

分布式系统

CORBA

操作

计算机知识 (4)

13-16

# 使用 CORBA 实现各种分布式应用系统的策略

The Strategies of Implementation of Distributed Application Systems by Using CORBA

詹永照 谢立 孙钟秀

(南京大学计算机科学系 南京 210093)

TP393

**Abstract** Building highly available applications and real-time applications is very hard by using conventional implementation of CORBA. In this paper, we mainly introduce the structures of CORBA systems supporting highly reliable applications and real-time applications, discuss the important problems to be solved, and finally consider the strategy of developing distributed applications by using Java Applets and CORBA.

**Keywords** Distributed object computing, CORBA, Reliability, Real time

## 1 引言

随着计算机网络应用的不断深入和普及,大规模的计算机网络将不断增加。在这种计算机网络中,不仅硬件设备型号、种类、规模相异,而且操作系统平台、程序设计环境及应用也各不相同,这就是大规模计算机网络的重要特征——异构性。人们迫切希望在这种计算机网络上共享信息、设备和软件,这就要求支撑这种应用的分布式系统具有良好的互操作性、可迁移性和可重用性,集成各组件产生一个连贯的可操作的异构系统。然而,处理分布计算环境中的异构性是很不容易的,特别是开发支撑并有效地利用异构的网络系统的应用软件和组件是非常棘手的。目前有许多程序设计接口和软件包能减轻单一异构平台的软件开发的负担,但能够处理分布异构环境下分别开发的系统的集成软件则很少。由于认识到这些问题,1989年成立的对象管理组(OMG)对研制、采纳和促进分布异构环境应用开发的标准进行了不懈的努力,提出了对象管理结构 OMA 和主要组件——公用对象请求中介 CORBA<sup>[1,2]</sup>。CORBA 是应分布对象计算技术的出现而提出的分布对象管理系统结构标准,它充分体现了分布对象计算系统的异构性、互操作性、可迁移性和可重用性,使用该能使程序设计者免去创建和管理分布式应用中许多乏味的、易出错的和不易迁移的细节,对于快速研制、开发坚定的各种复杂分布式应用系统具有广泛的应用前景。目前一些基于 CORBA 的分布

对象环境的产品也应运而生,如 IONA 的 Orbix, SunDOE 的 DOMF, HP 的 ORBPlus, IBM 的 DSOM, DEC 的 Object Broker, Expertsoft 的 PowerBroker 等。许多组织已使用或正在使用 CORBA 标准及产品研制各种分布异构环境应用系统。然而,由于 CORBA 标准和 CORBA 的常规实现没有直接论述象实时质量服务或高速性能、组通讯、部分失效和事件因果序等与分布计算有关的复杂的问题<sup>[3]</sup>,因而常规的 CORBA 实现很难建立高有效性的应用,更不能有效地支持实时应用。因此本文主要介绍如何对常规的 CORBA 实现作些扩充和优化,使之支持高有效性和实时性分布应用。同时由于互连网的普及和 Java 的流行,也对 Java 和 CORBA 的结合开发互连网上的分布计算作个简介。

## 2 CORBA 在可靠的分布式系统中的应用策略

### 2.1 可靠系统的模型

如果一个分布式系统尽管存在部分失效性、异步性和系统执行时的重构性但它的行为均是可预知的,则称该分布式系统是可靠的。可靠的应用应该是高度有效的,即尽管计算节点、软件对象和通讯链可能失效,但该应用仍能提供它的基本服务。好的通讯系统能使分布式应用比集中式应用更加可靠、更加模块化和更加可扩充。通常有三种模型有助于建立可靠系统,它们是消息队列 MQ、事务处理监视器 TPM 和虚拟同步 VS,每一模型代表一定的分布计

算的观点并且各有优缺点。

(1)消息队列 MQ。是解决进程间可靠通讯较简单的方法。进程 S 若希望可靠地把消息 M 传递给进程 R, 它可先提交消息给它本地的消息队列 MQ 管理者, S 的 MQ 管理者把消息的内容写到不易变的存储中以免失败时丢失, 一旦进程 S 提交了消息后, 就解除了有关传输消息 M 的任何活动, 而由 S 的 MQ 管理者代理存储和传输。S 的 MQ 管理者试图把消息传给 R 的 MQ 管理者, 若 R 的 MQ 管理者由于处于等待状态、现场失效或网络划分而无效时, S 的 MQ 管理者将周期性地传送消息直到 R 的 MQ 管理者有效。MQ 模型适合于异步单向通讯的应用, 其优点是易使用、实现和理解, 缺点是必须提供可恢复的信息存储器 and 缺乏双向通讯的能力。

