

一种基于高性能中间件的软件体系结构^{*})

谢嘉孟 彭宏 林嘉宜 曾安

(华南理工大学计算机科学与工程学院 广州510641)

摘要 软件体系结构是软件设计的基础,在不同的应用领域有具体的设计和表达。随着网络建设的完善,我国各级政府职能部门建立在网络上的应用系统日益增多,本文以交通管理部门应用为例,系统分析了政府部门应用系统的特点和要求,并利用软件体系结构的方法,提出了一种专为该类应用领域设计的基于高性能CORBA中间件技术的非复用连接和反应者/线程优先级体系结构,为政府特定领域的行业级应用软件设计提供了一种可行的解决方案。

关键词 软件体系结构,中间件,模型

A Software Architecture Based on High-Performance Middleware

XIE Jia-Meng PENG Hong LIN Jia-Yi ZENG An

(College of Computer Science and Engineering, South China University of Technology, GuangZhou 510641)

Abstract Software architecture is the base of software design, and it has various implementations in different application domains. With the development of network, many government's application systems are set up on network. In this article, a traffic administration system is used as an example. With the analysis of the characteristics and demand of such domain, using software architecture methods, a kind of architecture based on demultiplex connection and reactor/thread priority architecture of high-performance CORBA middleware is brought forward. It gives a feasible solution to software design of specific government domains.

Keywords Software architecture, Middleware, Model

1 引言

21世纪是知识经济的时代,计算机的广泛应用和互联网的迅速普及极大地改善了人类的生存方式和国家间的竞争模式,各国政府纷纷大力倡导和推行电子商务、电子政务。在我国,自上世纪80年代以来,政府部门信息化建设得到高度重视,其中许多政府职能部门纷纷建立起基于本行业网络的计算机应用系统。如何充分运用新观念和高科技手段,配合政府实现职能转变,更好地实践为人民服务的宗旨,成为政府信息部门工作的重中之重。

近年来,政府部门在信息化建设过程中,面临着大部分早期开发的应用软件需要全面改造的局面,而且新应用领域的拓展又给政府提出了挑战。究其原因,主要是由于早期系统的开发受限于当时的软硬件理论和技术状况。从软件工程的角度分析可见,当今的软件设计已经摆脱传统“手工作坊式”、单纯靠经验的个体开发方式,开始走向系统和规范化的工业生产过程^[1]。软件体系结构已从基于单机运行发展到支持基于网络的分布式计算和应用系统。期间,各种实用技术:如面向对象技术、组件技术、分布式计算技术等为软件开发提供了有力支持。本文通过分析政府部门应用系统的特点,从体系结构理论出发,以交通管理部门应用为例,提出了一种专为该类应用领域设计的基于高性能CORBA中间件技术的非复用连接和反应者/线程优先级体系结构,为政府特定领域的行业级应用软件设计和改造提供了一种可行的解决方案。

2 政府部门应用系统的特点和要求

我国目前的行政管理体制属于条块结合型,虽然不同的政府职能部门之间有一定程度的差别,但总的来说,对于同一

部门,均存在从中央到地方的垂直领导关系,其应用系统架构就是建立在这种“中央—省—市—县—…”形式的广域网基础之上。作为国家公共安全部门有机组成部分之一的交通管理部门,其职能是负责对所辖行政区域范围内的车辆、驾驶员、交通事故和交通违章等业务进行管理,交通管理系统是政府部门的一个典型应用系统。该系统是基于广域网的分布式实时数据处理系统,处理内容主要是特定行政区域范围内各地的车辆信息、驾驶员信息、交通事故和交通违章等业务信息,实现实时办理各项业务的功能。系统的特点与要求在政府行业级应用系统中具有代表性,简要归纳如下。

第一,整个系统逻辑上划分为相应的层次型“数据中心”,各业务子系统通过各级“数据中心”的数据交换,连接成业务流水线。如:本系统的基本业务数据库分布在各地级市,各省和中央建立中心数据库,加强统一监管能力,本省内不同地级市的数据互操作通过省级数据库服务进行协调处理,不同省的地级市数据互操作通过全国数据库服务分发至有关省、市数据库进行处理。系统要求数据中心功能强大,各级数据处理和通信具有实时性、可靠性。

第二,系统覆盖地理范围广泛,建立在广域网基础上。但由于各地经济条件不同,网络节点的物理连接方式存在差异:光纤、DDN、电话线等连接介质并存,随着无线联网技术的成熟,GPRS、CDMA等联网方式也逐渐被采纳。因此,要求应用系统具有灵活支持各种联网方式的能力,特别是在无线网络环境中运行的应用必须解决时延、带宽和安全等多方面的问题。

