

基于 CORBA 的多数据库系统互操作技术*

CORBA-Based Multidatabase System Interoperability Technique

肖明 胡金柱

肖毅

(华中师范大学计算机科学系 武汉 430079) (华中师大电子与计算机研究所 武汉 430079)

Abstracts In this paper, multidatabase system heterogeneity, object granularity, client request method, call mode, activation policy, object transaction services, concurrency services and query service are analyzed. By building CORBA-based interoperability architecture, multidatabase system well interoperability is implemented.

Keywords Multidatabase system, Interoperability, CORBA

1 引言

当前,随着多数据库系统中成员数据库数目的不断增大,需要提供一个世界范围内的互操作环境,以实现大量自治和异构的成员数据库之间文件系统、电子数据表等格式中存储的海量数据的信息共享。作为多数据库系统的一项关键技术的互操作性已成为当今国际学术界的一个研究热点,本文主要介绍基于 CORBA 的多数据库系统互操作技术。

2 CORBA 标准

OMG (Object Management Group) 推出的 CORBA 主要包括对象管理结构 OMA 和对象请求代理 ORB, ORB 为客户请求和目标对象实现之间提供透明的通信机制。在这一模型中,客户向 ORB 发出请求执行某一服务, ORB 负责找到服务器并实现,并将结果返回客户方,从而客户所处环境由分布式异构环境变成虚拟局部环境。

所有 CORBA 客户知道目标对象是其对象访问和接口,一个对象访问应属于对象工厂创建的对象,一个对象工厂本身就是一个 CORBA 对象,客户可以从对象工厂本身存取对象访问或使用名字服务器来查找。接口由 IDL 定义, IDL 接口是客户存取操作、概念和类型属性的集合,可用 IDL 或 C、C++、Java、Smalltalk 及 Ada 等语言两种方式编写。

ORB 的对象适配器 (Object Adapter) 提供对象实现存取服务,每个接口与 ORB 连接,允许 ORB 运用定位、激活、调用操作每个 ORB 对象。直到目前,仅定义

了基本对象适配器 (BOA), BOA 设计适用于大部分的对象实现,提供生成并解释对象访问,方法调用,实现的登记、激活和撤消,选择规则对象实现对于给定的对象访问和证实。最近,OMG 公布了一个新标准持久对象适配器 POA (Persistent Object Adapter),从而为 OODBMS 提供一些类似 ORB 服务,如对象访问生成和管理。

公共对象服务 (Common Object Services): OMA 接口主要包括以下三组:对象服务接口,公共设施接口和应用对象接口,对象服务提供利用 ORB 实现基本对象功能。每个对象服务具有准确定义的接口和功能语义,并具有正交关系,正交性就是允许对象同时使用几个对象服务,而不会产生任何冲突。有些标准接口具有即插即用和重用性。例如,当一个客户移动一些对象时,可以通过标准接口以支持生命周期服务。如果对象不支持标准生命周期服务,那么用户需要知道对象的“移动语义”及其响应接口。

公共设施 (Common Facilities): 由部件组成。提供在 CORBA 环境中应用对象的实用开发服务。标识两类设施:水平设施,垂直设施。水平设施由全部应用对象的使用设施组成。例如:用户接口,系统管理和任务管理。垂直设施是专门用于选定的应用域的部件。如运输,制造,电子商务或电信部门。

3 基于 CORBA 的多数据库系统技术

CORBA 的分布式体系结构将一个分布式异构系统作为相互作用的对象集合,即将分布式系统中各成员系统的资源模型化为对象,而成员系统提供的服务

*)湖北省自然科学基金资助项目。

被模型化为对象方法,这些方法组成对象接口。这样,在每个成员系统上可以定义一个服务接口,成员系统为这些服务提供实现。客户通过以公共语言表示的请求与异构系统进行交互,对象请求代理机制负责转换客户请求到可用服务、传递请求到适当系统,提供以公共语言表示的应答给客户。

实现基于 CORBA 的多数据库系统的基础是,将参加集成的成员数据库通过对象包装注册到 ORB 总线上,在注册各种数据库到 ORB 时,有如下一些关键的实现策略。

- 在多数据库系统的实现中,主要需要解决四个层次的异构性问题:

- 平台层:主要包括硬件和操作系统的异构性;

- 通信层:主要是通信协议的不同,CORBA 提供了位置透明性,允许客户通过对象的标识符存取对象,而与客户和对象之间的位置和通信协议无关,从而解决了通信层的异构问题;