(2)事务处理监视器 TPM。允许一个分布的 Client 应用用 Begin/End 的事务标记把一系列调用括起来。如果一个服务在执行中失败, TPM 将把事务处理中发出的调用退回。TPM 主要针对面向数据的应用, 这些应用是管理分布的持久的数据对象。TPM 也易于理解, 但主要局限是难以进行程序设计和分析实际执行开销及在许多情形下过分串行化。

(3)虚拟同步 VS。是一个比 MQ 和 TPM 机制更低一层次及更基础的分布式执行模型。模型的核心是失效猜测服务和组抽象。失效猜测服务检测有缺对象并保证非有缺对象对那些有缺对象认识的一致性。失效猜测器依靠“超时”去检测被怀疑的对象, 它仅检测崩溃失效。在 VS 中高有效性是由进程备份来保证的, 为此 VS 提供了组抽象和一个可靠的组播机制。高可靠的对象可在多个通讯节点建立, 然后把实例加入到对象组。对象组抽象允许程序员把一个单一的对象引用分配给实现同样接口的一系列网络对象。VS 实现了向前推进机制, 该机制是使崩溃的对象被重新启动和重新加入到它的对象组中。VS 主要适合于需要维持一个分布的易变状态的应用, 不足之处是低层次的程序设计接口和许多功能的不兼容性。

## 2.2 可靠系统 CORBA 结构

Silvano Maffei 提出了一个可靠系统的扩充 CORBA 结构, 该结构综合了消息队列、事务处理监视器和虚拟同步等三种可靠系统模型<sup>[4]</sup>, 其结构如图 1 所示。在该模型中, 异步应用可采用消息队列方法而不必花费由于事务监视而带来的开销; 紧匹配的面向进程的应用可在虚拟同步通讯子系统上高效地运行; 面向数据的应用可使用事务处理监视器去

协调存取分布式数据对象, 其中 ORB 是在一个虚拟通讯子系统像 Horus、Isis、Totem 或 Transis 上运行的, 这个增强的 ORB 支持对象组抽象, 即相同类型的 CORBA 对象可以一个单一实体命名和存取, 对象组允许 CORBA 对象执行时备份和允许 CORBA 请求高效地组播。事务处理监视器和消息队列管理者是在 ORB 上以插入式 (Plug-in) OMG 公用对象服务形式提供的。

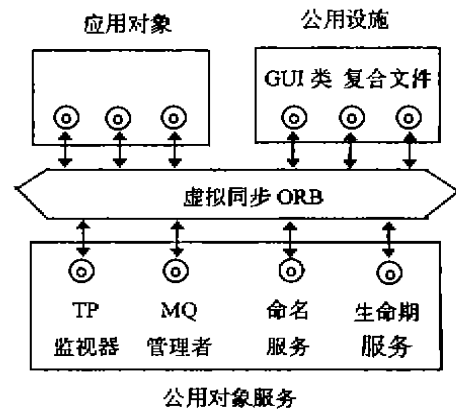


图 1 可靠系统 CORBA 结构

## 3 CORBA 在高性能实时系统中的应用策略

### 3.1 高性能实时系统需求

由于传统的 CORBA 实现系统开销大, 因而不能有效地支持实时 CORBA 的应用, 主要有以下一些原因: 过多的表示层转换和数据复制; 低效的服务器多路分解技术; 不可预测的用于网络读和写的缓冲算法; ORB 内部虚拟函数调用的长链; 不合适的操作系统系统调用的选择。因此研制实时 CORBA 的最终系统需求应该为:

- 说明点到点应用服务质量需求的策略和机制, 提供点到点服务质量保证的 ORB 最终系统应允许应用在高层次上用一些参数说明它们的服务质量需求。
- 优化的实时操作系统和网络。ORB 最终系统应能调度像 CPU、存储器等资源, 能控制存储吞吐量、网络适配器吞吐量和网络连接带宽及延时。
- 优化的实时通信协议。ORB 最终系统应能提供适合特定应用需求和网络环境的优化的通信协议。
- 优化的实时多路分解和分派。ORB 最终系统

