Performance Evaluation of a Modern Web Architecture

Johan Andre Lundar, School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, UK

Tor-Morten Grønli, Norwegian School of Information Technology, Oslo, Norway

Gheorghita Ghinea, School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, UK, & Norwegian School of Information Technology, Oslo, Norway

ABSTRACT

The past decade has marked a shift in Web development as users have become accustomed to Web applications with dynamic content and enhanced user experience. Different languages and technologies have been utilised to make way for such applications, gradually stretching existing standards and protocols to its limits. The field of Web development has been characterized by a lack of coherent architectural concepts, partly due to the absences of standard protocols that support modern communication styles. As a result of this, systems specifically designed for real-time data delivery have been required and realised with proprietary technology in the industry, consequently violating established software engineering principles such as modularity, consistency and simplicity. This paper explores how current Web technologies support the requirements of modern Web applications. A prototype application was developed in the last phase to demonstrate the efficacy of combining the WebSocket protocol and API together with the use of JavaScript as exclusive programming language at the client and server. Based on the findings of the research carried out it appears that the use of protocols and standards such as WebSocket, WebSocket API and Server-Sent Events caters for higher network performance, increased flexibility and standards compliance.

Keywords: AJAX, Javascript, Node.js, Performance, Representational State Transfer (REST), Web Architectures, WebSocket

1. INTRODUCTION

The most complete and appropriate style describing the general Web architecture is probably the Representational State Transfer (REST) style described by Fielding and Taylor (2002). An important principle in the work of Fielding and Taylor (2002) is that interaction between the client and server is performed synchronously over an open, standard network protocol; the Hypertext Transfer Protocol (HTTP) specified by Fielding et al. (1999). While the REST style has guided the creation of the modern Web, there has been a dissonance between the style and what has been realised in working systems in the recent years (Erenkrantz et al., 2007). AJAX

DOI: 10.4018/jitwe.2013010103
applications are possibly the most apparent sign of dissonance between the REST style and what has been engineered in the recent years.

2. RELATED WORK

According to Mikkonen and Taivalsaari (2007) there is a need to define the structure of Web applications more precisely to bring back tried and true principles and methods of software engineering. Moreover, the lacks of standard protocols that facilitate modern communication styles have especially affected architectural properties negatively in Web applications (Mesbah & van Deursen, 2008). To be able to understand these issues it is essential to begin to research the “classical” Web architecture and its influence on the success of the Web. Research presenting derivations of the classical Web architecture will then be visited before the focus is turned to Web engineering areas such as data delivery, programming styles and languages. The final section presents emerging technologies that seek to facilitate the engineering of contemporary Web applications.

2.1. Architectural Styles

The most complete and appropriate style describing the general Web architecture is probably the Representational State Transfer (REST) style described by Fielding and Taylor (2002). Fielding and Taylor (2002) focus on the rationale behind the Web’s architectural design and the software engineering principles upon which it is based. Visibility, reliability and scalability are presented as key factors guiding the architectural constraints of the REST architecture and Fielding and Taylor (2002) state that these can help to explain the success of the Web.

While the REST style has guided the creation of the modern Web, there has been a dissonance between the style and what has been realised in working systems in the recent years (Erenkrantz et al., 2007). Different Web architectures have therefore been proposed to try and bridge the gap between style and implementation. Khare and Taylor (2004) present and evaluate different extensions to the REST style to support distributed and decentralised systems. One of the issues presented by Khare and Taylor (2004) is that REST only allows one-way communication. Every request has to be instantiated by the client and every response has to be generated immediately, not only precluding servers from sending notifications to a client but also making asynchronous notifications impossible. While this ensures high scalability, it also makes it difficult to minimise latency. Khare and Taylor (2004) mentions a possible extension as REST with Polling (REST+P), described as a variant that can lower latency to some extent. This description is similar to what most Web applications have actually utilised in recent years, namely polling the server at regular intervals to provide high data accuracy and freshness (Bozdag, Mesbah, & van Deursen, 2009). However, the REST+P is presented only as a possible variant and not a proper solution to latency issues. Asynchronous REST (A+REST) is presented as a better solution. A+REST re-cast the concept of a resource in REST as an event source that emits event notifications, thus tackling the challenge of latency.

Event observation and notification is not a new concept. Rosenblum and Wolf (1997) argued that event observation and notification was a common architectural style for distributed systems, but that this approach was not yet scalable to the Internet. Khare & Taylor (2004) acknowledge that this concept is hard to realise in practice, but technologies such as the Web-Socket protocol (The Hypertext-Bidirectional working group, 2010) are emerging at the time of writing and can possibly change this. Several other extensions to REST is also presented by Khare and Taylor (2004) and ultimately combined to an event-based architectural style called ARRESTED. The other extensions include REST with Delegation (REST+D) and REST with Estimates (REST+E), but these are not central to the topic of this dissertation.

Erenkrantz et al. (2007) examines the REST style in relation to emerging Web architectures and argue that developers are struggling towards interactions far finer-grained than fetching
Related Content

Secure Online DNS Dynamic Updates: Architecture and Implementation
[www.irma-international.org/article/secure-online-dns-dynamic-updates/2630/](www.irma-international.org/article/secure-online-dns-dynamic-updates/2630/)

Motion, Graphics and Multimedia Production
[www.irma-international.org/chapter/motion-graphics-multimedia-production/31190/](www.irma-international.org/chapter/motion-graphics-multimedia-production/31190/)

Context-Aware Service Provisioning in Next-Generation Networks: An Agent Approach
[www.irma-international.org/article/context-aware-service-provisioning-next/2636/](www.irma-international.org/article/context-aware-service-provisioning-next/2636/)

An Approach Based on Service Components for Adapting Web-Oriented Applications
[www.irma-international.org/article/an-approach-based-on-service-components-for-adapting-web-oriented-applications/149999/](www.irma-international.org/article/an-approach-based-on-service-components-for-adapting-web-oriented-applications/149999/)

Design Challenges of Cloud Computing
[www.irma-international.org/chapter/design-challenges-of-cloud-computing/140817/](www.irma-international.org/chapter/design-challenges-of-cloud-computing/140817/)