

基于消息中间件的持久状态服务扩展模型

曾 一 梁宝兰 刘元勇

(重庆大学计算机学院 重庆 400044)

摘 要 本文提出了一个基于消息中间件的持久状态服务的扩展模型。与原持久状态服务不同的是本扩展模型增加了缓存管理,而且它是建立在基于内容订阅方式的消息中间件之上的。它不但移去了原持久状态服务频繁访问存储器而产生的瓶颈问题,而且以一种松散耦合的方式实现了在分布式信息系统中的持久对象数据的共享。

关键词 持久状态服务,扩展模型,消息中间件,分布式信息系统

An Extended Model of Persistent State Service Based on Message-oriented Middleware

ZENG Yi LIANG Bao-Lan LIU Yuan-Yong

(College of Computer Science, Chongqing University, Chongqing 400044)

Abstract This paper introduces an extended model of persistent state service based on message-oriented middleware. Distinguishing from original persistent state service, this extended model provides caching mechanism and also is supported by message-oriented middleware. This model not only removes the bottleneck at datastore side caused by frequent accesses, but also achieves a relax-coupling manner method of sharing persistent objects in distributed information system.

Keywords Persistent state service, Extended model, MOM, Distributed information system

1 引言

目前,面向对象技术已经广泛地应用系统建模和编写程序当中。在许多面向对象的程序中,特别是在当前迅速增长的分布式信息系统中,有很多对象的生命周期并不是随着进程的结束而消失的,这就使得保持对象状态的持久化越来越重要。特别是在大规模的分布式信息系统,例如拍卖系统、预定系统、交通信息等系统中,其数据分布在世界各地,并要求能实现数据的共享,而且其用户需要监视它所感兴趣的信息的变化情况。为解决对象持久状态特别是分布式计算环境下的对象持久化的问题,对象管理组织(OMG)在改进持久对象服务 POS(Persistent Object Service)的基础上提出了持久状态服务 PSS(Persistent State Service)^[1]。本文在 PSS 的基础上提出一个适合分布式信息系统的高速且稳定持久状态服务扩展模型,该模型可支持分布式信息系统中的持久对象共享的实现。

2 PSS 存在的两个缺陷

PSS 规范的主要目标是实现 CORBA 中伺服对象(servant)的持久化,为用户提供统一的接口来透明地访问异构存储系统,并使得 CORBA 变得更加友好。在持久状态服务应用中,用户使用 PSDL(Persistent State Definition Language)编写文件、定义存储对象类型和存储对象 home、PSDL 编译器对这些 PSDL 文件进行编译处理、生成目标语言的文件,实现对存储状态的管理,使开发者不必关心访问存储系统的细节。为了操纵一个存储对象,开发者可以使用一个具体编程语言的实体对象调用存储对象实例。一个存储对象实例可以

连接数据存储器其中的一个存储对象,从而可以直接访该存储对象的状态。

依靠关系数据库的 PSS 的实现是基于查询装载的,即 PSS 必须要建立与服务器端数据存储器的连接,然后在服务器端执行查询操作,并把结果返回给请求者。这种实现方式因为每次请求持久对象就要执行一次 SQL 查询操作,造成了存储器数据交换的瓶颈问题。再者,在一些信息分发系统当中,用户会要求不间断地监视感兴趣的信息,如果使用轮询法来刷新信息的变化,那么就会造成后端数据存储系统的负荷过载,浪费网络资源和刷新的延缓。

为了克服 PSS 的这两大缺陷,本文提出了一个基于消息中间件(Message-Oriented Middleware, MOM)的 PSS 扩展模型,来提高 PSS 的速度以及稳定性,并实现分布式信息系统环境下的信息共享。

3 基于消息中间件的 PSS 扩展模型

3.1 分布式信息系统结构图

在分布式信息系统当中,每个连接在消息总线上的服务器均有它所属的一个或多个客户,且有一个本地数据存储器^[2],如图 1 所示。客户可以通过 CORBA 访问它所属的服务器上的持久对象,而服务器之间则可以通过消息总线相互访问对方的持久对象数据,以提高系统信息的共享性。

扩展的持久状态服务不但可以完成原持久状态服务的所有任务,而且通过加上缓存管理来提高应用访问持久对象的速度;再者通过加上消息中间件的支持,可以灵活、方便地实现不同服务器上的数据共享。本扩展的持久状态服务还可以与对象事务服务(Object Transaction Service)相结合,实现高

