

Client Based Adaptive Load Balance In Web Server

Venkatesan.B,
Assistant Professor/IT,
Paavai engineering college,
Nammakal, India.
balavenkatbe@gmail.com

Aishwarya.L,
Department of IT,
Paavai engineering college,
Nammakal, India.
ishujoyal96@gmail.com

Kasthuri.S,
Department of IT,
Paavai Engineering College,
Namakkal, India.
kasthuririthik908@gmail.com

Abstract— The growth of internet services during the past few years has the demand for scalable distributed computing systems. Ecommerce systems concurrently serve many clients that transmit a large number of request. An increasingly popular and cost effective technique to improve server performance is load balancing where hardware and /or software mechanisms decide which t will execute the client request .Load balancing mechanisms distribute client workload among server nodes to improve overall systems responsiveness load balances has emerged as a powerful new technology to solve our paper focuses on a new generation of adaptive /intelligent dynamic load balancing technique with based on J2EE technology and JMX (java monitoring extension) is used to monitor the performance and status of the server. The effectiveness of the new balancing method will be demonstrated through exact measurement results compared with former traditional non adaptive methods

keywords; Distributed systems, adaptive load balancing ,J2EE application server, JMX.

I. INTRODUCTION

As the number of concurrent requests is increased on a standalone server, so the application exceeds the pre-estimated respond time, because the work load is too much on the server machine. At this time, there are two options to solve this problem: using faster machines or using multiple machines parallel. The first solution can be expensive and limited by the speed of a standalone machine. The second choice is more straightforward: deploy the same application on several machines and redirect client requests to those machines.

The system is transparent from outside, which means that client applications perceive a standalone very-server with one accessible IP address. To achieve the performance and transparency, load balancing algorithms must be utilized. Load balancing can improve system performance by providing better utilization of all resources in the whole system, which consists of computers connected by local area networks. The main objective of load balancing is to reduce the mean response time of requests by distributing the workload. Basically the dynamic load equalization is tough to simulate however best suited in heterogeneous atmosphere. Load equalization is required to issues the surplus dynamic work fairly to entire node within the whole cloud to realize the high resource utilization. Load equalization could also be static or dynamic. In static load equalization all data

concerning equalization selections are celebrated earlier and ignore the present state of the system. However dynamic load equalization algorithms gather state data and react to system state if it modified. Migration time is outlined because the total time needed in leave the resources from one node to a different and fault tolerance allows the rule to continue operative accurately within the event of some failure. Migration is the rule capable to decide to create a decision to come to a arrangement to make your mind up to choose to determine that it ought to make changes of load distribution throughout beheading of method or not. The migration time is taken for the analysis of the performance of each algorithm [3]. The OSI model was developed as a framework for developing protocols and applications that could interact seamlessly. The OSI model consists of seven layers and is referred to as the 7-Layer Networking Model [2]. Each layer represents a separate abstraction layer and interacts only with its adjoining layers. Load balancing mechanism can be realized on the Layer 3 - 7. OSI levels 3 and 4 can be supported balancing mechanisms via network router devices. On layers 5 and 7, 'URL Load Balancing' can be achieved. A lively example of 'URL Load Balancing' can be the following: the URL may be static (such as <http://www.xxx.net/home>) or may be a cookie embedded into a user session. An example of URL load balancing is directing traffic to <http://www.xxx.net/documents> through one group of servers, while sending <http://www.xxx.net/images> to another group. URL load balancing can also set persistence based on the "cookie" negotiated between the client and the server.

II. RELATED WORKS

In general, It provides many services to the clients to operate different tasks like keeping, organizing, sharing, searching the multimedia content like images, animations, audio, video, etc. It also supports different devices like laptops, tablets, smart phones etc for performing these operations. With lots of innovation in communications and technologies, some of the vendors are also delivering advanced features like camera-to-cloud to keep the live collecting data into cloud directly, multi-user concurrent operations (for team collaboration), live streaming, single & multi clients 3D graphical gaming etc. There is a lot of variation in utilization of server resources based on type of multimedia assignment for e.g., server need more CPU, RAM and storage space for processing gaming or videos when analyze to images or web pages. The earlier works which are on time based are won't be considerable for this research work.

