Funambol™ Data Synchronization Server Scalability

Introduction

With the popularity of the Internet, single system websites and services often cannot handle the load. New load balancing technologies have been developed to combine multiple systems into an integrated single service. In simple terms, they combine the processing power of multiple machines into a more powerful service with improved Quality of Service (QoS).

Different load balancing approaches address different areas of concern and QoS. All offer high availability and scalability of services by means of fault tolerance and distribution. This document addresses the specifics of clustered load balancing for the Funambol Data Synchronization server to achieve scalability.

Clustered Load Balancing

Various configurations of load balancing networks are possible. Most provide a different level of redundancy and, therefore, a certain availability as well as different methods of scalability and flexibility. Clustering is defined as a group of application servers that transparently run J2EE applications as if they were a single entity. Two forms of clustering are common: *vertical clustering* where you run multiple servers in a single system or *horizontal clustering* where a single server runs on multiple systems. An additional administrative server (also called the load balancer) distributes the traffic load for the service over the multiple J2EE applications.

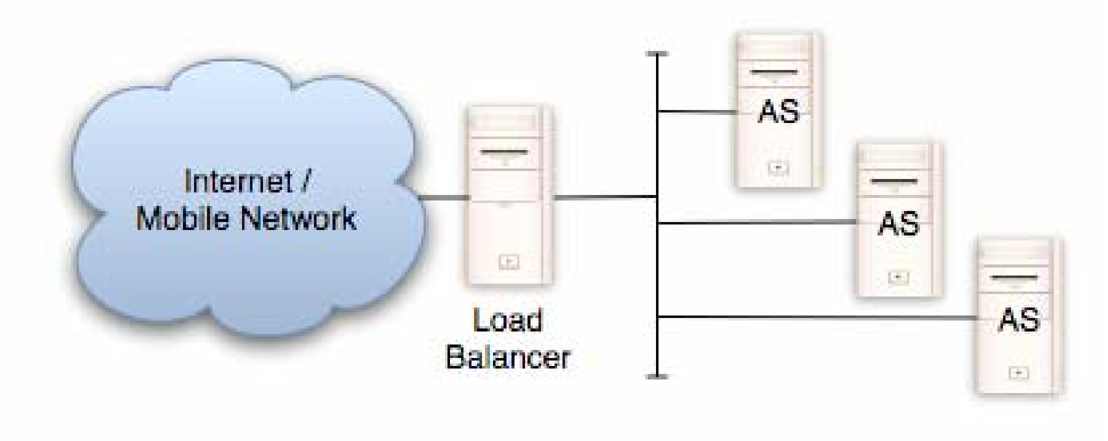

Figure 1: Clustered Load Balancing

The diagram illustrates the minimal network setup for supporting a horizontal clustered load balancing environment for the Funambol Data Synchronization server. On the left, you have the Internet with its customers and devices who want to access the service. The load balancer is connected to the Internet with a back-end network to which multiple application servers are connected. With this setup, the load balancer can divide the processing for incoming requests over the application servers. The application servers share a persistent store for SyncML session information and user data, as not all SyncML sessions for the same user might necessarily go to the same application server.

The cluster implementation for this environment can be based on two principals. The first is a sticky session in which the load balancer, based on a session id, always directs requests to the same application server. This requires no additional resources or functionality from the application server. The second principle is session replication.

Sticky Session Cluster

One of the most popular approaches is the combination of Apache and Tomcat. It uses a front-end – the load balancer – containing an Apache http server and application servers containing Tomcat with the Funambol web application. This is depicted below.

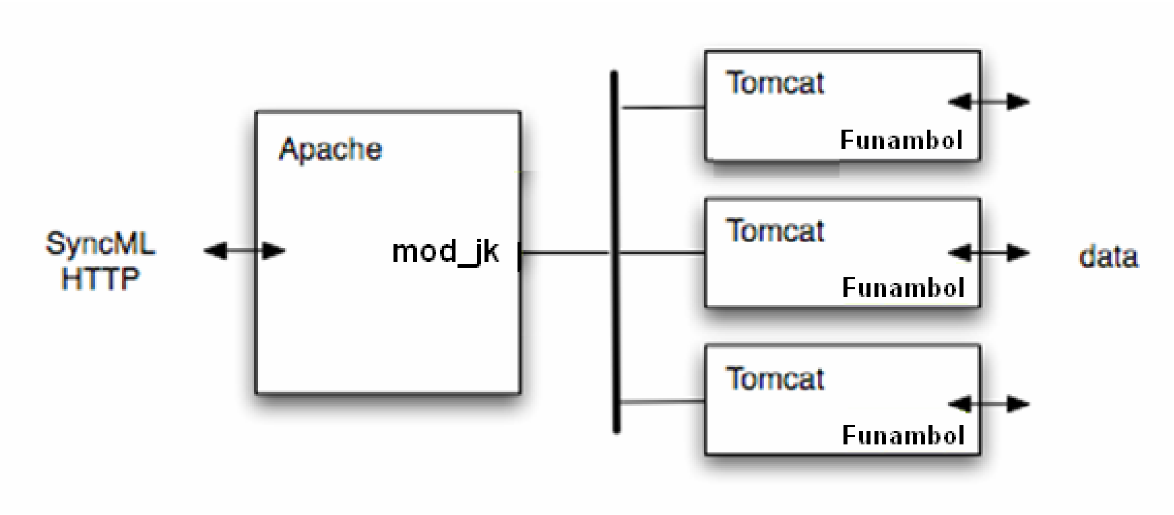

Figure 2: Apache –Tomcat Configuration

The diagram illustrates a setup of Apache that includes the Jakarta module with multiple Tomcat installations that embed the Funambol web application. The Apache server only needs to be configured with sticky session to have http requests/responses of the same SyncML session going to the same back-end Tomcat servers. The only thing the Tomcat installations need to share is access to storage containing data that is synchronized and the store to maintain SyncML persistent data between sessions.

Requirements

Every solution requires a different set of hardware and software. Below is a list of requirements for providing a load balanced Funambol Data Synchronization server solution as described in this document.

Software

1. Apache http server
2. Jakarta module for Apache
3. Tomcat
4. Database, for instance mysql (persistent store of SyncML info)
5. Funambol Data Synchronization Server

Hardware

1. One high performance system for the Apache installation
2. N systems for the horizontal scaling of Tomcat with Funambol
3. One system for the database of the persistent store

Depending on the data storage chosen, additional requirements might apply e.g. database storage could be integrated with the database also used for the persistent store.

Conclusion

The described load balancing solution is based on the well-known Apache-Tomcat configuration. It utilizes an open source infrastructure, lowering the cost of ownership. The solution is based on a well-known approach that can be combined with other load balancing techniques when needed to achieve the required QoS. QoS parameters like fault tolerance, availability and performance might involve additional requirements.

About Funambol

Funambol is the mobile open source company. Funambol's Mobile Application Server offers "push" email, multimaster PIM synchronization, and management facilities for mobile devices. Funambol, formerly known as Sync4j, is an open source development platform for mobile applications that has been downloaded more than any other wireless middleware product – 750,000+ times. The commercial version has been deployed at wireless carriers, Fortune 100 enterprises, hardware OEMs/ODMs and ISVs including customers such as Computer Associates. Funambol is headquartered in Redwood City, California with a development center in Italy. For more information, please visit http://www.funambol.com.

Worldwide Headquarters
643 Bair Island Road, Suite 305
Redwood City, CA 94063 USA
(650) 701-1450 x105

Europe
Via dei Valtorta 21
20127 Milano Italy
+39 02 2614 5383