曾 一 副教授,主要研究方向:软件工程。梁宝兰 硕士研究生,研究方向:软件开发工具与环境。刘元勇 硕士研究生,研究方向:软件开发工具与环境。

速、稳定的持久状态服务。

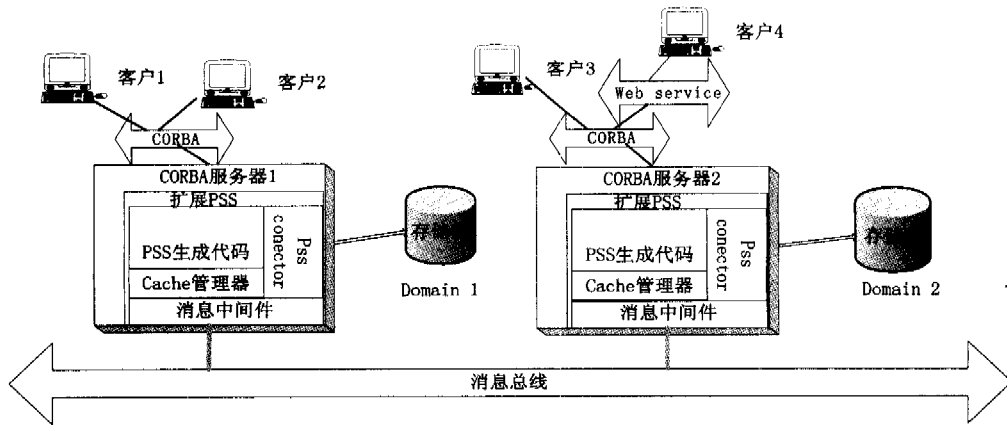


图1 系统结构图

3.2 缓存管理

在原持久状态服务当中，每次请求持久对象时都要从存储器中恢复持久对象，造成了存储器数据交换的瓶颈问题。为了提高应用访问持久对象的状态信息的速度，在扩展的持

久状态服务中将缓存管理作为对象持久化的扩展功能之一。缓存管理器把使用最为频繁或最近使用的本地或异地数据对象建立副本，以提高系统访问持久对象的速度^[3,4]。

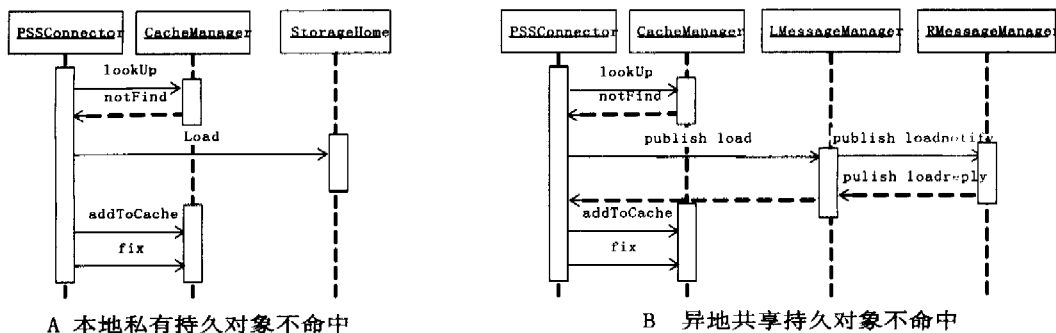


图2 缓存管理时序图

图2给出了该扩展模型中缓存管理控制的时序图。在分布式信息系统中的持久对象可以分为两类：私有持久对象（只供本地服务器访问）和共享持久对象（可供异地服务器访问）。在使用持久状态服务的开始时创建了一个PSSConnector，并由它来调用缓存管理器的lookUp(查找)方法，再根据给持久对象的类型以及命中与否来采取相应的动作。

1)若目标持久对象为私有持久对象且命中，则缓存管理仅仅使用fix(修复)方法将这个数据对象修复，然后返回给PSSConnector。

2)若目标持久对象为私有持久对象且不命中，则PSSConnector把请求转给目标对象所属的StorageHome，并由它从存储器中查找并装载该持久对象数据，然后再把该持久对象数据加入缓存。

