

# Web 服务事务的研究综述<sup>\*</sup>

官荷卿<sup>1,2,3</sup>

(中国科学院软件研究所 软件工程技术研发中心 北京 100080)<sup>1</sup>

(中国科学院软件研究所 计算机科学重点实验室 北京 100080)<sup>2</sup>

(中国科学院研究生院 北京 100039)<sup>3</sup>

**摘要** Web 服务的出现对分布式事务提出了新的需求,为了保证多个 Web 服务交互获得正确的执行和一致性的结果,有必要为 Web 服务提供事务机制。但是 Web 服务具有松散耦合性、运行时间长等特点,这使得传统的事务处理技术在处理 Web 服务事务时显得力不从心。本文对 Web 服务事务的关键问题进行了阐述,包括 Web 复合框架的事务性支持、Web 服务事务规范以及扩展事务模型等方面的研究内容,并指出了目前研究的不足以及未来的发展趋势。

**关键词** Web 服务, Web 服务事务, 事务模型

## A Survey on Web Service Transaction

GUAN He-Qing<sup>1,2,3</sup>

(Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing 100080)<sup>1</sup>

(Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080)<sup>2</sup>

(Graduate School of the Chinese Academy of Sciences, Beijing 100039)<sup>3</sup>

**Abstract** As Web service technologies have become firmly established and widely adopted, traditional transaction semantics and protocols have proven to be inappropriate for some Web services-based applications. These particular Web services-based transactions differ from traditional transactions in that they execute over long periods, they require commitments to the transaction to be negotiated at runtime, and isolation levels have to be relaxed. In this paper, we present a survey on various aspects of the research of Web service transactions, including Web service composite framework, extended transaction models, Web service transaction specifications and so on. Lastly, we try to give an indication of where Web service transactions will go in the future.

**Keywords** Web service, Web service transaction, Transaction model

## 1 引言

Web 服务(Web Service)是 Web 发展的自然产物<sup>[1]</sup>。它是一种新型的 Web 应用,提供独立的、自描述的、模块化的应用,具有良好的交互性和跨平台、跨语言的能力。Web 服务提供了相互独立的网络平台上应用连接和信息集成的方法,在分布式环境中,Web 服务需要相互协同工作并保持一致,得到正确和可靠的运行结果。我们可以借助事务机制来解决这类问题。由于 Web 服务自身的特点,如松散耦合性、运行时间长等,传统的事务处理技术在处理 Web 服务事务时显得力不从心。

目前,Web 服务中的事务处理通常是扩展传统的事务处理技术,研究人员提出了一些 Web 服务的事务规范和模型,主要有 BTP(Business Transaction Protocol)<sup>[4]</sup>、WS-C/WS-T(WS-Coordination/WS-Transaction)<sup>[5]</sup>、WS-CAF(Web Services Composite Application Framework)<sup>[6]</sup>等;其它的相关研究工作主要来源于事务性过程、Web 服务复合等研究领域,包括 Web 服务复合的事务支持<sup>[10~14]</sup>、事务流程的调度和优

化<sup>[15~19]</sup>。总的来说,Web 服务事务的理论和技術仍有很多关键问题亟待解决。

本文首先概述 Web 服务事务的基本特征和面临的主要挑战;然后介绍 Web 服务事务的研究现状,包括事务规范、模型和支撑技术,分析当前主要的研究问题和核心技术;最后总结全文,指出 Web 服务事务研究面临的挑战,并展望未来的工作。

## 2 Web 服务事务的基本概念和主要问题

事务是具有 ACID(Atomicity, Consistency, Isolation, Durability)特性的原子操作序列,用来保证应用程序对数据资源访问的一致性和可靠性。随着网络技术的发展和应用程序需求的变化,事务处理发展成为了分布式事务处理,由专门的事务处理中间件提供事务服务,分布式事务已成为构造可靠分布式应用的基础<sup>[8]</sup>。实现分布式事务的关键技术包括并发控制、故障恢复(如原子提交协议)和资源管理等。在商用系统中,分布式事务通常遵循国际/工业规范,如 CORBA OTS、X/Open DTP、J2EE JTS。

