say that 12.4% of improvement from content based .

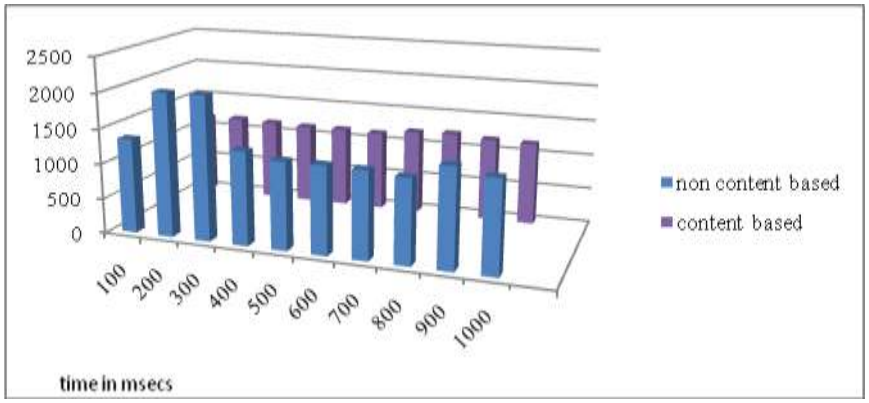

Fig3: Latency Comparison

The following table 1 provides comparative values obtained between the content and non content based methods

Table 1: comparisons between content & non content request routing

Parameters	Content Based	Non Content Based
Number of Requests	9461	9461
Response time	1117ms	1186ms
Latency	1172ms	1296ms
Throughput	3.9ms	2.3ms
utilization	1.30ms	1.04ms

V CONCLUSION

Web technology which has a very vast potential and is still unexplored. The capabilities of web computing are endless. One of the major issue of web service is load balancing because overloading of a system may lead to poor performance which can make the technology unsuccessful. The increase in web traffic and different services are increasing day by day making load balancing a big research topic. So there is always a requirement of efficient load balancing algorithm for efficient utilization of resources.

In our work the workload is distributed among various other nodes of clustered servers based on content of a request , it improve both resource utilization and job response time. Experimental results suggest that our proposed algorithm can balance the load of web server clusters effectively, make full use of the existing source of software and hardware, highly improve the server's performance, and even make the best use of the web server.

REFERENCES

[1] Zhiyong Xu, Jizhong Han published a "Scalable and Decentralized Content-Aware Dispatching in Web Clusters"paper in IEEE 2007.

[2] *DU* Zeng-Kui, *JU* Jiu-Bin published a "Distributed Content-aware Request Distribution in Cluster-based Web Servers " paper in IEEE 2003

[3] YunFeng Li, QingSheng Zhu, YuKun Cao published a paper "A Request Dispatching Policy for Web Server Cluster" in IEEE 2002.

[4] Chun-Wei Tseng, Rong-Ching W et.al "Revisiting Effectiveness of Content-Aware Switching for Web Traffic Distribution"presented in *6th IEEE Conference on Industrial Electronics and Applications 2011.*

[5] G. Teodoro T. Tavares B. Coutinho W. Meira Jr. D. Guedes "Load Balancing on Stateful Clustered Web Servers published in IEEE 2002.

[6] shanthi swaroop et, " analysis of load balancing in cloud computing" 2013, IJCSE

[7] Alok jain, High availability setup and Performance improvement for RRCAT Information Portal using Server Load Balancing published in International Conference on Cloud, Big Data and Trust 2013, Nov 13-15, RGPV

[8] Chumei Design &implementation of a load balancing model for web server cluster system 2012 IEEE

[9] Ankush P. Deshmukh,. Kumarswamy Pamu published Applying Load Balancing: A Dynamic Approach in international journal of advanced Research in Computer science and Software Engineering vol 2 issue 6, june 2012.

[10] Shaojun & j.sha published a paper on " analysis and algorithm of load balancing strategy of the web server cluster system" in ICCIP springer in 2012

[11] Kutja gilly, carlos juit & ramon pulgjaner published paper "An up –to - date survey in web load balancing" in springer – 2011 pp 699-706.

[12] J¨org D¨ummler,Thomas Rauber,Gudula R¨unger Semi-dynamic Scheduling of Parallel Tasks for Heterogeneous Clusters J¨org 2011 10th International Symposium on Parallel and Distributed Computing.