3)若目标持久对象为共享持久对象且命中，则缓存管理使用fix(修复)方法将这个数据对象修复，并通过消息中间件发送装载通知。

4)若目标持久对象为共享持久对象且不命中，则PSSConnector就可以通消息管中间件向目标服务器发送装载请求，目标服务器回复源服务器时携带该持久对象的状态信息然后返回给PSSConnector，并发布装载通知，源服务器收到回复后恢复目标持久对象，并把它加入缓存。

3.3 基于出版/订阅模式的消息中间件

在PSS服务当中加入消息中间件主要是实现分布式信

息系统下信息共享，并对共享的信息进行一致性维护。出版/订阅模式为信息的发送者与消息的接受者之间提供了一种松散耦合的通信方式，这种方式具有极大的信息传递灵活性和系统的可扩展性。

3.3.1 基于主题和基于内容的订阅模式

在发布/订阅模式中，有两种订阅方式，一种是传统的基于主题的订阅，另一种是基于内容的订阅^[5]。在基于主题的订阅方式中，发布者发布信息时为信息赋予一个主题，用来描述信息的主要内容，并根据订阅者所订阅的感兴趣的主题来发送符合订阅者要求的主题信息发送给订阅者。

基于内容的订阅则可以为订阅者提供更大的信息订阅灵活性，订阅者可以向发布者提供对信息内容的要求，只有内容符合条件的信息才会发送给订阅者。XML具有极其灵活的语法、良好的可扩展性和自描述性，能有效地构造和标识它所包含的信息，这就为灵活地发现、抽取、排序、过滤、整理以及管理信息提供了可能。

3.3.2 消息形式

下面以拍卖系统发布拍卖物品的信息为例，对比两种订阅方式。

1) 基于主题的订阅模式

- 物品1:主题: 艺术品/绘画
内容: 达芬奇, 蒙娜丽莎, 1513,
- 物品2:主题: 艺术品/绘画
内容: 毕加索, 格尔尼卡, 1937,

i201,在创建时初始化了 item i002 的基本信息,例如标题、分类、所属者、底价等。假如 item i201 是属于达芬奇的画,而 A 与 C 都对画感兴趣,但基于内容的订阅方式可以让 A 只订阅

齐白石的画,而 C 只订阅达芬奇的画,所以 US 服务器创建了 i002 后只向 France 服务器发送创建通知。

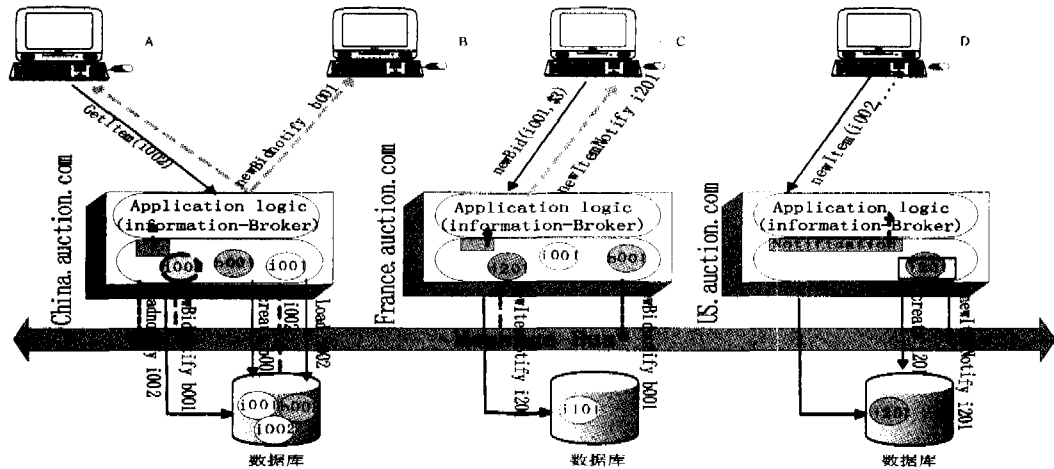


图 4 扩展持久状态服务运行范例

总结 本扩展模型在原持久状态服务模型的基础上加上了缓存管理以及基于内容订阅模式的消息中间件。通过加上缓存管理,可大大加快扩展的持久状态服务访问持久对象的速度;通过加入基于内容订阅模式的消息中间件,可实现分布式信息系统中服务器间的持久对象共享,而且在消息中间件上发布与订阅时使用 XML 来描述持久对象的状态信息可提高系统的灵活性和可扩展性。