应能提供像过滤器、延迟栈、直接多路分解和实时回调等机制,高效地、可预期地执行 CORBA 请求多路分解和分派。

- 优化的存储管理。ORB 最终系统的各个层次必须一起协作最小化数据复制。

- 优化的表示层。必须采用一系列技术减少各表示层转换的开销。

### 3.2 高性能实时 CORBA ORB 最终系统结构

Douglas C. Schmidt 等人提出了一个高性能实时 CORBA ORB 最终系统结构<sup>[5]</sup>,如图 2 所示。该结构包括:一个能在高速 ATM 网上执行千兆位传输率的优化了的操作系统 I/O 子系统;一组实时 GIOP/IIOP 协议,它提供了采用标准 CORBA 互操作协议进行高效的、可预期的请求传输;一个能实时地调度和分派 CORBA 请求的实时对象适配器;能优化 ORB 中关键资源开销和提供支持保证点到点服务质量应用及支持更层次化 CORBA 服务特征的特定应用的组件。

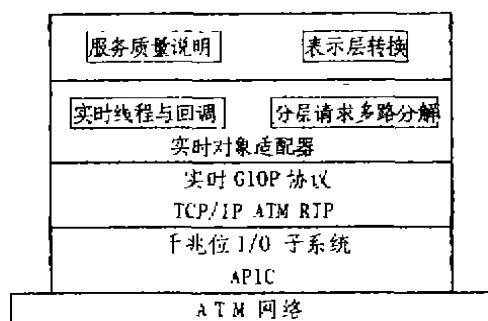


图 2 一个高性能实时 CORBA ORB 最终系统

千兆位的操作系统 I/O 子系统是建立在 APIC (ATM Port Interface Controller) 上,用一个零复制缓冲管理系统去增强传统的操作系统。即,在设备级上,APIC 直接和主系统总线及 I/O 设备交互,在最终系统缓冲池与 ATM 虚拟电路间传输 CORBA 请求无需额外的数据复制,I/O 设备的缓冲池支持周期和非周期的 CORBA 请求直接多路分解到用户和常驻核心线程共享的内存中。

性能敏感的实时应用经常不能容忍 TCP 的抖动和延时开销,也不能采用不可靠的 UDP。为了增强灵活性和性能,实时 GIOP 根据特定应用需要优化了高速网络 CORBA GIOP 并能适合于特定应用需求,如实时 ORB 有选择地省略了传输层的某些功能并直接在 ATM 或 ATM/IP 上运行,实时 GIOP

传输层通过像 ALF/ILP、千兆位 I/O 子系统与 APIC 网络适配器和底下的 ATM/IP 基础构件紧密结合。

实时对象适配器除了和 ORB 联合起来实现对象及把请求传递给合适的对象外,还负责实时回调和分层多路分解优化等操作的实时调度与分派。实时回调能保证对协议和 CORBA 应用提供单一速率的服务质量保证,当它和 APIC 的零复制缓冲及直接多路分解协作使用时,能为等时的多媒体应用显著减少同步化和前后切换的开销。为了使 CORBA 请求多路分解开销的最小化,实时 ORB 利用了分层的多路分解技术。该技术利用预定的多路分解键把 CORBA 请求直接映射到执行应用层实时回调的对象或方法中,为了进一步减少多路分解的层次,APIC 应与 ATM 虚拟电路联合起来进行编程直接分派 CORBA 请求。这一策略减少了多路分解时延并支持每一个请求或对象的点到点服务质量的保证。

特定应用代码和 CORBA 服务主要包括服务质量参数说明和表示层转换优化,实时 ORB 把连续媒体、大批量数据、低时延事务信息和高带宽信息流等区分开来,它们的服务质量说明用高级别参数定义,其它的服务质量用低级别参数定义,应用必须说明资源的服务质量以便 ORB 子系统能保证资源有效性。实时应用还必须说明每一个请求或对象的服务质量参数。对于整理 (Marshal)、还原 (Demarshal) CORBA IDL 类型,ORB 系统可有多种策略,实时 ORB 系统应根据时空情况优选一种方案。如 IDL 类型是以编译的还是以解释的存根和轮廓连接应取决于系统的时空要求。同样反复使用的已整理的数据单元应快速缓存,不必每次都要整个地加以整理。