<sup>\*</sup>本文研究得到国家 973 网构软件中间件平台模型和框架研究(编号 2002CB312005)、863 网络环境的系统软件核心技术及运行平台(编号 2001AA113010)、863 面向新型 ERP 的可重配 Web 应用服务器研究与应用(编号 2003AA413010)。官荷卿 博士研究生,主要研究分布式计算技术。

在 Internet 环境下, Web 服务的出现对分布式事务提出了新的需求, 并引发了一系列待解决的问题。Internet 应用被包装为 Web 服务, Web 服务在 Web 环境中不仅可被直接调用, 而且可被组合链接成为企业流程。这种流程和应用集成方式正在成为一种趋势。为了保证多个 Web 服务交互获得正确的、一致性的执行结果, 有必要为 Web 服务提供事务支持, 即 Web 服务事务机制。

众所周知, 目前大多数的事务管理器都是基于平面型事务模型实现的。这类事务管理器若要支持 Web 服务事务, 会面临很多困难, 主要表现在以下几点:

(1) Web 服务之间的松散耦合性。Web 服务并不等同于常规分布式事务中的服务, 不同的 Web 服务采用的实现技术不同, 有些 Web 服务本身提供的功能, 使得它可以像数据库对象一样参与到事务中, 有些则没有这些基本功能。但它们都可能参与到事务之中。而传统的事务处理技术并不会区别对待不同的参与者。

(2) Web 服务事务不一定要遵守传统事务的 ACID 特性, 尤其是原子性和隔离性。在 Web 环境下, 存在着各种各样的 Web 服务, 很多 Web 服务业务流程复杂, 运行时间超长。事务的隔离性是通过加锁机制实现的, 对于有这类 Web 服务参与的事务, 实现隔离性的代价太高, 会大大降低系统的性能。

(3) Web 服务事务参与者分布在 Internet 上, 彼此的交互通过广域网进行, 数据和管理都相对独立, 在设计 Web 服务的流程和协议时, 必须注意考虑管理和安全问题。例如为了完成事务参与者的互操作, 需要把事务的上下文在它们之间进行传递, 因而需要加入安全机制来防止事务上下文在传输过程中和驻留在目的地时不被伪造或篡改。

(4) 相比传统分布式事务, Web 服务事务有着更高的可靠性和可用性要求。因为 Web 服务的层次较高, 涉及面更广, 如果管理 Web 服务的事务监控器出现故障, 多个组织的业务流程可能都无法继续。

为了解决上述问题, 必须放宽 Web 服务事务对 ACID 的要求。Web 服务事务模型不但要支持传统的 ACID 事务, 还要能够支持非 ACID 事务, 以便适应 Web 服务事务自身的特点。

Web 服务兴起的时间不长, 所以针对 Web 服务事务的研究尚处于起步阶段, 已有工作还没有形成一个完整的体系。目前, 工业界侧重于研究 Web 服务事务规范和协议标准化, 而学术界则主要关注 Web 服务事务的模型和底层支撑技术。

### 3 Web 服务扩展事务模型

为了解决平面事务模型的不足, 研究者提出了各种扩展事务模型 (Extended Transaction Model, 也称为 Advanced Transaction Model)<sup>[25]</sup>。它们在平面事务的基础上加入特定的应用语义, 基于固定的控制和数据依赖, 在不同方面放宽了对 ACID 特性的要求, 以解决特定领域的问题。常见的扩展事务模型包括嵌套事务模型 (Nested Transactions)<sup>[25]</sup>、多层事务模型 (Multi-Level Transactions)<sup>[26]</sup>、Sagas<sup>[27]</sup>、分支汇合事务模型 (Split and Join Transactions)<sup>[28]</sup> 和柔性事务模型 (Flexible Transactions)<sup>[29]</sup> 等。Web 服务事务的研究可以从扩展事务模型的内容中获得借鉴, 从本质来讲 Web 服务事务模型就是一种复杂的扩展事务模型。表 1 显示了 Web 服务事务

