

一个支持 QoS 的实时 CORBA 中间件 ORB 的结构模型

An Architecture Model for QoS-Enabled ORB Based on Real-Time CORBA

汪 洋 王振宇

(康达联环信息系统工程有 限公司 武汉 430070)(武汉数字工程研究所 武汉 430074)

Abstract It is very important for real-time distributed application to provide QoS guarantees end-to-end. CORBA acting as a middleware standard is widely used to support the distributed application development. But the conventional CORBA(2. x) doesn't support directly real-time application. Although CORBA3.0 commits some specifications for real-time system and QoS, it does not define how to implement them. This paper presents a real-time QoS-enabled ORB architecture model based on the layered QoS model. And it emphasizes the features and optimizations required to develop the Real-time ORB from the underlying network layer to application layer, and how to integrate all the services and implementations into a whole solution to achieve the QoS requirements.

Keywords CORBA, Real-time CORBA, ORB, Quality of Service (QoS)

一、引言

分布式网络环境中的实时应用如控制指挥系统、生产流程控制系统、远程通信系统、视频会议系统、分布式多媒体应用系统等,都要求提供端到端的高性能的服务质量(QoS)保证,为此,要求在网络、操作系统、中间件到用户应用等各个层次上都能协调地满足相应的实时 QoS 需求。QoS 主要包括系统资源如网络带宽、时延、抖动、系统可靠性以及内存管理等方面,因此,要保证应用的 QoS,就是要保证应用的网络带宽、小的时延,减少系统的抖动,提高系统的可靠性并进行高效的系统内存资源管理。

CORBA(Common Object Request Broker Architecture)是 OMG 定义的分布式对象计算(DOC)中间件的技术标准,为分布式服务和应用的开发带来极大的灵活性和可重用性,它在不同的应用语言、操作系统、网络协议和应用结构间方便地提供给用户在应用层的端到端(end-to-end)的互操作性。因而,将 CORBA 应用到高性能的实时分布式系统的开发中,是 OMG 和许多基于 CORBA 技术开发应用的开发者们一直关注的问题。作者在利用 CORBA 开发智能楼宇集成管理系统 IBMS^[6]过程中发现传统的 CORBA(2.2)在许多方面不能很好地支持实时应用的 QoS 要求,而 CORBA3.0 规范在支持分布式实时应用方面提出了相应的标准,但没有提供具体的模型和实现方式^[4]。本文结合作者的开发体会,重点描述了实时 CORBA 的结构特征,进而对基于实时 CORBA 提供端到端 QoS 保证的 ORB 的结构模型及其主要组成部份的实现和优化进行了相应的探讨。

二、实时应用的 QoS 模型

QoS 原来是描述网络通信领域中数据链路层的速率和可靠性等,现在则主要是指分布式实时应用中的一个或多个对象的集体行为的质量需求集,它是一组与服务质量有关的对象的集体行为,一般包括网络层、操作系统、中间件和应用的终端层,因而 QoS 是一个全局的概念,端到端系统必须能有效地管理参与传输的每一层,将用户应用层的 QoS 需求转换成中间件层(CORBA 的 ORB)的 QoS 参数,中间层将 QoS

需要映射到操作系统所能支持的 QoS,操作系统层将 QoS 映射到网络层,经过网络层的传输,然后又以逆过程将相应的 QoS 参数逐层映射成用户应用端的 QoS^[7]。所以,一个实时应用的 QoS 其实是依赖于网络系统中各部分的 QoS,并将各层处理的 QoS 通过 QoS 的协调机制完成全局的 QoS 处理,如图1所示。

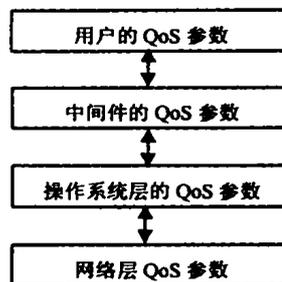


图1 QoS 层次模型

用户层的 QoS 参数指明了用户所希望的应用的服务质量如音频、视频质量的参数,它们一般分为与空间有关(如图像的宽、高等)和与时间有关(如延时、周期、传输速率)的相关参数说明。