第三,系统涵盖业务种类繁多,而且在软件生命周期内,业务内容将随着国家政策的调整而不断发生变化。因此,要求系统对业务变化具有很强的适应性和可维护性,并且能够建

^{*} 基金项目:国家自然科学基金重点项目(项目号:30230350);广东省科技攻关项目(项目号:A1020103)资助。谢嘉孟 博士研究生,研究领域:软件体系结构,智能技术。彭宏 教授,博士生导师,研究领域:体系结构、智能技术等。林嘉宜 博士研究生,研究领域:体系结构,数据挖掘。曾安 博士研究生。

立一种合理的系统维护模式,减少开发维护成本。

第四,系统的计算模型采用客户/服务器模型。系统在改造过程中经历了客户/服务器模型从二层到三层,系统设计技术从过程式到面向对象技术的演化,因此,软件设计要求使用基于三层结构的面向对象中间件技术,提出一种适合本行业应用的体系结构。

基于以上考虑,系统建设的核心是在设计上采用面向对象的技术,以提高系统的可扩展性和可维护性;在网络计算模型上采用三层应用系统结构,引入高性能的中间件,以提高应用性能,减少资源占用。其中,中间件的选择和相应体系结构的设计显得尤为关键。本文主要介绍这种基于高性能实时中间件的系统设计方案,重点阐述中间件的连接和并发结构。

3 一种基于高性能中间件技术的软件体系结构

目前, CORBA、Sun Java/RMI (J2EE)、Microsoft DCOM/COM+是大型应用开发中最流行的三种分布式对象模型。每种技术都有不同层次的平台和编程语言的兼容性,不同的技术适合不同类型的应用;DCOM 最适合工作组应用;JAVA/RMI 最适合互联网应用;而 CORBA 在对创建支持多种系统、多种联网方式、多种业务类型的企业级应用方面显得更加合适,对大型企业级软件开发的技术风险较低。因此,本系统选用 CORBA 中间件,并在连接和线程体系结构方面进行改进,构成了客户方和服务器方完整的客户/服务器操作的连接和线程体系结构设计方案。

3.1 实时 CORBA 是支持专门领域高性能应用的技术保证

实时 CORBA 是专为要求硬实时性和软实时性应用而设计的,所谓软实时性应用是指对系统响应时间有严格要求的高性能应用。实时 CORBA 为这类应用提供了标准的接口和策略来配置处理器资源、通讯资源和内存资源等系统资源。而且,除了 GIOP(General Inter-ORB Protocol)和 IIOP(Internet Inter-ORB Protocol)外,CORBA 规范还允许 ORB(Object Request Broker)实现定义环境相关的 ORB 互联协议 ESIOP(Environment Specific Inter-ORB Protocol),这些协议可以在特定应用领域提供 QoS、安全等方面的特性和保证,来满足性能敏感的应用对带宽、延迟和拥挤的严格要求。

3.2 CORBA 逻辑服务器体系结构

如图1所示。

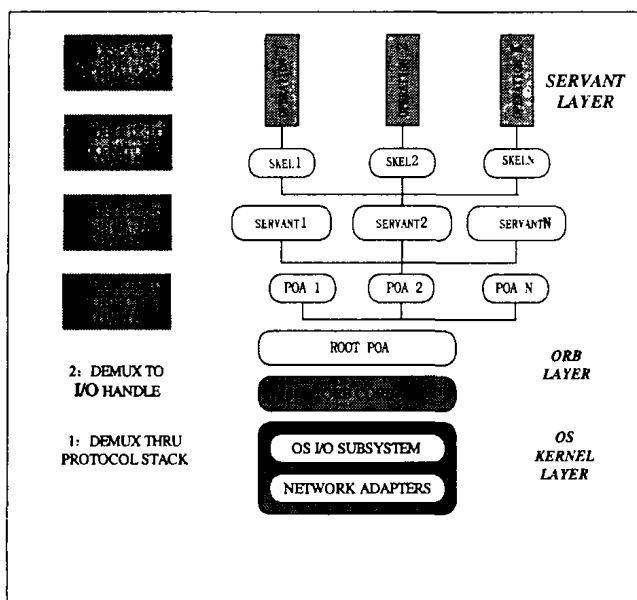


图1 CORBA 逻辑服务器体系结构

第一、二步:操作系统协议栈从低层到高层逐层分发客户请求,依次通过网络接口层、数据链路层、网络层和传输层直到用户/核心态的边界(即套接字层)。

第三、四步:ORB 核心利用客户对象钥匙的地址信息定位合适的 POA(Portable Object Adapter)和伺服器。当定位具有分级结构的 POA 需要顺着 POA 的结构逐级分发。