和传统事务在 ACID 要求上的区别:

表 1 Web 服务事务和传统事务的比较

	传统事务	Web 服务事务
原子性 (Atomicity)	必需, 事务的操作要么全部生效, 要么全部不生效	放宽, 情况允许的话尽量满足原子性的要求
一致性 (Consistency)	必需	必需
独立性 (Isolation)	必需, 事务产生的效果, 只有在事务完成后, 其他事务才可见	放宽
持久性 (Durability)	必需, 事务效果永久保持	必需, 由原子性决定永久保存的集合

## 4 Web 服务事务规范

与学术界的研究相比, 工业界更注重的是 Web 服务事务规范和协议的标准化。在各个规范中对 Web 服务事务模型进行了一定的扩展。常见的 Web 服务事务的规范包括 BTP、WS-C/WS-T、WS-CAF 等。

### 4.1 BTP

BTP 是 OASIS 组织在 2002 年提出的一个面向 Web 应用、协调多个自主参与者的事务支持协议。它基于事务的两阶段提交, 提出用两阶段终止协议 (two-phase completion protocol) 来解决 Web 服务的事务问题, 在 Web 服务环境下支持长持续时间、参与者自主的 B2B 业务事务流程。

BTP 定义了两种类型的扩展事务:

(1) 原子 (Atom): 具有原子性, 即事务中的操作要么全部执行, 要么全部不执行。

(2) 内聚 (Cohesion): 放宽了原子性的要求。

尽管 BTP 使用两阶段协议, 但是它并不支持 ACID 语义。在事务提交的第一和第二阶段之间, BTP 需要用户明确地表明自身是否加入进来。

表面看来 BTP 给出了处理 Web 服务事务的可行方法, 然而在实际应用中并不好用。BTP 把参与者和事务协调者绑得太紧。如为了处理内聚, 在提交事务的时候必须明确指明哪些参与者需要提交, 哪些参与者需要取消 (cancel)。而在传统的事务系统中, 用户根本不必关系事务涉及到的参与者。可以说 BTP 对事务是一种集中式的管理, 这和 Web 服务的松散耦合性是相抵触的。另外, BTP 把几乎所有的工作, 如事务提交过程中的隔离性、事务的回滚操作等等, 都推给了后端参与者。这无疑增加了用户的工作量。最后, BTP 不能支持传统的 ACID 事务。

HP-WST (HP's Web Services Transactions)<sup>[21]</sup> 是业界第一个为支持 Web 服务上事务而设计并实现的事务服务系统。它基于 BTP 规范, 将小范围内的两阶段提交事务聚合为更大范围上的非 ACID 事务。HP-WST 包括三部分: HP-WST 协调者、事务参与者 Java 类库、Java 客户端类库。

### 4.2 WS-C/WS-T

IBM、Microsoft 于 2002 年 8 月发布了 WS-C/WS-T 规范。该规范建立在 SOAP、WSDL 等 Web 服务的标准之上。它在满足传统事务 ACID 性质的基础之上引入协调器 (Coordinator), 扩展了现有的事务处理技术, 提出了适用于 Web 服务的协同工作框架及协议。

WS-T 规范描述了与 WS-C 规范中描述的可扩展协调框架 (Coordination Framework) 一起使用的协调类型。它定义

了两种协调类型:原子事务(Atomic Transaction, AT)和业务活动(Business Activity, BA)。前者用于兼容原有的 ACID 事务,后者则可用于协调长事务等非 ACID 事务。在构建要求分布式活动的输出结果一致的应用程序时,开发者可以使用这两种协调类型之一或者同时使用两种。

WS-C/WS-T 的优点在于协调框架和事务类型的分开,具有良好的可扩展性。另外,WS-T 定义了两种事务类型,一方面,可以处理传统的 ACID 事务,这使得 Web 服务和传统的系统之间的互操作不会因为事务类型不一致而受到影响。另一方面,WS-T BA 的使用给了 Web 服务提供者很大的发挥空间,它们可以根据各自的 Web 服务自身的特点来提供个性化的补偿机制。