参考文献

1 OMG. Persistent State Service Specification. Object Management Group. <http://www.omg.org/cgi-bin/doc?ptc/00-12-07>, January 2000

2 Liebig C, Cilia M, Betz M, et al. A publish/subscribe CORBA Persistent State Service Prototype. In: Proc. of the IFIP/ACM International Conference on Distributed Systems Platforms, 2000. 231~255
 3 Garcia-Banuelos L, Duong P, Nedelec T, et al. A Component-based Infrastructure for Customized Persistent Object Management. In: Database and Expert Systems Applications, Proc. 14th Intl. Workshop on, 2003. 536~541
 4 曾一,徐珂. 基于组件架构的对象持久层框架设计. 计算机科学, 2005(12)
 5 刘振涛,熊璋,欧阳元新. 基于内容的订阅方法及其在 MQSeries 中的实现. 计算机工程, 2004, 30(16)

(上接第 292 页)

$p_{10} = \{t1, t3, t4, t9\}$, $p_{11} = \{t1, t3, t4, t6, t9\}$, $p_{12} = \{t1, t3, t4, t7, t9\}$,
 $p_{13} = \{t1, t4, t8, t5\}$, $p_{14} = \{t1, t4, t6, t8, t5\}$, $p_{15} = \{t1, t4, t7, t8, t5\}$
 $p_{16} = \{t1, t2, t4, t8, t5\}$, $p_{17} = \{t1, t2, t4, t6, t8, t5\}$, $p_{18} = \{t1, t2, t4, t7, t8, t5\}$
 $p_{19} = \{t1, t3, t4, t8, t5\}$, $p_{16} = \{t1, t3, t4, t6, t8, t5\}$, $p_{17} = \{t1, t3, t4, t7, t8, t5\}$

我们可以从每一条转换路径得到一个测试用例,该测试用例用来检查目标程序是否存在这样一条路径。

总结和进一步的工作 本文提出的基于 UML 模型的测试用例的生成方法利用 UML 状态图,为测试用例生成提供信息。该方法的特点是:①完全基于 UML;②生成较少的测试用例;③便于实现自动化。

对基于 UML 生成测试用例的研究仍存在很多不足。目前对基于 UML 生成测试用例的研究,主要是以单个模型图作为研究对象;基于设计描述的形式化测试准则还较少;对 UML 模型图做进一步精化或扩展其语义后,使用常规方法进行测试用例生成的情况较多,直接使用 UML 模型图提供的信息的方法较少;在具体的方法研究中,假设的前提和约束太多,还不能够达到实用的程度。因此,对基于 UML 生成测试用例的深入研究将是我们未来工作的重点。主要有以下几个目标:

第一,我们可以将状态图作为多个交互类集成测试的基

础,在状态图上定义测试标准,生成的测试规格说明书必须满足这些准则并在数据库中保存状态转换描述、控制、数据流等信息。

第二,深入研究在生成针对某一层测试的测试用例时,综合利用待测试系统的各种模型图的测试用例生成方法。

第三,提出一种与面向对象软件开发过程集成的测试过程,设计出能够实现面向对象软件测试活动中测试用例生成、测试执行和结果评价这三个过程的软件自动化测试框架。

参考文献

1 Mosley D J, Posey B A. 软件测试自动化. 邓波,黄丽娟,曹青春,等译. 北京:机械工业出版社, 2003
 2 Barbey S, Strohmeier A. The problematics of testing object-oriented software. In: SQM'94 Second Conference on Software Quality Management, 1994, 2. 411~426
 3 Object Management Group. UML Specification V1. 5, 2003. <http://www.omg.org/cgi-bin/apps/doc?formal/03-03-01.pdf>
 4 Gallagher L, Offutt J. Integration Testing of Object-oriented Components Using FSMS: Theory and Experimental Details; [Technical Report]. CMU/ISE-TR-04-04, 2004
 5 Binder R. 面向对象系统的测试. 华庆一,王斌君,陈莉,译. 北京:人民邮电出版社, 2001
 6 Offutt J, Abdurazik A. Generating Tests from UML Specifications. In: Proceedings of the 2nd International on Unified Modelling Language. Beyond the Standard(UML'99), 1999