III. MATERIALS AND METHODS

Network-based load balancing-

This type of load balancing is provided by network router devices and domain name servers (DNS) that service a cluster of host machines. For example, when a client resolves a hostname, the DNS can assign a different IP address to each request dynamically based on current load conditions. The client then contacts the designated server. Next time a different server could be selected for its next DNS resolution. Routers can also be used to bind a TCP flow to any back-end server based on the current load conditions and then use that binding for the duration of the flow. High volume Web sites often use network-based load balancing at the network layer (layer 3) and transport layer (layer 4). Layer 3 and 4 load balancing (referred to as “switching” [1]), use the IP address/hostname and port, respectively, to determine where to forward packets. Load balancing at these layers is limited, however, by the fact that they do not take into account the content of client requests. Higher-layer mechanisms – such as the so-called layer 5 switching described above – perform load balancing in accordance with the content of requests, such as pathname information within a URL.

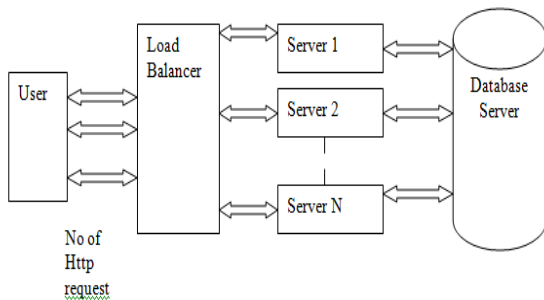


Figure 1 Load balancing Architecture

V. FUNCTIONS OF LOAD BALANCE MODEL

A. Operating System - based load balancing

This type of load balancing is provided by distributed operating systems via clustering, load sharing, or process migration mechanisms. For instance Microsoft provides a new clustering possibility: Microsoft Cluster Server (MSCS) This special Microsoft software provides services such as failure detection, recovery, and the ability to manage the servers as a single system. Clustering is a cost effective way to achieve high-availability and high-performance by combining many commodity computers to improve overall system processing power.

B. Middleware-based load balancing

This type of load balancing is performed in middleware products, often on a per-session or per-request basis. For example, layer 5 switching has become a popular technique to determine which Web server should receive a client request for a particular URL. This strategy also allows the detection of “hot spots,” i.e., frequently accessed URLs, so that additional resources can be allocated to handle the large number of requests for such URLs. It can also be applied on top of consumer level (COTS) networks and operating systems, which helps reduce cost. In addition, middleware-based load balancing can provide essential to use an adaptive load balancing algorithm, which tries to distribute tasks in an intelligent way using

load information of the nodes formed between shoppers and server clusters.

IV. TECHNOLOGIES

JAVA:

Introduction of Java

An Object Oriented Programming Language developed at Sun Microsystems. A Virtual Machine (run-time environment) that can be embedded in web browsers (such as Netscape Navigator, Microsoft Internet Explorer) and operating systems. A standardized set of Class Libraries (packages), that support:

- creating graphical user interfaces
- controlling multimedia data
- communicating over networks

Advantages of java:

- Simple

Java is simpler to learn for programmers if they think in terms of objects and methods. Java has eliminated the complexities present in C++. Java supports garbage collection to further simplify the language as it takes the burden of the memory management off the programmer. Java does not use header files and it eliminates the C processor. Constructs like struct and union have been removed. Java also the operator overloading and multiple inheritance features of other object oriented languages. Perhaps the most important simplification, however is that java does not use pointers. Java automatically handles the referencing and de-referencing of objects for you. Thus it frees us from having to worry about dangling pointers, invalid pointer references and memory leaks.