然而,WS-C/WS-T 并不完备。例如,它并没有给出处理 Heuristics 的方法,对于分布式恢复也言之甚少。

### 4.3 WS-CAF

2003 年 7 月,Arjuna、Fujitsu、IONA、Oracle、Sun 等公司发布了 WS-CAF 规范。该规范包括三个子规范:WS-Context (Web Service Context 或 WS-CTx)、WS-CF (Web Service Coordination Framework)、WS-TXM (Web Service Transaction Management)。

在 WS-CAF 中,定义了 Web 服务上下文的数据结构。WS-CTX 负责上下文的管理,它定义一种方式允许任意的 Web 服务增加上下文。WS-CF 定义了一个名为 coordinator 的软件级代理,用于管理上下文。

WS-TXM 定义三种类型的事务:

(1) ACID 事务:即传统的 ACID 事务。

(2) 长时间活动 LRA (Long Running Action), 不具有 ACID 属性,但是仍然具有原子性。和 WS-T 的 BA 非常相似,也使用补偿事务。

(3) 业务流程 BP (Business Process): 复杂的业务处理。

相比前面两种规范,WS-CAF 显得更有优势。它和 WS-C/WS-T 相似,但是更加完善。WS-CAF 能够处理 Heuristics。WS-TXM 定义的三种事务类型,使得它可以兼容传统的 ACID 事务。业务流程(BP)提供了良好的扩展性。值得一提的是,WS-CAF 框架允许 WS-C 或者 WS-T 插入其中。

### 4.4 其他规范

OMG Activity Service<sup>[22]</sup>, JSR95<sup>[23]</sup> 以及 JSR156<sup>[24]</sup> 等规范对 Web 服务事务研究都具有很重要的参考价值。OMG Activity Service 可以支持长时间、复杂的商业事务处理,但是 Activity Service 直接基于 OMG OTS 和 ORB,并没有 XML 方面的支持。JSR95 (J2EE Activity Service for Extended Transactions) 和 JSR156 (Java API for XML Transactioning) 是 J2EE 规范为了解决复杂的分布式事务处理所进行的努力。

### 4.5 小结

表 2 总结了 WS-C/T、WS-CAF 和 BTP 之间的各种不同点和相似点:

表 2 主要 Web 服务事务规范比较

	WS-C/T	WS-CAF	BTP
协调架构 (Coordination framework)	WS-C	WS-CF	没有,依赖于两阶段协议
事务处理框架 (Transaction framework)	没有,但目前定义的协议包含典型的模式(AT 和 BA);以后会添加其他的协议	没有	使用通用协议,没有模式。采用静态定义。
严格的原子模型 (Strict atomic model)	原子事务需要严格的 ACID 特性,特别是在与传统的的事务处理系统进行互操作时。	ACID 事务具有严格的 ACID 特性,专门为与传统的的事务处理系统进行互操作而设计的。	只具有原子性,其他属性通过服务指定(而通过协议指定无效)。使用 Open-top 协议;使得与现有的事务处理系统进行互操作变得非常困难
放松的事务模型 (Relaxed model)	业务活动允许灵活的参与者清单	LRA 和 BP 允许灵活的参与者清单	内聚允许灵活的参与者清单。需要把参与者公开给应用程序或总结器
范围 (Scope)	支持。业务活动管理了各个范围间的关系,而且允许范围嵌套	支持。LRA 支持 Scope 的任意级嵌套,BP 则可以划分为多个业务范畴 (business domain)	不支持。内聚管理范围内部的关系
能给多数人意见组灵活的结果	能,通过业务活动	能,通过 LRA 和 BP	能,通过内聚
参与多数人意见组的灵活性	有,根据业务活动协议,参与者可以退出	有,LRA 和 BP 的参与者可以选择退出	有,参与者可以除去内聚
服务行为 (Service behavior)	通过协议定义	通过协议定义	服务定义行为(不由 BTP 进行指定)
业务逻辑/协调器分离	截然分开	截然分开	混合在一起(Open-top 协议需要业务逻辑和协调器之间的强耦合)
专门用于 Web 服务	是	是	不是。需要在规范/协议方面做大量的额外工作
失败恢复能力 (Failure recovery)	协议已优化	支持分布式恢复 (distributed recovery),但还需要进一步的优化	重新驱动 (Re-drive) 协议

