Enhancing the Trustworthiness of Web Services Coordination

Wenbing Zhao
Department of Electrical and Computer Engineering,, Cleveland State University, USA

INTRODUCTION

Many e-commerce companies such as Amazon.com, Yahoo.com and eBay.com have been offering Web services to their partners and customers. Through such Web services, new value-added services could be provided and hence higher revenues would be generated. Essentially, the Web services technology is transforming the World Wide Web from a predominantly publishing platform to a programmable platform, which undoubtedly will make it easier to conduct business online, and enable automated business-to-business communications (Papazoglou, 2003; Zhao et al., 2008). The Web services technology is particularly useful for Application Service Providers that offer various on-demand services and software-as-a-service (SAAS) to their customers (Chakrabarty, 2007). Such service-oriented computing is attractive to many businesses because they can save valuable resources and money by avoiding installing and maintaining sophisticated enterprise software on-site. Furthermore, most of business interactions are transactional, which require well-defined standardized coordination support. To meet this requirement, a number of specifications have been proposed and ratified by OASIS (Feingold & Jeyaraman, 2009; Little & Wilkinson, 2009; Freund & Little, 2009).

In this article, we present an overview of our recent works on enhancing the trustworthiness of Web services coordination for business activities and transactions. The approach is based on what we call application-aware Byzantine fault tolerance. We argue that it is impractical to apply general-purpose Byzantine fault tolerance algorithms for such systems in a straightforward manner. Instead, by exploiting the application semantics, much lighter weight solutions can be designed to enhance intrusion tolerance, and hence the trustworthiness of systems that require Web services coordination.

DOI: 10.4018/978-1-4666-5888-2.ch750

BACKGROUND

Web Services Coordination

In this section, we provide an overview of Web services coordination and its applications for business transactions and activities.

Web services interactions are becoming more and more complex in structure and relationships. More complex means we need longer time to execute them, because of business latencies and user interactions. The Web Services Coordination specification (WS-Coordination) (Feingold & Jeyaraman, 2009) describes an extensible framework for plugging in protocols that coordinate the actions of Web services applications. Such coordination protocols can be used to support a variety of business applications, including those that require strict consistency and those that require agreement of a proper subset of the participants. The framework enables a Web service to create a context needed to propagate an activity to other Web services and to register for a particular coordination protocol.

There are two types of business transactions. One follows the traditional atomic transaction semantics, and the other is referred to as business activities, which implies that the atomicity property may be relaxed. The former is suitable for short transactions that require strong atomicity, such as a fund transfer transaction. The latter is more suitable long running transactions, such as those used in supply chain management. Based on WS-Coordination, two specifications, namely Web Service Atomic Transaction (WS-AtomicTransaction) (Little & Wilkinson, 2009) and Web Service Business Activity (WS-BusinessActivity) (Freund & Little, 2009), were standardized by OASIS to address the coordination needs for common types of business transactions.

Web Services Atomic Transactions

The Web Services Atomic Transactions specification defines a coordination framework for Web services atomic transactions. In a distributed atomic transaction, all participants must reach the same final agreement as to whether the transaction has succeeded or not. This is ensured by the coordination mechanisms specified in WS-AtomicTransaction. In WS-AtomicTransaction, there are three actors for each transaction: Completion Initiator, Coordinator and Participants. Each provides a different set of services for the atomic transaction, and they interact with each other via two protocols, the Completion Protocol and the Two-Phase Commit Protocol (Gray & Reuter, 1993) (Tanenbaum & Steen, 2002).

The completion initiator is responsible to start and terminate a transaction. It also provides the Completion Initiator Service so that the coordinator can inform it the final outcome of the transaction, as part of the completion protocol. The coordinator provides the following services:

- **Activation Service:** At the beginning of a transaction, the initiator invokes the Activation Service for creating a coordinator object, which will generate a new coordination context for the transaction and return it to the initiator. The coordination context contains a unique transaction identifier and an endpoint reference for the Registration Service. This coordination context will be included in every request messages within the transaction boundary.
- **Registration Service:** The participants and the completion initiator use this service to register their endpoint references for other associated participant-side services. Later these endpoint references will be used by the coordinator to contact the participants during the two-phase commit.
- **Coordinator Service:** When a participant gets a Prepare request from the coordinator, it places its vote by invoking the coordinator service. The participants also use this service to notify the coordinator their acknowledgments to the commit/abort request. The participants obtain the endpoint reference of the Coordinator Service during the registration step.
- **Completion Service:** The initiator invokes this service to notify the coordinator to start a distributed commit. The Completion service, together with the Completion Initiator service on the participant side, implements the WS-AtomicTransaction completion protocol. The endpoint reference of the Completion Service is returned to the initiator during the registration step.

The participant provides the Participant Service, which allows the coordinator to solicit votes from, and to send the transaction outcome to the participants according to the two-phase commit protocol.

The Completion Protocol is used by the completion initiator to initiate the atomic termination of a transaction. When the initiator decides that it is time to commit the transaction, it sends a Commit request to the coordinator. The coordinator will then launch an instance of the Two-Phase Commit (2PC) protocol to carry out the coordination for atomic commitment of the transaction. When the 2PC completes, the coordinator notifies the initiator the outcome of the transaction (i.e., Committed or Aborted). If the request from the initiator is Rollback instead, the coordinator will abort the transaction directly.

The 2PC Protocol is used by the coordinator and participants to guarantee atomic commitment of a transaction, and it executes in two phases. During the first phase, i.e., the prepare phase, the coordinator sends a Commit request to all registered participants soliciting their votes. When the coordinator receives votes from all participants, or a timeout has occurred, it starts the second phase, i.e., the commit phase, to notify the participants the outcome of the transaction.

2PC has two variants used for different resources, Volatile 2PC and Durable 2PC. Volatile 2PC is used for volatile resources such as caches and Durable 2PC focuses on durable resources like a database. Participants must register in an appropriate protocol before the termination of the transaction. A participant can register in more than one protocol. Upon receiving a Commit request from the initiator in the completion protocol, the coordinator begins the prepare phase first for every participant who has registered in the Volatile 2PC protocol by sending a Prepare request to

Related Content