- Object-Oriented

Java is an object-oriented language. It mainly focuses on the data in the application and methods that manipulate the data, rather than thinking strictly in terms of procedures. Java comes with an extensive set of classes, arranged, that you can use in your programs.

- Distributed

Java was built with network communication in mind. It had a comprehensive library of routines for dealing with network protocols such as TCP/IP, HTTP, and FTP. Hence it is easy to read a remote file or resources, as it is to read a local file. Java applications are open and can be accessed across the Internet.

- Interpreted

Java is an interpreted language. The Java compiler generates byte-code for the JVM (Java Virtual Machine). This Java byte-code is platform independent and the programs can be run on any platform the JVM has been ported to.

- Robust

Java has been designed for writing reliable or robust software. It is a strongly typed language, which allows for extensive compile time checking for potential type mismatch problems. The lack of pointers and pointer arithmetic features increases the robustness of Java programs by abolishing the pointer related bugs. Exception handling is another feature in Java that makes for more robust programs.

- Secure

Security is a critical part of the Java environment. Java allows us to create virus-free programs and prevent malicious code security, which can be achieved by a digital signature to Java code. The origin of the code can be established in a cryptographically secure and unforgettable way.

- Java Virtual Machine

Java is compiled to byte-codes whose target architecture is the Java Virtual Machine (JVM). The virtual machine is embeddable within other environments, e.g., web browsers. Utilizes a byte-code verifier when reading in byte-codes. The Class Loader is employed for “classes” loaded over the network.

- Struts

Apache Struts is a free open-source framework for creating Java web applications.

Web applications differ from conventional websites in that web applications can create a dynamic response. Many websites deliver only static pages. A web application can interact with databases and business logic engines to customize a response.

Web applications based on JavaServer Pages sometimes commingle database code, page design code, and control flow code. In practice, we find that unless these concerns are separated, larger applications become difficult to maintain.

One way to separate concerns in a software application is to use a Model-View-Controller (MVC) architecture. The Model represents the business or database code, the View represents the page design code, and the Controller represents the navigational code. The Struts framework is designed to help developers create web applications that utilize a MVC architecture.

The framework provides two key components:

- A "request" handler provided by the application developer that is mapped to a standard URI.
- A "response" handler that transfers control to another resource which completes the response.
- MySQL

MySQL is popular for web applications and acts as the database component of the LAMP, MAMP, and WAMP platforms

(Linux/Mac/Windows-Apache-MySQL-PHP/Perl/Python), and for open-source bug tracking tools like Bugzilla. Its popularity for use with web applications is closely tied to the popularity of PHP and Ruby on Rails, which are often combined with MySQL. PHP and MySQL are essential components for running popular content management systems such as Joomla!, e107, WordPress, Drupal, and some BitTorrent trackers. Wikipedia runs on MediaWiki software, which is written in PHP and uses a MySQL database.

The following features are implemented by MySQL but not by some other RDBMS software:

Multiple storage engines, allowing you to choose the one which is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time):

Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, Cluster, BDB, EXAMPLE), and Maria

Partner-developed storage engines (InnoDB, solidDB, NitroEDB, BrightHouse)

Community-developed storage engines (memcached, httpd, PBXT)

Custom storage engines

Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

- JFreeChart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

a consistent and well-documented API, supporting a wide range of chart types;

a flexible design that is easy to extend, and targets both server-side and client-side applications;

support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

- Apache Tomcat Server

Apache Tomcat is a web container, or application server developed at the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Tomcat includes its own internal HTTP server.

