

Scalable Architecture for High Traffic Web Application

Abstract

This paper describes the proposed architecture for SpryLive, a Video on Demand Web Application, in order to allow a scalable solution. This architecture may also be used for other applications with the specific changes.

Prerequisites

Hardware used:

- 5 Web Servers
- 2 Database Servers
- 4 Flash Media Servers

Software used:

1. Linux as operating system
2. Apache and nginx as web servers
3. MySQL 5.x for database
4. Adobe Flash Media Server 3.5 for streaming
5. PHP 5.2.6 as the scripting language for the web application
6. Perl as the scripting language for transparent jobs (servers status, moving files, backup)

Horizontal Scalability

Horizontal scalability offers the possibility to add as many servers as we want in order to keep the solution up and running. The bigger is the number of users using the application, the bigger are the hardware requirements. The number of servers we may add is unlimited. Also, the architecture is capable of distribute nodes in different geographic locations.

Web servers will be:

- *gateway.sprylive.com* - this server will be used to interpret the request and redirect to the dynamic or static servers.
- *dynamic1.sprylive.com* - this server will be used to serve only the dynamic content (PHP scripts)
- *dynamic2.sprylive.com*
- *static1.sprylive.com* - this server will be used to serve the static content (movies, images, css files, javascript files etc.)
- *static2.sprylive.com*

A disaster recovery solution and a monitoring solution will be installed in order to reduce accidental downtime.

The database server will be:

- database1.sprylive.com - the main database server
- database2.sprylive.com - secondary server used for backup purposes

The Flash Media Servers will work in a cluster to avoid bottlenecks and the components of the cluster are:

- edge1.sprylive.com
- edge2.sprylive.com
- edge3.sprylive.com
- edge4.sprylive.com

Vertical Scalability

In order to obtain a high number of concurrent connections for the proposed hardware architecture, we covered the following software aspects:

1. Avoid unnecessary code through a clean design of the application
2. Database design optimization
3. Avoid unnecessary database calls
4. Code optimization
5. SQL queries optimization
6. HTML code optimization
7. Proper usage of caching mechanisms

The vertical scalability, as a software approach, should be made in earlier software development phases. Trying to vertical scale an application when it's already in production it's possible but may lead to higher costs.