中间件层的 QoS 参数描述了应用服务的请求,有些是可以直接应用用户层的 QoS 参数,有些则需要经过相应转换。

操作系统层的 QoS 描述了应用层定义的 QoS 对操作系统服务的请求。至此,已完成将用户的 QoS 需求转换成了系统级的 QoS 参数,以控制和管理系统的资源如系统调度、CPU 资源、磁盘空间和所需内存资源等。

网络层的 QoS 包括网络的带宽、传输的协议、报文包的大小、优先级、传输顺序、丢包率等。

在以上各层的 QoS 数据中,用户层的 QoS 是由用户在应用时来定义;操作系统级和网络层的则是根据用户层的 QoS 经相应映射和转换后由相应的操作系统和具体的网络技术来保证,而中间件层的 QoS 是完成从用户层的 QoS 到底层操作系统层和网络层的映射和转换管理,在整个 QoS 模型中占有十分重要的地位,而且基于不同的中间件技术,其处理的模型

汪 洋 博士研究生,主要研究方向为软件工程、中间件技术。王振宇 研究员,博士生导师,主要研究方向为软件工程、面向对象技术等。

是不同的。

三、实时 CORBA 概述

OMG 推出的实时 CORBA 规范 1.0^[1]是在 CORBA 2.2 规范和消息规范 (Messaging Specification)^[2]的基础上进行了相应的扩展,对固定优先级的 CORBA 应用支持端到端的可预测性,而动态调度规范^[3]是实时固定优先级调度的扩展,对应用程序调度服务增加了动态调度的特性。实际上,实时 CORBA 1.0 的主要特征来自 GIOP/IIOP 1.1 和消息规范中的服务质量 (QoS) 策略的基本框架,并将所有这些特征规范都归结到 CORBA 3.0 中^[4]。

对于实时应用而言,QoS 是保证实时性的重要方面。实时 CORBA 通过定义标准界面和 QoS 策略,保证应用系统指定端到端的 QoS,并定义相应的策略来管理 ORB 的处理、通信和内存资源。在实时 CORBA 中,通过以下方式扩展原有 CORBA 规范,支持端系统的实时性:在结构上扩展了 RTCORBA::PriorityMapping、RTORB、RTCORBA::Treadpool、RTPOA 和 RTCORBA::Current 等五个部分,实现优先级定义和映射策略、RTCORBA::Mutex 互斥界面、客户方显示连接策略、服务方线程策略、RIOP 协议、实时调度服务、实时事件服务等,避免优先级反转和非确定性的发生^[1]。同时,实时 CORBA 对传统 CORBA 的性能也进行了相应的优化,这些包括数据传输优化、内存使用优化、请求的分解与分发优化、尽量减少操作延迟等^[9]。

为了保证实时应用的 QoS,除了增加 CORBA 对实时应用的支持外,一个高效合理的体系结构是保证整个实时应用程序中 QoS 实现的关键。这个结构包括从底层的网络层到上层的应用层进行整体的协调与合作^[5]。因此,定义一个实时 CORBA 的 ORB 终端系统应包括网络接口、操作系统 I/O 子系统、通信协议以及 CORBA 中间件部件及相应的服务,如图 2 所示。

