Chapter 11

Geographically Distributed Cloud-Based Collaborative Application

Bogdan Solomon  
University of Ottawa, Canada

Dan Ionescu  
University of Ottawa, Canada

Cristian Gadea  
University of Ottawa, Canada

Marin Litoiu  
York University, Canada

ABSTRACT

The amount of multimedia content on the Internet has been growing at a remarkable rate, and users are increasingly looking to share online media with colleagues and friends on social networks. Several commercial and academic solutions have attempted to make it easier to share this large variety of online content with others, but they are generally limited to only sending Web links. At the same time, existing products have not been able to provide a scalable system that synchronizes disparate Web content sources among many users in real-time. Such a goal is especially desired in order to provide the benefits of cloud deployments to collaborative applications. Many Web-based applications cannot predict the number of connections that they may need to handle. As such, applications must either provision a higher number of servers in anticipation of more traffic, or be faced with a degradation of the user experience when a large number of clients connect to the application. Cloud-based deployments can alleviate these issues by allowing the application’s server base to auto scale based on the user demand. A cloud deployment can also employ servers in different geographic locations in order to offer better latency and response times to its clients. Moving a collaborative application from using a single server to a cloud and then to a distributed cloud is not a trivial matter, however. This chapter will show our experience with how such a transition can be performed, and will present the architectural changes that had to be implemented at the server and cloud level in order to create a distributed execution that resides in the cloud.

DOI: 10.4018/978-1-4666-2488-7.ch011
INTRODUCTION

Web 2.0 has dramatically transformed the way in which information is collected and presented to online users, and the enormous popularity of social networking has created a growing appetite for online multimedia content. Users are no longer satisfied with just viewing simple HTTP pages but expect the ability to share and collaborate with other people, such as friends or colleagues, online and in real-time. This can be seen with services like Google Docs (2011), where multiple users can work on the same document at the same time, and the document is stored on Google’s remote servers in the cloud. Other popular websites which offer the ability for online content sharing between users include Facebook and Twitter, yet in both cases, a message posted by a user containing links to photos or videos is later viewed by one or more users separately. In many ways, this is in no way different than sending an e-mail with either links or attachments.

Applications which allow users to collaborate in real-time have been developed by companies like Cisco in the form of WebEx (Cisco, 2011) and Citrix in the form of GoToMeeting (Citrix, 2011). Such applications generally use extra Web browser plugins in order to achieve the collaboration; however, a number of commons characteristics can be seen:

1. Users can communicate via text as well as video/audio.
2. All users view the same shared state of the collaboration.
3. Users can bring external data (documents) into the collaboration.

A similar collaborative application called “Watch Together” was developed and presented in Gadea (2011) by the authors. Watch Together used Adobe Flash on the client side, which connected to an open-source Java based server named Red5. The server application was responsible for ensuring that the client state remained synchronized across all the clients in a collaborative session. The server application also managed the creation of sessions, the joining of users to existing sessions, and the destruction of sessions. Figure 1 shows the user interface of Watch Together with six collaborative users sharing a YouTube video. Two of the users are streaming video/audio from their Webcams.

While the single-server application performed the desired functionality, a single-server architecture is not scalable. A simple JEE server can be clustered without much difficulty since, from a scalability point-of-view, it will not matter which server the clients connect to. In the case of a collaborative application, data must be passed between servers while collaborative session data has to be replicated across all servers to which clients are connected. This problem becomes even more complicated when the cloud, and therefore the server, is distributed across multiple geographic locations.

This chapter will present an approach to moving the collaborative application from a single server to a cloud, and then to a geographically distributed cloud. The rest of the chapter is structured as follows: Section 2 will present the requirements for the geographically distributed cloud-based application. Section 3 will introduce the architecture of the single-server application. Section 4 will show the changes made to move the application to the cloud, while Section 5 will show the changes which are required to make the application run in a distributed cloud. Finally, Sections 6 and 7 will present future work and conclusions.

REQUIREMENTS AND OVERVIEW OF THE MIGRATION PROCESS

The current application is a Web-based client-server application in which many clients connect
Related Content

Activity Theory to Guide Online Collaborative Learning Instructional Design
Siti Rosni Mohamad Yusoff and Nor Azan Mat Zin (2012). *International Journal of Systems and Service-Oriented Engineering* (pp. 15-29).
[www.igi-global.com/article/activity-theory-to-guide-online-collaborative-learning-instructional-design/78916?camid=4v1a](www.igi-global.com/article/activity-theory-to-guide-online-collaborative-learning-instructional-design/78916?camid=4v1a)

A Formal Language for XML Authorisations Based on Answer Set Programming and Temporal Interval Logic Constraints
[www.igi-global.com/article/formal-language-xml-authorisations-based/52594?camid=4v1a](www.igi-global.com/article/formal-language-xml-authorisations-based/52594?camid=4v1a)

Method Using Command Abstraction Library for Iterative Testing Security of Web Applications
[www.igi-global.com/chapter/method-using-command-abstraction-library-for-iterative-testing-security-of-web-applications/188208?camid=4v1a](www.igi-global.com/chapter/method-using-command-abstraction-library-for-iterative-testing-security-of-web-applications/188208?camid=4v1a)

Separation of Concerns in Mobile Hypermedia: Architectural and Modeling Issues
[www.igi-global.com/chapter/separation-concerns-mobile-hypermedia/66469?camid=4v1a](www.igi-global.com/chapter/separation-concerns-mobile-hypermedia/66469?camid=4v1a)