综上所述,Web 服务事务的规范并没有形成一个统一的版本,就上述的规范来说,它们存在很多缺点,最主要的缺点

如下:

(1) 不少规范最初并不是针对 Web 服务事务设计的(如

BTP),只不过它们也满足了 Web 服务事务的某些方面需求。这类规范并未涉及 Web 服务事务特有的问题。

(2) 对于一些关键的技术难点,如 Web 服务事务互操作时的安全性保证、事务隔离性(虽然绝大多数的 Web 服务事务都要放宽隔离性,但是放宽并不意味着完全不要)的实现等等,并没有给出切实可行的解决方案,全部需要由用户程序自己来实现,这无疑增加了用户程序的负担。

(3) 有些解决方案,如 WS-C/WS-T 协议,还处于研究状态,本身并不完备,它们对于一些细节,如错误处理,都尚未定义。

(4) 大多数解决方案都没有实现了的系统,有些甚至连原型系统都没有。

(5) 各个规范并不相互兼容,它们之间的竞争状态不利于 Web 服务规范的形成。

从研究的角度看,Web 服务事务已有的规范主要是给出了事务涉及方之间的交互,其中借鉴了扩展事务的研究成果,放宽了事务的 ACID 特性,但目前还未从理论上给出 Web 服务事务的模型,也缺乏相应的形式化描述和良构性证明。同时已有的工作考虑到 Web 服务的自治性,对 Web 服务事务的支持,基本是通过补偿事务方式,从而避免整个 Web 服务复合被放弃。但补偿事务的语义等价很难实现,因为当需要回滚某个事务时,由于其各个 Web 服务的处理结果可能已经被其他事务所使用,所以不仅应对其各个 Web 服务本身进行补偿,还应应对所有直接和间接依赖于该 Web 服务的其他事务也进行补偿,而这将是非常困难的,级联回滚也会造成系统性能的降低。同时,由于很难事先预测所有依赖于各 Web 服务的其他事务的集合,因而很难事先编写 Web 服务事务的补偿事务程序,从而造成在补偿时不得不依赖于人工干预才能完成。另外,由 Web 服务的提供方逐个提供补偿事务,这本身也不是件容易的事,可操作性差。要使得 Web 服务事务走向实用必须要在模型级界定清楚补偿的作用,避免补偿事务本身带来的缺陷。

## 5 Web 服务复合的事务支持

Web 服务复合(Web Service Composite)为复杂的 Web 应用提供了有效的解决方案,提高了 Web 服务组件的可重用性和利用率。在进行 Web 服务复合时如何提供对 Web 服务事务的支持也是目前 Web 服务复合研究工作需要解决的问题之一。

### 5.1 Web 服务复合框架

文[10]介绍的 WebTransact 从 Web 服务复合的框架出发,提供了 Web 服务的事务功能。它在 WSDL 的基础上提出了一种 Web 服务事务语言—WSTL(Web Service Transaction Language),并定义了与之对应的事务模型。其中,WSTL 用于描述 Web 服务的内容和事务属性,以及定义 Web 服务的事务交互模式,而 WebTransact 的事务模型给出了名为 2L-guaranteed-termination 的正确性标准,此标准针对 Web 环境放宽了事务的原子性。然后,根据这个正确性定义给出了保障 Web 服务正确执行的协议。但 WebTransact 框架没有全部实现。

文[11]使用了名为 Unit of Work 的嵌套事务模型框架,在此框架下,用户并发访问共享数据时并不对资源加锁以提高并发度。此框架还提出一套编程接口,以对长时间运行的业务流程提供事务支持。

