Architectural Foundations of WSRF.NET

Marty Humphrey, University of Virginia, USA
Glenn Wasson, University of Virginia, USA

ABSTRACT

State management has always been an underlying issue for large-scale distributed systems, but only recently has it been brought to the forefront of Grid computing with the introduction of the Web Services Resource Framework (WSRF) and its companion, WS-Notification. WSRF advocates standardized approaches for client exposure to and potential manipulation of stateful services for Grid computing; however, these arguments and their long-term implications have been difficult to assess without a concrete implementation of the WSRF specifications. This paper describes the architectural foundations of WSRF .NET, which is an implementation of the full set of specifications for WSRF and WS-Notification on the Microsoft .NET Framework. To our knowledge, the observations and lessons learned from the design and implementation of WSRF. NET provide the first evaluation of the WSRF approach. A concrete example of the design, implementation, and deployment of a WSRF-compliant service and its accompanying WSRF-compliant client are used to guide the discussion. While the potential of WSRF and WS-Notification remains strong, initial observations are that there are many challenges that remain to be solved, most notably the implied programming model derived from the specifications, particularly the complexity of service side and client code and the complexity of WS-Notification.

Keywords: Grid computing; .NET framework; Web Services notification; Web Services Resource Framework

INTRODUCTION

There is probably no single best approach with regard to state management in distributed systems. The fundamental issue is not whether state exists in the services that comprise the distributed system (most people believe that the description of a non-trivial distributed system must include some representation of state) but rather what a client can assume about the state of the particular service with which the client wants to interact. For years, architects and system designers have compared the relative virtues of stateful services and stateless services. Simplistically, on one hand, it is argued that stateless services scale better and are more fault-tolerant, while, on the other hand, stateful services support terser messages that, hence, are more efficient, can be more intuitive to design, and, indeed, can scale well due to recent advances in software support for services. The general theme regards the notion of
conversation — specifically, what can and should a client say in its next request to the service?

Until recently, state management in Grid computing was not a first-class architectural concern. The Grid community largely relied on the Globus Toolkit (Globus Project, 2004), a collection of tools for wide-area, cross-domain computing. Prior to 2002-2003, Globus was not constructed as a collection of services, but rather a collection of semi-independent tools that individually facilitated remote job execution, remote file transfer, and so forth. The Globus Toolkit lacked a single architectural principle with regard to state management.

In 2002-2003, the Open Grid Services Infrastructure (OGSI; Tuecke et al., 2003), under the broader umbrella of the Open Grid Services Architecture (OGSA; Foster et al., 2002), synergized the traditional approach of performing Grid computing via Globus (Globus Project, 2004) or Legion (Grimshaw et al., 1999) with the emerging commercial approach of Web Services. Web Services would provide much of the underlying XML-based protocols for communication between services, while OGSI would provide a canonical referencing and/or manipulation of state. That is, OGSI constrained the appearance (to potential consumers) and behaviors of services, arguing that such constraints would make the overall service composition and subsequent execution more predictable and easier to assess and manage.

In January 2004, a team from IBM and the Globus Alliance introduced the Web Services Resource Framework (WSRF) as an attempt to refactor many of the concepts in OGSI in order to be more consistent with today’s Web Services (Czajkowski et al., 2004a). In contrast to early versions of the Globus Toolkits, the central theme of WSRF is the manipulation of state. In the W3C’s Web Service Architecture (WSA; Booth et al., 2004), services are either stateless, or any reference to state in the client-server protocol is an application-level concern. In WSRF, the argument is that there is great value in the canonical referencing and/or manipulation of state, paralleling the argument in OGSI that there was great value to the canonical behavior and appearance of services. The difference between OGSI and WSRF is that WSRF requires no modification to Web Services tooling. Of course, the significant research challenge for the community is to determine the extent to which WSRF and WS-Notification adds value above the Web Services approach. Almost immediately after the introduction of WSRF, a healthy debate emerged on this subject, particularly its similarities and differences with the Web Services Composite Application Framework WS-CAF (Little, Webber, & Parastatidas, 2004) and REST (Fielding, 2000).

WSRF.NET is an implementation of the WSRF specifications and WS-Notification on the Microsoft .NET Framework and is the first publicly available implementation of the full set of specifications for WSRF and its associated WS-Notification (WSRF Project, 2004). In this paper, we build upon and update our earlier assessment of WSRF.NET (Humphrey et al., 2004). In creating WSRF.NET, we significantly leveraged our experience designing and implementing OGSI on .NET, OGSI.NET (Wasson et al., 2004). We describe how we have interpreted the WSRF and WS-Notification suite of specifications, and, most importantly, we attempt to assess the resulting package, particularly in terms of the programming model. We do not claim that our programming model is the only programming model for WSRF and WS-Notification, but we argue that it is a logical consequence of the implementation of the specifications on the .NET Framework. Additionally, the difference between WSRF and WSRF.NET was difficult at times to discern. Overall, while the potential of WSRF and WS-Notification remains strong, initial observations are that there are many challenges that remain to be solved, most notably the implied programming model derived from the specification, particularly the complexity of service-side code, client-side code, and WS-Notification.

The outline of this paper is as follows. After providing a brief overview of the WSRF approach at the time of this writing, we describe WSRF.NET, which is our open-source implementation of the WSRF suite of specifications. Next, we present a use-case for constructing...
Related Content

On Transforming a Road Network Database to a Graph for Localization Purpose
[www.igi-global.com/article/on-transforming-a-road-network-database-to-a-graph-for-localization-purpose/152333?camid=4v1a](www.igi-global.com/article/on-transforming-a-road-network-database-to-a-graph-for-localization-purpose/152333?camid=4v1a)
A Trust-Powered Technique to Facilitate Scientific Tool Discovery and Recommendation
[www.igi-global.com/article/a-trust-powered-technique-to-facilitate-scientific-tool-discovery-and-recommendation/132754?camid=4v1a](www.igi-global.com/article/a-trust-powered-technique-to-facilitate-scientific-tool-discovery-and-recommendation/132754?camid=4v1a)

Architectural Foundations of WSRF NET
[www.igi-global.com/chapter/architectural-foundations-wsrf-net/26918?camid=4v1a](www.igi-global.com/chapter/architectural-foundations-wsrf-net/26918?camid=4v1a)

Fog Computing Technology for Cooperative Information Processing in Edge-Centric Internet of Things Environments
(2019). *Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities* (pp. 55-88).
[www.igi-global.com/chapter/fog-computing-technology-for-cooperative-information-processing-in-edge-centric-internet-of-things-environments/235132?camid=4v1a](www.igi-global.com/chapter/fog-computing-technology-for-cooperative-information-processing-in-edge-centric-internet-of-things-environments/235132?camid=4v1a)