- 数据库系统层:主要体现在数据模型、查询语言、事务管理、并发控制机制以及查询优化方法等方面的异构性;

- 语义层:语义冲突是由于大多数数据库是各公司独立开发的,其中包括模式冲突、数据冲突和行为方法冲突。

后两层在 DOM 平台上多数据库系统中的异构性问题,可以通过开发一个包括全局查询管理、全局事务管理和模式集成的全局层来解决,其模式集成技术是今后研究的一个热点。

- 由于 CORBA 提供了互操作的面向对象框架,建立一个基于 CORBA 的分布式对象管理体系结构之上的多数据库系统,使用 IDL 定义对象及实现,因此,基本设计问题是确定对象的粒度。在通过 CORBA 去注册 DBMS 时,对象可以是 RDBMS 中的一行,OODBMS 中的一个对象、一组对象或整个数据库。对于小粒度对象,如一个表,所有处理这些表的 DBMS 机制,如查询处理、事务控制等,都必须由集成的多数据库系统提供支持。当一个关系库作为一个对象注册时,所有的处理仍由响应的 DBMS 完成,不需由多数据库系统提供。对象粒度能影响部分 ORB 的性能,ORB 实际仅支持 BOA,类的插入和删除需要重新编译 IDL 代码和重建服务器。如果对象粒度不适合,将引起 ORB 较大开销,如果 OODBMS 适配器允许输出合适的 OODBMS 粒度对象,通过 OODBMS 的相互协调去处理各自的对象,这一适配器能解除 ORB 上的开销。另一个解决方案是使用动态服务器/骨架接口,此接口最初用于 inter-ORB 互操作,提供动态增加新对象类型、代码重编译和服务器重建。

• 34 •

- CORBA 提供二种客户请求到服务器的方式:一个接口对应一个实现,一个接口对应多个实现之一,

一个接口对应多个实现。方式 1 中,接口的操作与实现的方法之间是一一对一关系,它实现简单,但由于接口的每个操作都必须被实现的一个方法所支持,操作和方法必须是严格一对一的匹配关系,因此缺乏灵活性。方式 2 中,虽然接口的操作与实现的方法之间也是一对一关系,但同一接口可以有多种实现,这就提供了相当的灵活性,以便为接口中的一个操作定义多个不同的方法。方式 3 中,同一接口可以包括多种实现,每个实现只完成接口的一部分功能。

- CORBA 定义了三种调用模式:同步、延时同步和单向。在同步模式中,客户等待请求操作完成,限制了客户之间的多对象并行事务操作。在延时同步模式中,客户在服务器选定后继续执行,并接受服务器的返回结果直到操作执行完毕。在单向模式中,客户发送请求不必接收应答。CORBA 不支持异步模式,是由于客户接收异步消息,此时它也作为服务器。换句话说,操作的异步模式能完成在两个 CORBA 对象之间相互发送单向请求。这种层到层(peer-to-peer)应用的唯一缺点是增加了客户代码的复杂性,对于多数据库系统中的对象,同步调用模式可以充分满足,延时同步模式或层到层应用主要用于并行执行中,例如:为了提供并行查询,多数据库的全局查询管理程序将不等待查询完毕,就通过 SendQuery 方法向局部 DBMS 提交查询。因此,SendQuery 方法不在同步模式中调用。

- CORBA 的对象类型允许两种调用方式:动态方式和存根方式。以存根方式调用,客户使用的代码存根在运行时不能被改变,它的实现很简单。以动态方式调用,客户可在运行中定义和建立请求,它为用户提供了更大的灵活性。若在实现中所有的对象是已知的,且客户使用的接口不会经常改变,一般采用存根式调用。图 1 所示为数据库对象中的一次调用操作。

- 当向 CORBA 注册对象时,需要规范每种对象的激活策略。接口中的操作与实现的方法之间存在一对一的关系。激活策略是指如何在服务器上启动每个对象实现。可以使用 4 种激活策略:1)共享式:一个服务器可以同时支持多个对象实现的执行;2)非共享式:一个服务器同时只能支持一个对象实现的执行;3)单个服务器方法式(server-per-method):每个方法的执行都要启动一个新的服务器;4)持久式:服务器不是自动启动的。启动后,其运行与共享式相同。在多数据库系统中有些对象需要并发激活,可通过服务器中的线程运用共享激活方法,或者在对象间非共享模式下激发的隔离服务器来完成。因为一个服务器一次只能给一个对象提供服务,因而对于同一服务器控制的对象