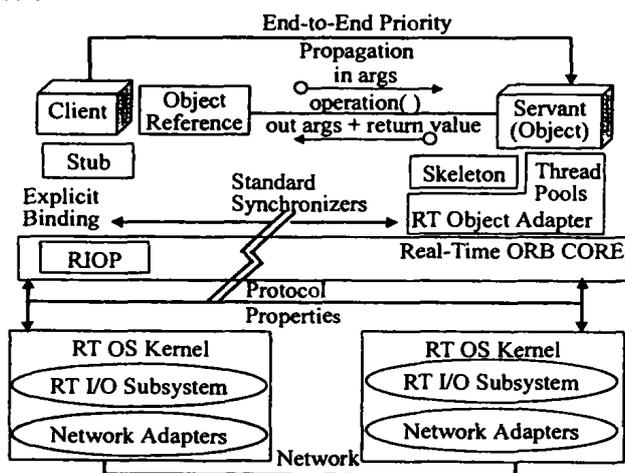


图2 实时 CORBA 的 ORB 端系统的结构模型

从上图可以看出,为保证 CORBA 客户端和服务端之间的端到端操作的 QoS,实时 CORBA 端系统的 ORB 不仅要提供垂直的(从网络接口到应用层)高效集成管理,同时还要提供水平(端到端)的集成管理,在各个层面上保证应用的 QoS。

3.1 网络通信层资源管理

在低层通信层提供相应的策略和机制保证足够的支持资源,这些支持包括用于特殊引用连接选择的管理以及高级 QoS 特征如控制 ATM 虚电路信元的传输速率等。

3.2 操作系统层的调度机制

ORB 利用操作系统的调度机制来调度端到端的应用层的活动。实时 CORBA 规范中实现固定优先级实时系统,这些机制负责管理操作系统中线程调度的优先级,实时 CORBA 的实现环境要求操作系统支持在应用层描述调度策略和优先级,操作系统能支持实时特征。

3.3 实时 ORB 端系统

ORB 负责处理 CORBA 的客户端和服务端端的通信请求。实时 ORB 端系统必须提供标准的接口供应用系统描述其向 ORB 所要求的资源请求。消息规范中已定义了相应的策略框架:允许应用来配置 ORB 端系统的资源以控制 ORB 的行为,这些资源包括处理器资源(如线程池、优先级机制和内部处理的互斥量),通信资源(通信协议及协议的绑定、传输层的联接、网络信息灯),内存资源(如消息队列的缓冲区大小、线程池的大小等)。实时 CORBA 应用中的 ORB 的实时性能是整个系统结构中的最关键因素。

3.4 实时服务

有了实时 ORB 管理端系统和通信资源还仅只解决了部分问题,实时 ORB 必须同时在更高层的服务和应用部件中为端到端之间高效、可扩充、可预计的操作提供相应的支持。如全局调度服务可用来管理和调度分布式资源,这样的调度与 ORB 联合可为端到端的操作时限提供相应的支持。

四、实时 ORB 终端系统关键部分的实现策略

以 CORBA 作为中间件的应用系统要实现端到端的 QoS,关键在于 ORB 端系统对 QoS 的支持。本节将结合图 2 的结构,围绕 QoS 的层次模型,重点讨论实时 ORB 端系统中的各个层次的关键部件及其实现策略,包括网络适配器、操作系统的 I/O 子系统、通信协议、内存资源管理以及公共对象服务等^[6]。通过这些关键部件的实现,从整体结构上有效地保证应用的 QoS。

4.1 网络适配器层

在网络层可以采用 ATM 网络或者是以太网技术来保证应用的 QoS,保证端系统缓冲池之间的 CORBA 请求能够在网络层快速传递,为 QoS 提供保证。

由于 ATM 网络技术是面向连接的,并对同一连接的数据包进行统一处理,所有数据包经相同的路由,可保证应用的带宽设定,每个信元短且分组固定(每个信元 53 字节),可控制时延和时延抖动,可提供不同的业务级别等特点,使得 ATM 网络在保证实时应用的 QoS 方面具有十分明显的优势。

另一方面,基于以太网(IP)的资源预留协议(RSVP),通过用户端给每个业务流(或连接)申请资源(包括缓冲区及网络带宽的大小)预留,在网络的每个节点上为这种应用预留相应的资源,因而可提供端到端的 QoS 网络层的保证。