## 4 CORBA 在互连网计算中的应用策略

Java 语言环境、WWW 和 CORBA 是互补的软件技术,当它们联合使用时可为在 Internet 上研制开发多用户分布应用提供一系列强有力的工具<sup>[6]</sup>。Eric Evans 等人提出了一个三者结合的分布应用模型<sup>[7]</sup>,如图 3 所示。该模型中,所有用户的 Client 软件用 Java Applet 来实现,这些 Applet 使用 CORBA 和它的远程应用程序的组件进行远程操作,用户需要这些 Applet 时就可透明地下载,因此省去了手工分布和安装任何特定应用的软件,这些 Applet 在 Browser 的 Java 上执行并以图形用户界面 (GUI) 的形式呈现在用户面前,用户通过 GUI 提出请求,其

中一些可以通过 ORB 产生对远程对象的 Applet 调用的方法,可以在服务器上执行远程对象服务程序而实例化这些远程对象,Applet 接受它的远程调用的结果并以 GUI 的形式显示出来。当然情形还可以

更复杂一些,比如,远程对象又可以作为其它宿主机的远程对象的 Client,或者 Applet 本身又包含一个远程对象,它的方法可以由在服务端上运行的软件来调用。

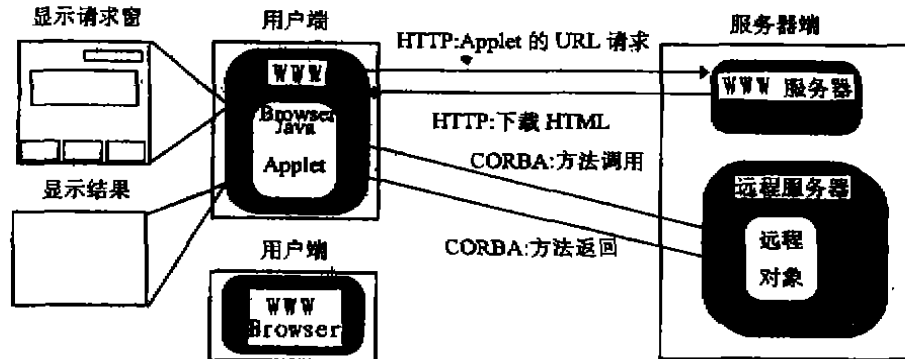


图3 Java CORBA 结合概况

**结束语** 分布对象计算(DOC)融合了分布计算系统和面向对象程序设计两个重要的软件技术,它使用面向对象技术高效、灵活和坚定地多机异构网络环境下分布可重用的服务和应用。CORBA 就是应分布对象计算技术的出现而提出的分布对象管理系统结构标准,它充分体现了分布对象计算系统的异构性、互操作性、可迁移性和可重用性。但是,由于 CORBA 标准和 CORBA 的常规实现没有直接论述像实时质量服务或高速性能、组通讯、部分失效和事件因果序等与分布计算有关的复杂的问题,用 CORBA 建立高有效性的应用是很难的,因此必须根据应用性质对公用对象服务、公用设施乃至 ORB 作些扩充。传统的 CORBA 实现系统开销大,因而更不能有效地支持实时 CORBA 的应用,一个实时 CORBA 的应用系统必须提供服务质量需求说明和保证机制并且对实时操作系统及网络、实时通讯协议、实时多路分解和分派、存储管理和表示层转换等进行优化。Java 语言环境、WWW 和 CORBA 是互补的软件技术,当它们联合使用时可为在 Internet 上研制开发多用户分布应用提供一系列强有力的工

具,Java CORBA 的结合是互连网上分布计算新的发展方向。

#### 参考文献

- 1 Vinoski S. CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments. IEEE Communications Magazine, Feb. 1997:46~55
- 2 Ben-Natan R. A Guide to The Common Object Request Broker Architecture. McGraw-Hill, 1995
- 3 汲化. 关于开放分布计算模型的研究:[南京大学计算机科学与技术系博士学位论文]. 1996
- 4 Maffei S, Schmidt D C. Constructing Reliable Distributed Communication System with CORBA. IEEE Communications Magazine, Feb. 1997:56~60
- 5 Schmidt D C, et al. A High-Performance End System Architecture for Real-Time CORBA. IEEE Communications Magazine, Feb. 1997:72~77
- 6 Leppinen M, et al. Java and CORBA-Based Network Management. IEEE Computer, June 1997:83~87
- 7 Evans E, Rogers D. Using Java Applets and CORBA for Multi-User Distributed Applications. IEEE Internet Computing, May-June 1997:43~55