客户与其它客户间的请求,必须等到当前请求完成之后才可进行,若服务器在对象生存周期中为其保存临时数据,那么,对一个对象的所有请求必须由同一服务器服务。例如,在共享模式下激活一个全局事务管理程

序,保持不同线程中事务处理的上下文联系就很有必要。但是,若全局事务管理程序在非共享模式下激活的,那么,实现此功能只需对每个激活的事务采用一个进程即可。

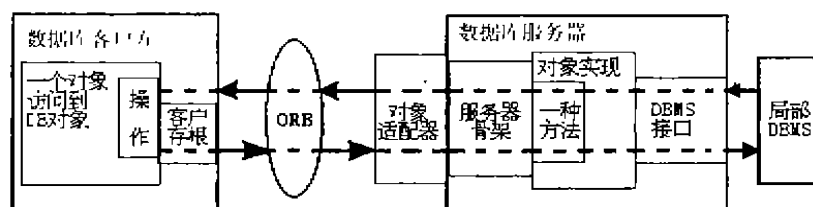


图1 运用 ORB 为数据库对象中的一次调用操作

CORBA 通过提供一个互操作基本结构来处理平台与通信层的异构性。因此,一个基于 CORBA 的数据库系统设计主要针对系统的上层,如模式集成,全局查询处理和全局事务管理程序。这明显降低了多数据库系统设计与实现的复杂性。由于 CORBA 与 COSS 一起为管理分布式对象提供了基本数据库功能,COSS 中所包含的数据库相关服务如对象事务处理服务,备份及恢复服务,并发服务和查询服务。若这些服务可在 ORB 中实现,就可以通过相关对象服务的标准接口来开发 CORBA 上多数据库系统的全局层。例如通过使用一个对象事务处理服务,实现针对相关 DBMS 的对象事务处理服务规范中所定义的接口,从而实现全局事务管理。

对象事务处理服务(OTS)规范描述了一种支持基于 CORBA 结构的分布式异构环境下的单一或嵌套事务处理。OTS 允许多个分布式对象共同提供原子事务处理和恢复机制。为实现这一功能,OMG 定义了一些任务,如事务处理客户、事务处理对象,可恢复对象和事务处理上下文。

在一个典型应用中,事务处理客户启动一个事务处理,获得一个 ORB 提供的控制对象,并为每个控制对象生成一个事务处理线程。一个事务处理线程包含控制、协调事务处理的信息,控制对象用于获得终止程序和协调程序对象。事务处理客户用终止程序来停止或提交事务处理。协调程序提供一个嵌入两阶段(2PC)协议的事务处理对象接口。事务处理对象有些具有可恢复性,称为可恢复对象。可恢复对象运用协调程序对象将一个资源对象寄存到 ORB 中,由 ORB 来提交或终止资源对象。由此,ORB 具有分布式原子事务处理功能。

并发控制服务(CCS)协调多个客户的资源共享。CCS 并不定义什么是资源,资源定义及正确识别是由

客户完成的。实际应用中,一个对象就可能是一种资源,对象实现是由 CCS 来协调对对象的并行存取。CCS 设计为通过 CORBA 对象事务处理服务来协调并发事务处理行为。CCS 使用锁机制来协调资源的并行、有五种锁模式:意向(intention)读、读、更新、意向写和写,一个客户可同时持有相同资源中的多个锁。一组锁和一个资源构成一个锁集。若某对象是一种资源,将在内部建立一个锁集。为了能释放事务处理所持有的锁,CCS 定义了一种锁协调程序,能够保留一种资源中的多个锁,从而使得处理带有嵌套事务的锁更为方便。如一个子事务处理可获得一个由其上一级事务处理锁定的资源中的锁,然后释放此锁,从而避免其上一级事务处理遗漏此锁。此方法功能上等价于嵌套事务处理的锁代理(delegation)。

对象查询服务提供对象集上的基于谓词的查询操作。为了提供在查询系统中最大范围的查询互操作性,OQS 支持 SQL 查询和 OQL 两种方式。

4 基于 CORBA 的多数据库系统实现

国际上,对于建立在 CORBA 基础上的多数据库系统的开发也非常重视,比较成功的系统有土耳其中东工业大学和 Alberta 大学合作完成的 MIND (METU-Interoperable) 系统。该系统在 DEC 公司的 CORBA 产品 ObjectBroker 和 COSS 软件之上,开发出了具有良好互操作性的多库系统,实现了对关系数据库和对象数据库的互操作。

