



Chapter VIII

Automatic Replication for Web Applications

Pablo Galdámez, Instituto Tecnológico de Informática, Spain

Abstract

This chapter proposes a software architecture to enhance the availability of Web-based applications at a reasonable cost. The architecture is pluggable to existing Web-applications, so that no modifications are required to existing code. There are no specific hardware or server software requirements. Availability is achieved by replication, and strong replica consistency is automatically guaranteed by using off-the-shelf group communication components. The proposed architecture is explained in detail, and it is compared against some of the most relevant highly-available solutions used nowadays. The most distinctive feature of the proposed architecture is that it explicitly addresses replica consistency, providing automatic strong replica consistency. Besides automatic consistency, attention is given to simplicity and pluggability.

Introduction

Since the invention of the World Wide Web, we have seen an explosive increase in the number of running Websites and a continuous increase in the importance of Websites as fundamental parts of business strategies. Today, more and more companies and organizations rely on Websites to offer their products and services, up to the extent that Website unavailability means, in many cases, profit losses. As Website availability is increasingly seen as a must, Website owners need to invest resources to reduce down time to a minimum. To that end, several technologies have been developed over the past years and probably some others will appear soon.

Big companies usually solve the availability problem with large investments in hardware, software, and personnel that can be prohibitive for medium and small businesses. That is one of the reasons why small companies usually lack a real highly-available solution. Most of them just blindly rely on their Internet providers. Internet providers host Websites on shared or dedicated servers running, in the best case, on high-end computers with replicated disks and uninterruptible power supplies. Availability is “guaranteed” by the hardware quality they provide for the Website, by the quality of their Internet connection, and by manual intervention in case of crashes. Even though this common solution may seem enough for a wide range of Websites, there are many others that cannot afford the potential downtime. These increasingly greater numbers of small companies do need an affordable true highly-available solution. Unfortunately, high-availability concerns are only perceived when unavailability appears. In this sense, high availability is similar to other computer system threats, such as those related to computer security, that disrupt computer-based businesses. The solution, as in those cases, is a combination of awareness, prevention, and a balance between cost and risk. Risk should be measured in potential profit losses, and cost should include hardware, software, personnel, and technical support.

This chapter focuses on affordable highly-available solutions for small Websites, where reduced cost implies low hardware cost, no significant code development or recoding, and little proprietary software dependencies. In particular, this chapter is specifically devoted to explain mechanisms for automatic replication applicable to existing Web applications. To do so, the chapter shows a quite simple architecture that may cover a gap among the currently-existing highly-available solutions, providing an affordable true highly-available solution. The architecture is basically made by integrating a set of inexpensive off-the-self components, it does not require specialized hardware or software, and it does not need the Website to be recoded (if it fits a specific class of Websites). However, it is a true highly-available solution that could be installed over a cluster or span a set of physically-separated servers. It guarantees full consistency among the server replicas, and recovery time after failures is negligible from the user perspective.

The central core of the architecture is a simple piece of software that should be implemented. In this sense the proposed architecture could be interesting for Website owners and Internet providers that look for cheap highly-available solutions for their current Websites, but it could be even more interesting for system programmers and highly-available solution providers looking for new ideas for their products.

The chapter covers the subject in depth, explaining technical details required to understand and develop the proposed architecture. Readers will benefit if they have some computer

24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/automatic-replication-web-applications/5221

Related Content

The Development of Decentralized Governance Models for Web 3 Ecosystems
Tanuj Surve and Risha Khandelwal (2023). *Concepts, Technologies, Challenges, and the Future of Web 3* (pp. 91-107).
www.irma-international.org/chapter/the-development-of-decentralized-governance-models-for-web-3-ecosystems/329858

Hyperlink Structure Inspired by Web Usage
Pawan Lingras and Rucha Lingras (2010). *Web Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 2034-2047).
www.irma-international.org/chapter/hyperlink-structure-inspired-web-usage/37728

PowKMeans: A Hybrid Approach for Gray Sheep Users Detection and Their Recommendations
Honey Jindal, Shalini Agarwal and Neetu Sardana (2018). *International Journal of Information Technology and Web Engineering* (pp. 56-69).
www.irma-international.org/article/powkmeans/198359

A Hybrid Cloud Model for Cloud Adoption by Multinational Enterprises
Wu He and Feng-Kwei Wang (2016). *Web-Based Services: Concepts, Methodologies, Tools, and Applications* (pp. 881-905).
www.irma-international.org/chapter/a-hybrid-cloud-model-for-cloud-adoption-by-multinational-enterprises/140834

Class Level Test Case Generation in Object Oriented Software Testing
N. Gupta, D. Saini and H. Saini (2008). *International Journal of Information Technology and Web Engineering* (pp. 18-26).
www.irma-international.org/article/class-level-test-case-generation/2644