### 5.2 Web 服务复合的工作流语言

若干用于 Web 服务复合的工作流语言,如 XLANG<sup>[12]</sup>、BPEL4WS<sup>[13]</sup>、WSFL<sup>[14]</sup>,对 Web 服务事务也提供了一定程度的支持。XLANG 是微软在 Web 服务复合领域提出的一种语言,主要应用于描述企业间的 Web 服务交互。XLANG 可以支持长时间运行事务,同时异常处理和相应的恢复操作也可以在 XLANG 中指定,而且在 XLANG 中可以集成 Web 服务回复和超时的要求,但是在 XLANG 中无法定义 Web 服务的 QoS 属性。BPEL4WS 实际上通过在 WS-C 和 WS-T 规范中定义的协调框架(Coordination Framework)和协调协议(Coordination Protocol)来实现复合 Web 服务事务的。WSFL(Web Service Flow Language)是 IBM 提出的一种基于 XML 的 Web 服务复合描述语言,它强调 Web 服务的复合和相应的消息交互序列描述,本身并没有涉及到业务流程的模型。

## 6 事务流程的调度和优化

在 Web 服务环境中,事务流程的调度和优化是 Web 服务事务监控执行中的关键技术。Web 服务事务的调度和优化包括对 Web 服务事务的执行顺序进行检测、验证和优化。其中,检测和验证是为了分析 Web 服务执行顺序流中是否存在不可达节点以及是否存在死锁和活锁,而优化是对 Web 服务执行和调度顺序进行改进,使得优化后的执行效果等同于优化前,而性能得到提高。与之有紧密技术关联的研究是 Web 服务复合的调度优化和事务性过程的调度。

### 6.1 Web 服务复合的分析和优化

在 Web 服务复合的分析和优化方面,较为广泛采用的是并行编译技术。并行编译技术最早出现在 Beinstein 提出的计算并发执行的概念中,随后广泛应用于编程语言、程序验证和优化、并行计算等方面,主要技术有数据依赖性分析、程序约简、操作交换等。一些用于分布计算的并行编译算法,能自动检测和并行化方法调用,如文[15]提出了一种交换性分析算法,能自动并行化对对象的操作。它从对象操作的粒度上进行分析,将某些操作进行交换,并且自动生成并行化代码。并行编译对于数据读写的并发控制要求比较严格,而 Web 服务复合的优化更着重于流程,因此并行编译技术经过一些改进就可应用于 Web 服务的调度优化。例如,文[16]提出了一种分析算法,基于程序依赖图(Program Dependence Graph)将复合的 Web 服务进行代码分离,并重新生成符合原来语义的程序图和新代码,将 Web 服务复合并行化。文[17]提出了一种 Web 服务复合的行为分析(Behavioral Analysis)算法,通过检查 Web 服务复合的各个状态,分析各服务调用以及执行步骤的静态依赖关系,将服务调用并行化,减少网络通信开销,从而提高 Web 服务复合的性能。

### 6.2 事务性过程的调度和优化

在事务性过程的调度方面,文[18]利用了有向图来描述流程,并提出利用图归约方法来证明结构的正确性,文[19]提出了一种基于有序共享锁(Ordered Shared Lock)的动态调度算法 Process Locking,用于事务性过程的并发调度。

**总结与展望** 总的说来,针对 Web 服务事务的研究还比较薄弱。迄今为止,Web 服务还不能在分布式环境里使用事务。Web 服务间的事务、以及对复杂业务流程的支持已经引起人们的关注,这些技术弱化了传统事务模型的 ACID 事务保证。与传统事务相比,Web 服务事务一个最大的不同在于

(下转第 40 页)