4.2 实时通信协议

对支持 QoS 的传输协议而言,应该能支持网络吞吐量、延时以及可靠性要求,因此,ORB 端系统必须提供相应的通信协议以优化应用程序的要求和网络环境。但 CORBA 的 GIOP 协议是为了解决不同 ORB 之间的互操作,而 IIOP 协议则解决了如何将 GIOP 协议映射到 TCP/IP 协议。然而,传统的 TCP/IP 协议由于传输的时延和抖动,因此并不支持实时数据的传输,不可靠的协议如 UDP 缺少冲突控制、端到端流量控制和速度控制等,因而在网络中容易产生传输的冲突和数据的丢失。所以,IIOP 协议无法支持实时应用的传输要求。

由于 CORBA 规范并没有对 GIOP 协议的底层通信协议

做出明确规定,因而可用基于支持 QoS 的网络协议如 ATM 传输协议和基于 RSVP 的 IP 协议来开发相应的 ORB 产品,以更好地满足实时应用的端到端的 QoS 要求。

4.3 操作系统的实时调度

对系统资源的调度和使用最终都是通过操作系统完成的,即使 ORB 有能力描述 QoS 需求,但如果操作系统不支持可预计的 I/O 操作的话,ORB 仍然不能为应用程序提供端到端的 QoS 保证。因此,应用程序定义的 QoS 参数向操作系统提出请求后,操作系统将完成将面向应用的 QoS 参数需求转换成系统硬件资源的需求,进行资源需求的 QoS 映射,通过这种映射,ORB 端系统通过操作系统去调度相应的系统资源如 CPU、内存以及设置存储器的吞吐量、网络适配器的吞吐量、网络连接的带宽和延时等。

同时,为了处理系统中多个应用对保证 QoS 的请求,操作系统的调度机制必须允许高优先级的 CORBA 请求能够中断低优先级的操作而竞争使用相关系统资源,以不让其堵塞。

4.4 实时对象适配器

对象适配器负责完成对象实现与 ORB 之间的连接通信,将客户的请求分发给相应的目标对象。传统的 ORB 通过多个层次的多路分发请求要花费大量的系统资源,特别是当 ORB 管理大量的对象或者是在 IDL 接口中有大量的操作出现时,分发的效率就更低。为了减少系统开销,实时 ORB 端系统应尽量减少分发的层次,同时,采用多路分发技术,利用预分配的键值将 CORBA 请求直接映射成执行实时应用层的对象和方法。为了进一步减少多路分发的层次,可直接在底层的网络层如 ATM 虚电路上分配 CORBA 请求,以减少多路分发的延时并支持每个请求和每个对象的端到端的 QoS。

同时,实时对象适配器还负责对象的实时调度,确保影响客户端 QoS 需求的服务器端的相应策略在对象引用(object reference)中得到相应的保证,并根据优先级机制和模型来完成实时对象的调度。

4.5 优化内存管理

由于数据复制要消耗大量的 CPU、内存和 I/O 总线资源,因此,在 ORB 端系统的每个层次(网络层、操作系统 I/O 子系统协议层、对象适配器和表示层等)之间应相互协作,以达到最小的数据复制的目的。内存缓冲区可设计成在 ORB 的不同层次之间可以传递和预分配,这样,高层的处理可直接读取底层的缓冲区;在高低层之间尽量不传递数据本身,而是缓冲区中的地址。最理想的是零复制内存管理,为此,可将网络适配器的地址与内存地址统一进行编码,使 ORB 直接存取网络适配器的存储区,同时,为 I/O 设备提供的缓冲池对周期性和非周期性 CORBA 请求进行直接分解,直接在用户端和驻留核心的共享内存之间完成,以此来保证零复制,从而大大优化内存的管理,提高内存的使用效率,进而保证应用的 QoS。

4.6 线程池

在实时 CORBA 的服务器端,由于实时对象适配器要完成基于优先级的实时调度,因而可定义一个线程池来实现。每个对象适配器必须与一个线程池相关联,而每个线程池可以与多个 POA 相关。一个线程池中可包含多个线程服务,多线程能够区别不同类型的服务,如高优先级和低优先级服务。由于实时 CORBA 规范没有为多线程 CORBA 的服务器定义标准的 API 编程接口,因而,如果不使用特别的 ORB 特性,就不可能使用 CORBA 来编写多线程实时系统。为此,利用一个并发模型来实现服务器端的多线程机制,ORB 读取每一个来自低层通信机制的请求,处理完成后,进行转发并继续读取后续的请求。

