Dispatcher Based Dynamic Load Balancing on Web Server System

Harikesh Singh, Jaypee University of Engineering & Technology, India
Shishir Kumar, Jaypee University of Engineering & Technology, India

ABSTRACT

The traffic increasing in the network creates bulk congestion while the bulk transfer of data evolves. Performance evaluation and high availability of servers are important factors to resolve this problem using various cluster based systems. There are several low-cost servers using the load sharing cluster system which are connected to high speed networks, and apply load balancing technique between servers. It offers high computing power and high availability. A distributed website server can provide scalability and flexibility to manage with emergent client demands. Efficiency of a replicated web server system will depend on the way of distributed incoming requests among these replicas. A distributed Web-server architectures schedule client requests among the multiple server nodes in a user-transparent way that affects the scalability and availability. The aim of this paper is the development of a load balancing techniques on distributed Web-server systems.

Keywords: Cluster System, Dispatcher, Distributed System, Distributed Web Server System, Load Balancing

1. INTRODUCTION

In a distributed approach, firstly ensure that processing power is as close to the users as possible and the second is to ensure a high degree of robustness, for example via the use of data replication; and the third is to enable hardware to be easily added as the resource demands of the applications running on the distributed system start increasing. The best-known distributed system is the moderately simple World Wide Web which consists of a very large number of clients running browsers and a large number of web servers. A web server is the computer program that is responsible for accepting HTTP requests from the clients and serving them HTTP responses along with optional data contents which usually are web pages such as HTML docs and linked objects, etc.

A web server has both hardware and software requirements. A hardware requirement includes a high configured hardware with the high processor speed and high connectivity. In software requirements, there is a need of special software known as web server software. There is much web server software available today, e.g., Apache Tomcat Server, Microsoft Internet

DOI: 10.4018/ijsda.2012040102
Information Service (Microsoft IIS), Sun java web server, etc. Any single computer system having all the Internet services for a small group of company would include the HTTP server (Web pages and files), FTP server (file downloads), NNTP server (newsgroups) and SMTP server (mail service) also known as a web server system. Each of these services has integrated in a separate computer or in multiple computers in ISPs or large companies. A data centre for a large public Web site could contain hundreds and thousands of Web servers.

Load Balancing is simply an activity or process which is used to distribute the load which is constantly applied to single point to the different points. This technique is useful in many fields like in mobile communication, LAN Servers, WAN Servers, web servers, etc. It is required in websites for Inter Relay Chat; service provided by high bandwidth FTP servers, NNTP and DNS servers. In computer networks, it is a technique to spread work between two or more computers, links, CPU’s, hard drives or other resources in order to get optimal resources utilization, maximizing throughput, and minimizing response time.

2. RELATED WORK

2.1. Client-Based Approach

Any requests from popular websites can be routed from the client side in any replicated web server architecture even when the nodes are loosely (or not) coordinated. Routing to the Web cluster can be provided by Web clients or by client-side proxy servers (Cardellini et al., 1999). These approaches are not universally applicable. The way of accessing the Netscape Communication site through the Netscape Navigator browser is the first example of a client based solution.

Web clients, if they are aware of the Webserver system’s replicated servers, can actively route requests. After receiving a request, the Web client selects a node of the cluster and, after resolving the address mapping, submits the request to the selected node, which is then responsible for responding to the client. The same client can send another request and reach another server (Figure 1). Netscape Communications’ approach is one example of Web-client scheduling. The load-balancing mechanism for the Netscape Web-server system’s multiple nodes is as follows. When a user accesses the Netscape home page (http://www.netscape.com), Navigator selects a random number i between 1 and the number of servers and directs the user request to the node wwwi.netscape.com (Cardellini et al., 1998; Sanghi et al., 2002).

2.2. DNS-Based Approach

Distributed Web-server architectures that use request routing mechanisms on the cluster side are free of the problems of client-based approaches. Architecture transparency is typically obtained through a single virtual interface to the outside world, at least at the URL level. The cluster DNS—the authoritative DNS server for the distributed Web system’s nodes-translates the symbolic site name (URL) to the IP address of one server (Cardellini et al., 1999).

Moreover, every web client browser typically caches some address resolution besides providing a node’s IP address, the DNS also specifies a validity period (Time-To-Live, or TTL) for caching the result of the logical name resolution. When the TTL expires, the address-mapping request is forwarded to the cluster DNS for assignment to a Web-server node; otherwise, an intermediate name server handles the request. We distinguish the DNS-based architectures by the scheduling algorithm that the cluster DNS uses to balance the Web-server nodes’ load (Figure 2). With constant TTL algorithms, the DNS selects servers on the basis of system state information and assigns the same TTL value to all address-mapping requests. Alternatively, adaptive TTL algorithms adapt the TTL values on the basis of dynamic information from servers and/or clients (Cardellini et al., 1998; Sanghi et al., 2002).
Clinical Practice Guidelines Formalization for Personalized Medicine

www.igi-global.com/article/clinical-practice-guidelines-formalization-for-personalized-medicine/95956?camid=4v1a