IDEA GROUP PUBLISHING **ITJ3477**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, **International Journal of Web Services Research**, Volume 4, Issue 1
edited by Liang-Jie Zhang © 2007, Idea Group Inc.

A Dynamic Two-Phase Commit Protocol for Adaptive Composite Services

Weihai Yu, University of Tromsø, Norway

Calton Pu, Georgia Institute of Technology, USA

ABSTRACT

Next-generation applications based on Web services impose additional requirements on the use of coordination protocols with various optimizations, such as the two-phase commit protocol (2PC). This article analyses the well-known 2PC optimizations "presumed commit" and "presumed abort," and presents an improved 2PC that is suitable for Web-service-based applications. More specifically, the protocol allows every individual service provider to choose dynamically the most appropriate presumption for any distributed transaction. This new capability is especially useful when a composite Web service is integrating component services that make different presumptions in their commit protocols. The protocol does not introduce extra overhead to the previous 2PC variants in terms of number of messages and log records, and it is easy to understand and realize. Our simulation shows that the choice of appropriate presumption has significant influence on system performance, and that in some heterogeneous settings, combining different presumptions in individual transactions outperforms adopting only one single presumption.

Keywords: adaptive composite services; distributed transactions; dynamic presumptions; two-phase commit

INTRODUCTION

One of the goals of Web services is to enable next-generation applications to be dynamically composed of invocations on different data services. The composition of services can be specified using Web services flow specification languages such as BPEL4WS (BPEL4WS, 2003). Consider, for instance, a virtual manufacturer consisting of a chain of distributed outsourced services such as marketing, R&D, procurement, manufacturing and distribution. Tasks within this supply chain typically consist of access to one or more data services. For example, approval of some engineering changes might involve updates in data services of a number of component manufacturers. Technically, these tasks can be modeled into a number of distributed transactions. For many of these transactions, it is crucial that global atomicity, among other properties, is guaranteed: Either

the entire transaction gets succeeded or nothing takes effect. The global atomicity becomes a non-trivial property when composing heterogeneous component services, one of such technical challenges being addressed in this article.

In addition to guaranteeing correctly the global atomicity of composite Web services, the runtime overhead and delay of commit protocols may lead to increased data contention that in turn leads to poor scalability and performance of applications. Trade-offs of various kinds are therefore necessary. At a higher level, weaker properties than global atomicity are adopted in order to achieve better scalability and performance at the expense of more sophisticated application logic. A complex task, such as the approval of engineering changes mentioned previously, is usually not modeled into a single globally atomic distributed transaction. Instead, application semantics and alternative correctness criteria are explored for long-running activities (Elmagarmid, 1992). However, globally atomic distributed transactions are still necessary to simplify application development. In theoretical models such as process algebra (Fokkink, 2002), processes are composed of actions that are assumed to be atomic, so that no exploration of their inner structure is needed. In practice, the complex task is decomposed into smaller inter-dependent atomic transactions, using mechanisms like recoverable message queues and compensation (Leymann, 1998). The smaller atomic transactions are distributed, typically involving a data service and a recoverable queue at a different site (Bernstein & Newcomer, 1997). This article contributes to the support of the smaller atomic transactions, where we provide global atomicity but relax the overhead and delays associated with the presumption part.

Atomic commit protocols are a key element in supporting global atomicity of distributed transactions. *Two-phase commit protocol* (*2PC*) is the de facto standard atomic commit protocol (Bernstein et al., 1987; Gray & Reuter, 1993). It is widely agreed that 2PC is important to guarantee correctness properties in the complex distributed world, whilst at the same time it reduces parallelism due to high disk and message overhead and locking during windows of vulnerability. Liu et al., (1994) even showed that distributed *commit* processing can have more effect on system performance than distributed *data* processing. With an increasing number of potential atomic actions in an N-tier application architecture based on Web services, the performance penalty of commit processing may be aggravated as the system grows in scale and size.

There are a number of optimizations to the basic 2PC (Chrysanthis et al., 1998; Gray & Reuter, 1993; Lampson & Lomet, 1993; Mohan et al., 1986; Samaras et al., 1995). Some of them are so widely used that they are built into commercial systems and become part of transaction processing standards, including OMG transaction service (OMG, 1998) and its Java mapping Java transaction service (Sun Microsystems, 1999), XA (X/Open, 1991), and the newly proposed WS-AtomicTransaction (WS-AtomicTransaction, 2005) for Web services.

Standard efforts reflect this composition for transactional Web-service-based applications. WS-Coordination (WS-Coordination, 2005) provides a general framework for supporting coordination among different service providers. Under this framework, two particular coordination types are defined: WS-BusinessActivity (WS-BusinessActivity, 2005) is used for decomposing a complex task into a set of inter-dependent atomic transactions; WS-AtomicTransaction (WS-AtomicTransaction, 2005) is used for providing global atomicity of distributed transactions. While these interfaces support the composition of different variants of commit protocols, a serious technical challenge arises when component services choose commit protocols with potentially incompatible presumptions that may jeopardize global atomicity of the composite service. Instead of forcing the components to make the same presumption, we believe a better solution is to develop an adaptive commit protocol that

Related Content

Security and Privacy Issues of Big Data

José Moura and Carlos Serrão (2019). *Web Services: Concepts, Methodologies, Tools, and Applications (pp. 2197-2229).*

www.igi-global.com/chapter/security-and-privacy-issues-of-big-data/217939?camid=4v1a

Quality Models for Multimedia Delivery in a Services Oriented Architecture

Krishna Ratakonda (2009). *Managing Web Service Quality: Measuring Outcomes and Effectiveness (pp. 48-73).*

www.igi-global.com/chapter/quality-models-multimedia-delivery-services/26074?camid=4v1a