- Apache Tomcat Architecture

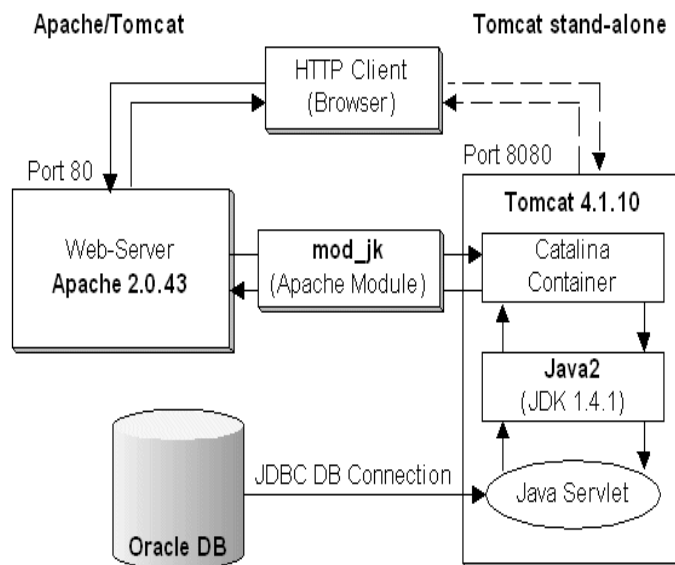


Figure 3 Apache Tomcat Architecture

- Server

In the Tomcat world, a Server represents the whole container.

Tomcat provides a default implementation of the Server interface., and this is rarely customized by users.

- Service

A Service is an intermediate component which lives inside a Server and ties one or more Connectors to exactly one Engine. The Service element is rarely customized by users, as the default implementation is simple and sufficient: Service interface.

- Engine

An Engine represents request processing pipeline for a specific Service. As a Service may have multiple Connectors, the Engine received and processes all requests from these connectors, handing the response back to the appropriate connector for transmission to the client. The Engine interface may be implemented to supply custom Engines, though this is uncommon.

- Host

A Host is an association of a network name, e.g. www.yourcompany.com, to the Tomcat server. An Engine may contain multiple hosts, and the Host element also supports network aliases such as yourcompany.com and abc.yourcompany.com. Users rarely create custom Hosts because the StandardHost implementation provides significant additional functionality.

- Connector

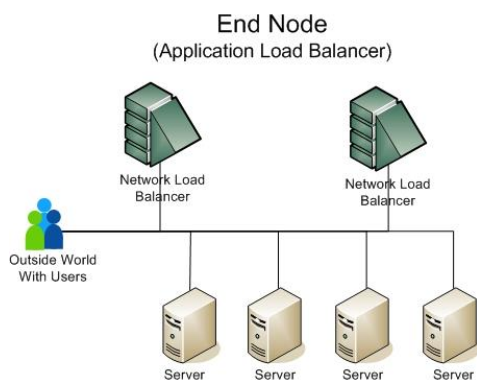
A Connector handles communications with the client. There are multiple connectors available with Tomcat, all of which implement the Connector interface. These include the Coyote connector which is used for most HTTP traffic, especially when running Tomcat as a standalone server, and the JK2 connector which implements the AJP protocol used when connecting Tomcat to an Apache HTTPD server. Creating a customized connector is a significant effort.

- Context

A Context represents a web application. A Host may contain multiple contexts, each with a unique path. The Context interface may be implemented to create custom Contexts, but this is rarely the case because the StandardContext provides significant additional functionality.

- Network Load Balancing End-node Configuration Pattern

Load balancing technology is used to balance workload across servers to improve availability, performance, and scalability. Network Load Balancers are implemented at the workgroup/server switch layer. Load balancing increases performance consistency and application availability and are therefore recommended for NIH enterprise applications.



A one-to-one or one-to-many mapping can be used to access a specific server or a group of servers respectively. Additionally, it offers multiple algorithms for mapping user requests to servers (e.g., round-robin, random, or depending on server utilization) and provides proxy services. The End-node Configuration does not provide NAT.

Therefore the load-balanced servers can access other resources in the network directly without having to utilize the load balancers' proxy services; this facilitates access to backup and other services. When deployed singly, a load balancer can improve performance by efficiently allocating workload across multiple servers. In order to deliver improved availability, the load balancers must be deployed in pairs, with hot standby configured. Otherwise, the load balancer can become a single point of failure for the servers.