- Conf. vol. 1, May 2002. 399~403
- 26 Yeh C-H. IPMA: An interference/power-aware MAC scheme for heterogeneous wireless networks. In: Proc. IEEE Int'l Symposium on Computer Communications, vol. 2, June/July 2003. 849~854
  - 27 Yeh C-H, You T. A power-controlled multiple access scheme for differentiated service and energy efficiency in mobile ad hoc networks and wireless LANs. In: Proc. IEEE PIMRC'03, vol. 1, Sep. 2003. 765~771
  - 28 Yeh C-H. The advance access mechanism for differentiated service, power control, and radio efficiency in ad hoc MAC protocols. In: Proc. IEEE Vehicular Technology Conf. Vol. 3, Oct. 2003. 1652~1657
  - 29 Yeh C-H, Zhou H, Ho P-H, Mouftah H T. A Variable-radius Multichannel MAC protocol for High-Throughput Low-Power Heterogeneous Ad Hoc Networking. In: Proc. IEEE Globecom'03, Dec. 2003. 1284~1289
  - 30 Yeh C-H. High-throughput interference-aware MAC protocols for heterogeneous ad hoc networks and multihop wireless LANs. In: Proc. IEEE Globecom'03, Dec. 2003
  - 31 Tian Hui, Li Ying Yang, Hu JianDong, et al. A MAC protocol supporting multiple traffic over mobile Ad Hoc networks. In: The 57th IEEE Semiannual Vehicular Technology Conf. Vol. 1, Apr. 2003. 665~669
  - 32 Cai Z, Lu M. SNDR: a new medium access control for multi-channel ad hoc networks. In: Proc. IEEE VTC, vol. 2, 2000. 966~971
  - 33 Hsu S-H, Hsu C-C, Lin S-S, Lin F-C. A multi-channel mac protocol using maximal matching for ad hoc networks. In: Proc. 24th Intl. Conf. on Distributed Computing Systems Workshops, March 2004. 505~510
  - 34 郭伟,郑相全,等. 短波抗干扰电台组网技术研究. 电子科技大学通信抗干扰技术国防重点实验室预研基金项目书, 2002

(上接第 16 页)

对补偿而不是回滚的需求。另外一个重要的改变在于对成功情况的定义。“全部或者没有”语义适用于大多数情况,但事务成功的概念已经逐渐变成业务流程的成功,而不是事务所有部分的全部完成。

Web 服务事务的研究,应该着眼于解决 Web 服务事务与传统分布式事务的不同点,提出 Web 服务上事务的相关模型和形式化理论,解决 Web 服务事务监控执行的关键技术问题,提出 Web 服务上事务调度的优化算法,并通过实现事务监控器的软件原型验证所提出模型和算法。此外,目前关于 Web 服务事务的监控执行则缺乏深入的研究,特别是在 Web 服务事务的调度和资源管理方面。Web 服务支持多种通信方式,包括同步调用、异步调用、事件通知等。同时,Web 服务的事务协调本身也是一个 Web 服务。需要给出相应的算法调度 Web 服务事务,重新组合执行顺序,变换通信方式,提高 Web 服务事务执行的并发度和系统吞吐率。这些是 Web 服务事务的研发中应解决的关键问题和未来的研究方向,也是我们正在从事的工作。