第五、六步:POA 使用操作名称找到合适的 IDL(Interface Define Language)框架,IDL 框架把请求缓冲的内容分解成操作参数并执行所要求的对象操作。

严格按功能进行分层的 ORB 分发机制并不适合高性能应用场合,理由如下:

(1)处理效率不高

客户请求逐层分发的开销比较大,特别是当 IDL 接口具有比较多的操作和一个对象适配器管理多个伺服器时,性能受到影响更大。

(2)导致优先级倒转和系统响应时间的不确定性

这种分层结构,客户请求严格按 FIFO 顺序处理,高优先级的请求可能需要等待低优先级的请求处理完之后才能处理,低优先级请求的处理时间可能并不确定,导致请求的优先级别高,但处理的优先级别并不高,该请求的系统响应时也不可确定,不能满足高性能应用严格追求系统响应时间的要求。

ORB 核心是 CORBA 中实现 GIOP 协议的部件,它负责连接的建立和实现并发体系结构。连接和并发处理体系结构是影响 ORB 核心性能的决定因素,为了提高 ORB 核心的性能,满足高性能应用的要求,必须选择合适的软件体系结构来实现连接管理和并发处理。

3.3 多路复用连接的体系结构

多路复用连接是最常用的连接体系结构,它通过尽量减少所需要的 TCP 连接数,来实现可伸缩性的应用,这种结构面临的最大挑战是解决同步 read(读),write(写)操作的问题。

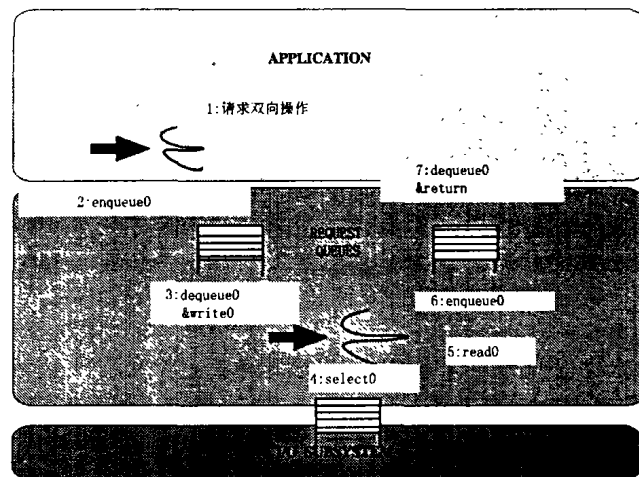


图2 活动的连接体系结构

3.3.1 活动的连接体系结构 如图2所示,这种结构主要基于生产者/消费者多线程模型。操作过程如下:(1)应用线程将操作放到队列中,(2)write 线程从队列中取走操作,并在套接字上发出写操作,(3)read 进程从套接字上读取返回,并将返回放在队列中,(4)应用从队列中取走返回。优点:实现容易,这是经典的多线程模型。缺点:分发开销大,实现效率低。

3.3.2 领先者/随行者(Leader/Follower)连接的体系结构 操作过程如下:(1)应用线程直接执行 write 操作;(2)执行 write 操作并等待回应的应用线程执行 select,并在套接

字上等待回应,该应用线程叫领先者;(3)在领先者之后等待读取回应的线程将阻塞在 ORB 核心管理的信号灯上;(4)当回应到达客户方时,领先者读取该回应,用 GIOP 头中的序列号确定回应的所属,如果回应是领先者自己的请求结果,领先者返回,并释放随行者的信号灯,随行者成为领先者;如果回应是其它线程的,领先者通知该线程读走回应,自己继续在套接字上阻塞。优点:减少了分发次数,提高了效率。缺点:复杂的实现逻辑和锁机制,带来极大的锁开销。