- JMX

JMX Module helps the load balancer to connect the J2EE web servers. JMX Module uses Java Management Extensions technology and can connect to any JVM (Java Virtual machine) to get the information. JMX Module contacts the Mbean server on the remote JVMs and gets the information like thread count, load average.

- Modules

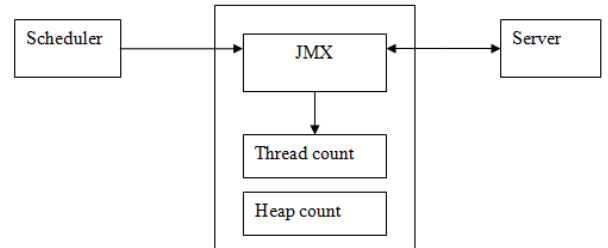


Figure 4 Module

- Web user interface module

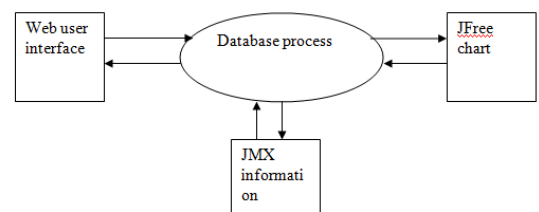
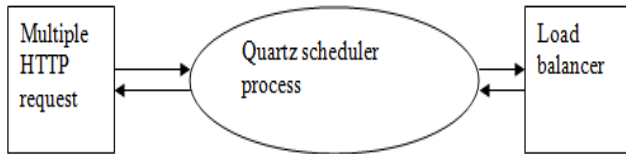


Figure 5 Web User Module

This module provides the user the option to make configurable changes and start the http Technology used is Struts which based is on MVC pattern. Web user interface module helps the user to



view/delete the web server details

- Sample Usage – Scheduler

Figure 6 Scheduler Module

As a new order is initially placed, schedule a Job to fire in exactly 2 hours, that will check the status of that order, and trigger a warning notification if an order confirmation message has not yet been received for the order, as well as changing the order's status to 'awaiting intervention'.

- Scheduler Module

Quartz Scheduler library is used to incorporate scheduling framework

- Core Load balancer Module

The load balancer module forms the core part of the entire system Load balancer system does the following:

Gets the Http request from the browser Starts the scheduler and schedules the monitoring job Monitor job continuously monitors the servers behind load balancer and collects statistical info

- JMX Module

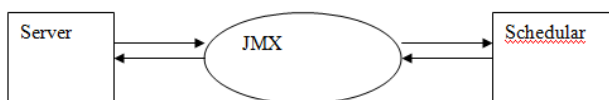
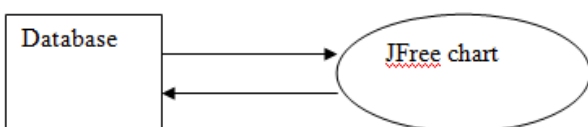


Figure 7 JMX Module

JMX Module helps the load balancer to connect the J2EE web servers JMX Module uses Java Management Extension technology and can connect to any JVM (Java Virtual machine) to get the information JMX Module contacts the Mbean server



on the remote JVMs and gets the information like thread count, load average

- Web based Admin Module

The web based Admin module is developed using Struts 2framework The admin module allows the user to configure and manage the load balancer Admin can be made to trigger series of Http requests to the load balancer

- Database Module

Figure 8 Database Module

The database used in the project is Mysql.MySQL is an open source relational database management system The database is needed for the project to allow the load balancer to load config files and insert JMX results

- Web based Admin Module

The web based Admin module is developed using Struts 2 framework The admin module allows the user to configure and manage the load balancer Admin can be made to trigger series of Http requests to the load balancer

- Requirements

- Load Balancer must be able to use multicast
- Load Balancer needs to be able to reach all cluster nodes in order to distribute/redirect requests
- Load Balancer needs to be able to communicate with clients over the specified address/port
- Load Balancer needs to be able to receive data on multicast 230.0.0.2:27512 to hear when cluster nodes are available
- Any cluster node needs to be able to send data on multicast 230.0.0.2:27512 to inform the Load Balancer of their availability
- Any cluster node needs to be configured with a cluster-island value greater than 0 in order to inform the Load Balancer of their availability.