为了表示并发机制,实时 CORBA 规范定义了一个标准的线程池模型,该模型允许服务开发人员预分配一个线程池并设置相应线程的属性如省缺的优先级等。线程池对实时的 ORB 终端系统非常有用,可限制内存资源(如栈空间)的数量等,而且,线程池可以用来优化缓冲区,并决定是否配置缓冲区,以更有效地控制使用内存资源。

4.7 优化表示层

表示层是将应用程序的数据转换成适当的数据格式,完成高层应用和低层数据格式之间的编码和解码的转换。这种转换由 IDL 编译产生的客户方 Stub 和服务方 Skeletons 来完成。可采用多种技术来减少表示层转换的开销,可在由表示层转换成的编译代码和解释代码之间进行相应的折衷处理,因为编译的代码效率高但要求更多的内存,而解释的代码则效率低但更紧凑。所以,作为实时 ORB 的端系统应该为解释型的和编译型的两类代码提供一个最佳平衡点。

4.8 全局调度服务

由实时操作系统定义的调度是在相对低层上实现的,这往往要求开发者实现将高层应用的 QoS 要求映射到低层的操作系统中的相关机制,如设置线程的优先级、虚电路的带宽和时延参数等,显然,这种要求对许多应用的开发者来说是极不直观的。为了在更高层次上以更直观的方式来允许应用程序定义相应的调度需要,实时 CORBA 规范定义了一个全局调度服务,该服务是一个 CORBA 对象,专门负责调度系统资源来满足应用的 QoS 的需要。应用程序可以利用实时调度服务通过各种参数(如最坏情况下和执行时间或周期)来描述他们的操作要求。

结束语 CORBA3.0 虽然提出了相应的规范来支持实时分布系统的应用,但如何利用这些规范来实现可用的应用系统,仍需要众多的 CORBA 应用者的共同努力。要保证基于 CORBA 的分布式环境中的端到端应用的 QoS,需要在整个系统的各个层次上协调配合,保证用户层的 QoS 请求能逐层映射到低层的相关参数,充分利用系统资源保证 QoS 的实现。作者在利用 IONA 公司的 CORBA 产品 Orbix 开发智能大楼综合管理系统 IBMS 时^[6],充分利用了多线程(线程池)机制、共享内存方法,并开发了相应的实时服务来提高系统的 QoS,保证分布于大楼内各个子系统的实时信息(包括视频信息)在所有的客户端的实时性。

参考文献

- 1 OMG. Real-time CORBA Joint Revised Submission. OMG Document orbos/99-02-12 ed., March 1999
- 2 OMG. CORBA Messaging Specification. OMG Document orbos/98-05-05 ed., May 1998
- 3 OMG. Dynamic Scheduling. OMG Document orbos/99-03-32 ed., March 1999
- 4 Siegel J. A Preview of CORBA3.0. Computer, May 1999. 114~116
- 5 Schmidt D C, Levine D L, Schantz R. The Design and Performance of Real-time Object Request Broker. Computer Communications, 1998, 21: 294~324
- 6 汪洋,王振宇. 基于 CORBA 的智能楼宇管理系统 IBMS 的设计与实现. 计算机与数字工程, 2001, 2: 17~26
- 7 Kalogeraki V, Morser L E, Melliar P M. A CORBA Framework for Managing Real-Time Distributed Multimedia Applications. In: Proc. of the 33rd Hawaii Intl. Conf. on System Sciences, 2000
- 8 Hong J W-K, Kim J-S, Park J-K. A CORBA-Based Quality of Service Management Framework for Distributed Multimedia Services and Application. IEEE Network, March/April, 1999. 70~79
- 9 骆志刚,刘锦德. 实时 CORBA. 计算机科学, 1999, 26(11)