3.4 非复用连接的体系结构

这是专门为本应用系统设计的体系结构,如图3所示。

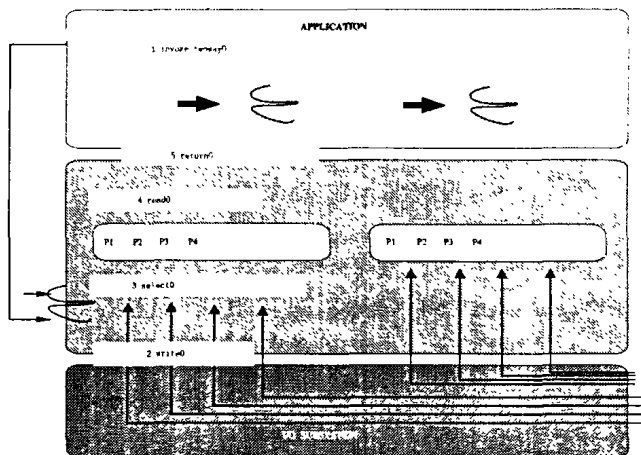


图3 非复用连接的体系结构

客户线程的 TSS(thread specific storage)保留预先建好到服务器的连接表,对每个优先级,一个线程分别拥有一条连接,可实现线程不共享套接字,所有 write,read,select,return 操作没有竞争问题。优点:可以保存端到端的优先级,没有资源竞争,同步开销小。缺点:增加了连接数,适合静态配置的应用,如优先级别少并且数目固定的应用。

针对交通管理这类系统,其应用种类比较固定,且优先级别少(包括:受理、复核、审批、制证等几类),因此,非常适合使用这种体系结构。

3.5 线程池体系结构

与非复用连接的体系结构配合,在此提出一种反应者(Reactor)/线程优先级体系结构,如图4所示。

这种线程体系结构把一组反应者和一组具有不同运行优

先级的线程相关联,在每个线程内,反应者执行以下操作:(1)分发来自客户方的请求,(2)read 请求并将它分发给相应的伺服器,由伺服器完成操作。优点:减少了上下文切换和锁处理的开销,它只需要在不同优先级线程之间,对伺服器的状态进行锁定,它还支持优先级和请求速率相关联的调度和分析技术。缺点:在一个线程内,对客户请求串行化,降低了并行性。



图4 反应者/线程优先级体系结构

在该系统中,我们利用非复用连接和反应者/线程优先级体系结构构成了针对本类应用系统客户方和服务方完整的客户/服务器操作的连接和线程体系结构设计方案。

结论 本文以交通管理部门应用为例,系统分析了政府部门应用系统的特点和要求,利用软件体系结构的理论和方法,提出了一种专为该类应用设计的基于高性能 CORBA 中间件技术的非复用连接和反应者/线程优先级体系结构,为政府特定领域的行业级应用软件设计提供了一种可行的解决方案。

参考文献

- 1 Shaw M, Garlan D. Software architecture: perspectives on an emerging discipline[M]. Prentice Hall Inc, 1996
- 2 Tanenbaum A S, van Steen M. Distributed Systems-Principles and Paradigms[M]. Prentice Hall Inc, 2002

(上接第92页)

阻塞和 OCC 算法中不必要的重启。本文中介绍了类 SCC 算法以及 SCC-1S, SCC-2S, SCC-nS. SCC-1S 并不需要额外的冗余而与 OCC-BC 算法一样. SCC-2S 需要一个额外的悲观(可推测)镜像,因而在满足时间限制条件方面比 OCC-BC 算法执行得更好. SCC-nS 需要 n 个额外的可推测镜像. 类 SCC 在解决冲突时并没有充分利用事务的截止时间和优先级. 虽然避免了优先级倒置问题,但同时在选择减少错过截止时间的任务数量的方法方面也失去了很多灵活性。

参考文献

- 1 Thomas R H. A majorith consensus approach to concurrency control for multiple copy databases. ACM Transaction on Database System, 1979, 4(2): 180~209
- 2 Haritsa J R, Carey M J, Linvy M. On being optimistic about real-time constraints. In: Proc. of the 1990 ACM PODS Symposium, April 1990
- 3 Haritsa J R, Carey M J, Linvy M. On being optimistic about real-

- time constraints. In: Proc. of the 1990 ACM PODS Symposium, April 1990
- 4 Braoudakis S, et al. Concurrency Control Protocols For Real-time databases. Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 1995
- 5 Bestavros A. Computer Science Department, CLA, Boston University, Boston, MA 02215 USA
- 6 Bestavros A. Speculative Concurrency Control: A position statement; [Technical Report TR-92-016]. Computer Science Department, Boston University, Boston, MA, July 1992
- 7 Bestavros A, Braoudakis S. A family of speculative concurrency control algorithm: [Technical Report TR-92-017]. Computer Science Department, Boston University, Boston, MA, July 1992
- 8 Bestavros A. Speculative concurrency control algorithm for Real-Time Databases. ; [Technical Report TR-93-002]. Computer Science Department, Boston University, Boston, MA, July 1993
- 9 Abbott R, Garcia-Molia H. Scheduling real-time transactions: A performance evaluation. In: Proc. of the 14th Intl. Conf. on Very Large Data Bases, Los Angeles, Ca, 1988
- 10 Menasce D, Nakanishi T. Optimistic versus pessimistic concurrency control mechanisms in database management systems. Information Systems, 1992, 7(1)