VI. CONCLUSION

This project provides an insight into load balancing on Web application server clusters. The major achievement in the proposed system is the optimization of system performance. The load balancing system results in higher availability and scalability necessities in an enterprise, Web-based application.

REFERENCES

- [1] J. LINDFORS, M. FLEURY, THE JBOSS GROUP: JMX: Managing J2EE with Java Management Extensions. SAMS Publishing Inc., 2002.
- [2] J. BASNEY AND M. LIVNY: "Deploying a High Throughput Computing Cluster," High Performance Cluster Computing, vol. 1, May 1999.
- [3] C. O'RYAN, F. KUHN, D. C. SCHMIDT, O. OTHMAN, AND J. PARSONS: "The Design and Performance of a Pluggable Protocols Framework for Real-time Distributed Object Computing Middleware", in Proceedings of the Middleware 2000 Conference, ACM/IFIP, Apr. 2000.

- [4] D. SCHMIDT, M. STAL, H. ROHNERT, AND F. BUSCHMANN: *Pattern-Oriented SoftwareArchitecture: Patterns for Concurrent and Networked Objects*. Wiley, 2000.
- [5] L.M.CABRERA: "The influence of workload on load balancing strategies", in Proc.Summer USENIX Conf., pp. 446-458, June 1986.
- [6] W.LELAND AND T.OTT: "Load balancing heuristics and process behavior", in Proc. ACM SIGMETRICS Conf. Measurement and Modeling of Computer Syst., May 1986.
- [7] JACK SHIRAZI: *Java Performance Tuning, Second Edition*, O'Reilly,
- [8] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: Univ. Michigan Press, 1975.
- [9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proc. IEEE Int. Conf. Neural Netw., Nov.–Dec. 1995, pp. 1942–1948.
- [10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in Proc. IEEE Int. Conf. Evol. Comput. May 1998, pp. 69–73.
- [11] X. Zhang, S. Hu, D. Chen, and X. Li, "Fast covariance matching with fuzzy genetic algorithm," *IEEE Trans. Ind. Eng.*, vol. 8, no. 1, pp. 148–157, Feb. 2012.
- [12] W. Ip, D. Wang, and V. Cho, "Aircraft ground service scheduling problems and their genetic algorithm with hybrid assignment and sequence encoding scheme," *IEEE Syst. J.*, vol. PP, no. 99, p. 111.
- [13] F. Gonzalez-Longatt, P. Wall, P. Regulski, and V. Terzija, "Optimal electric network design for a large offshore wind farm based on a modified genetic algorithm approach," *IEEE Syst. J.*, vol. 6, no. 1, pp. 164–172, Mar. 2012
- [14] H. Cheng and S. Yang, "Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks," *Eng. Appl. Artif. Intell.*, vol. 23, no. 5, pp. 806–819, 2010
- [15] A. Pathan and R. Buyya, "A taxonomy and survey of content delivery networks," *Grid Computing and Distributed Systems (GRIDS) Laboratory*, University of Melbourne, Parkville, Australia, vol. 148, 2006.
- [16] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Mass.: Addison-Wesley, 1989.M.
- [17] Suresh Babu Kuntumalla and Lakshumaiah Maddigalla, "Enhanced Load Balancing in Clustered Cloud-based Multimedia System", *International Journal of Engineering Trends and Technology (IJETT) – Volume 29 Number 4 - November 2015*.