### 参 考 文 献

- 1 Curbera F, Nagy W A, Weerawarana S. Web services: Why and how. In: Proc. of the OOPSLA 2001 Workshop on Object-Oriented Web Services. 2001
- 2 Dalal S, Temel S, Little M, Potts M, Webber J. Coordinating Business Transactions on the Web. IEEE Internet Computing, Jan./Feb. 2003
- 3 Cabrera F, Copeland G, Freund T, et al. Specification: Web Services Coordination. Available at <http://www-106.ibm.com/developerworks/library/ws-coor/>, Aug. 2002
- 4 Potts M, et al. Business Transaction Protocol. Available at: <http://www.oasis-open.org/business-transaction/>, Jun 2002
- 5 Cabrera L F, et al. Web Service Transaction (WS-T). Available at <http://www-106.ibm.com/developerworks/library/ws-transpec/>, Aug. 2002
- 6 OMG. OMG's Activity Service specification. Available at: <http://cgi.omg.org/cgi-bin/doc? orbos/2000-06-19>, 2000
- 7 Bunting D, Chapman M, et al. Web Services Composite Application Framework (WS-CAF), Ver1. 0. Available at <http://developers.sun.com/techtopics/webservices/wscaf/>, July 2003
- 8 Gray J, Reuter A. Transaction Processing Concepts and Techniques. Morgan Kaufmann Publishers, 1993
- 9 Pires P F. WebTransact: A Framework for specifying and coordinating reliable web services compositions: [technical report es-578/02]. Federal University of Rio De Janeiro, Brasil. 2002
- 10 Pires P F. WebTransact: A Framework for specifying and coordinating reliable web services compositions: [technical report es-578/02.] Federal University of Rio De Janeiro, Brasil, 2002
- 11 Bennett B, Hahm B, et al. A Distributed Object Oriented Framework to Offer Transactional Support for Long Running Business Processes. In: Proc. IFIP/ACM Intl. Conf. on Distributed Systems Platforms, New York, NY, USA April 2000, Lecture Notes in Computer Science 1795, Springer-Verlag, Berlin, April 2000. 331~348
- 12 Thatte S. XLANG: Web Services for Business Process Design. Available at: <http://www.gotdotnet.com/team/xml-wsspecs/xlang-c/>, 2001
- 13 Curbera F, Golland Y, Klein J, Leymann F, Roller D, Thatte S, Weerawarana S. Business Process Execution Language for Web Services (BPEL4WS). Available at: <http://www-106.ibm.com/developerworks/library/ws-bpel/>, July 2002
- 14 Leymann F. Web Service Flow Language (WSFL1.0). Available at: <http://www-306.ibm.com/software/solutions/web-services/pdf/WSFL.pdf>, May 2001
- 15 Rinard M C, Diniz P C. Commutativity Analysis: A New Analysis Technique for Parallelizing Compilers. ACM Trans. Program. Lang. Syst., 1997, 19(6): 942~991
- 16 Mangala G N, Karnik N M. Synchronization Analysis for Decentralizing Composite Web Services, SAC 2003, 2003. 407~414
- 17 Naoto S, Shin S, Kinichi M. Optimizing Composite Web Services through Parallelization of Service Invocations. ED0C'02, Switzerland, Sep. 2002
- 18 Sadiq W, Orłowska M. Analyzing Process Models using Graph Reduction Techniques. Information Systems, 2000, 25(2): 117~134
- 19 Schuldt H. Process Locking: A Protocol Based on Ordered Shared Locks for the Execution of Transactional Processes. In: Proc. of the ACM Symposium on Principles of Database Systems (PODS'01). Santa Barbara, California, USA, May 2001. 289~300
- 20 Catania N, Kumar P, Murray B. Web Service Event (WS-Events) Version 2. 0. Hewlett-Packard Company, July 2003
- 21 Hewlett-Packard. hp web services transactions 1.0 tech preview. <http://www.hpmiddleware.com/downloads/pdf/wst-spec-sheet.pdf>, 2002
- 22 OMG. OMG's Activity Service specification. Available at <http://cgi.omg.org/cgi-bin/doc? orbos/2000-06-19>, 2000
- 23 Sun. JSR 95: J2EE™ Activity Service for Extended Transactions. Available at <http://www.jcp.org/en/jsr/detail?id=95>, Aug. 2003
- 24 Sun. JSR 156: XML Transactioning API for Java (JAXTX). Available at <http://jcp.org/en/jsr/detail?id=156>
- 25 Mohan C. Tutorial: Advanced Transaction Models Survey and Critique. In ACM SIGMOD Intl. Conf. on Management of Data, Minneapolis, May 1994
- 26 Weikum G. Principles and Realization Strategies of Multi-Level Transaction Management. In ACM Transactions on Database Systems, March 1991
- 27 Garcia-Molina H, Salem K. SAGAS. In: Proc. of ACM SIGMOD Conf. on Management of Data, 1987. 249~259
- 28 Kaiser G E, Pu C. Dynamic Restructuring of Transactions. In Database Transaction Models for Advanced Applications, Aug. 1991
- 29 Bukhres O, Elmagarmid A, Kuhn E. Implementation of the Flex Transaction Model. In IEEE Data Engineering, 1993, 16(2): 28~32