在 MIND 中,将组合起来的局部 DBMS 封装在一种类属数据库对象中,MIND 定义了 CORBA IDL 中类属数据库对象的接口,并提供多种实现。一种是针对局部 DBMS 中的每个成员,称为局部数据库 Agent (LDA)。目前支持包括 Oracle 7, Sybase, Adabas D 和 MOOD (METU OO Database System)。LDA 对象用

于在规范数据模型中局部 DBMS 的输出模式,并可全局查询语言转化为局部查询语言。这一层实质上提供了一个同构的数据对象集合。MIND 的全局层包含一个全局事务管理程序(GTM),一个全局查询处理程序和一个模式集成器,GTM 跟踪子事务处理,对 LDA 对象用 2PC 协议进行全局提交或全局终止,并检测全局死锁。

全局查询处理程序负责分析和分解查询,根据模式集成服务及全局查询优化中获得的信息,分解全局查询,并将全局子查询送往相应的 LDA 对象。在全局查询处理中,需要进行查询优化,通常采用基于统计推理的优化技术,包括减少场地的 IO 开销,减少查询的通信量,减少场地间的负载不平衡,并强调利用子查询的并行性。

局部数据库代理对象负责向局部 DBMS 交付操作,与全局事务管理对象一起共同完成全局事务的原子提交。各个局部数据库代理对象并行地执行全局子查询。另外,全局查询处理还负责处理从局部数据库代理对象返回的局部结果,进行过滤和合并,生成最终的全局查询结果。

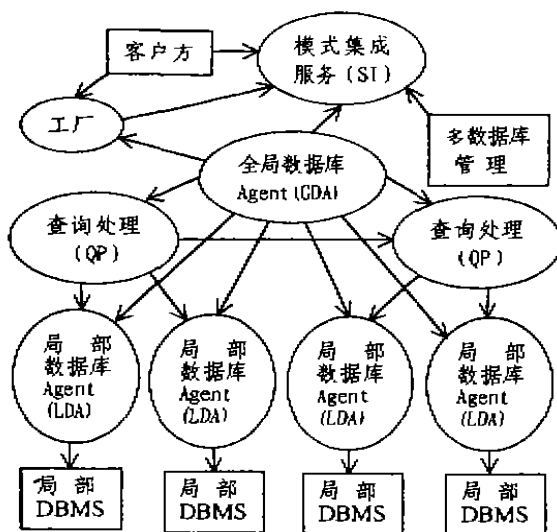


图 2 MIND 系统的体系结构

模式集成服务控制全局模式信息,其输出模式的集成是通过使用一种基于 CORBA 接口定义语言的对象定义语言(ODL)来实现的。MIND 还提供了一个公共数据模型和一个基于 SQL 的单一全局查询语言。

图 2 为 MIND 系统的体系结构。一个 MIND 客户仅知道全局数据库 Agent,工厂及模式集成服务对象的接口定义,客户运用 ORB 所提供的命令服务来发现工厂的对象访问,然后向工厂中产生的 GDA 对象发出请求,GDA 封装了 CORBA 对象中的全局事务管理程序和全局查询管理程序。客户还可从模式集成服务中获得可用资源的名字,从命名服务中获得模式集成服务的对象访问。

结论 CORBA 为设计和实现分布式多数据库集成系统提供了非常好的方法和中间件功能,并且大大减轻了开发工作量,使系统具有良好的开放性、可扩展性和可伸缩性。它不仅有助于集成结构化的数据库系统,还可以集成非结构化的文件系统、快速表系统、 workflow 系统等。

随着基于 Internet/Intranet 应用系统的开发。CORBA 2.2 中,异构的应用实体通过 ORB 相互协作,不同厂商开发的 ORB 可通过一组协议进行互操作,其中包括: IIOP (Internet Inter-ORB Protocol), GIOP (General Inter-ORB Protocol), ESIOP (Environment Inter-ORB Protocol),通过这组协议或一个桥,能将一个 ORB 转换到另一个。总之,基于 CORBA 标准的多数据库系统技术有待进一步深入研究和开发,并有着广泛的应用前景。

参考文献

- 1 Dogac A, Dengi C, Kilic E. Distributed object computing platforms. *Communication of ACM*, 1998, 41(9): 95~103
- 2 Dogac A, Dengi C, Kilic E. METU Interoperable Database System Available at: <http://ftp.srdc.metu.edu.tr/pub/mind/papers/mind.ps>, 1996
- 3 The Common Object Request Broker Architecture and Specification, Revision 2.2, February 1998
- 4 李贵,尹朝万,郑怀远. 大规模多数据库系统的互操作机制. *计算机科学*, 1998, 25(1): 36~39