

• 博士论文 •

文章编号: 1000—3428(2004)11—0016—03

文献标识码: A

中图分类号: TP393.11

面向消息的实时中间件的研究与实现

余腊生, 王鲁鹏

(中南大学信息科学与工程学院, 长沙 410083)

摘 要: 研究了实时信息的特征、实时服务网络模型和面向消息的中间件设计方法, 并在此基础上对实时出版订购模型、消息传递的中间件分别进行了设计, 构造了实时中间件整体的模型, 最后给出了面向消息的实时中间件的部分实现。

关键词: 实时中间件; 消息; 出版; 订购; 实时服务; 通信对象; 控制块

Research and Implementation of Real Time Middleware Oriented Message

YU Lasheng, WANG Lupeng

(College of Information Engineering, Central South University, Changsha 410083)

[Abstract] This paper researches and designs some main technologies of real-time message-oriented middleware. This research mainly focuses on the characteristics of the real-time information, the real-time service network model and the design of message-oriented middleware. Based on this research, the real-time publish-subscribe model and the message passing middleware are designed separately. And then the design of the whole real-time message-oriented middleware is made. In the end, a simplified real-time message-oriented middleware is implemented.

[Key words] Real-time middleware; Message; Publish; Subscribe; Real-time service; Communication object; Control block

传统的分布式实时系统必须经过精心定制以便能在专用平台上操作。虽然这些定制的系统为实时应用的操作提供了所需的条件, 但是它们创建了一个不灵活的结构, 存在不容易修改、升级和与第三方产品集成的缺点。

在现在大规模的分布式系统中, 经常集成了实时和非实时的应用问题, 而一般这两类问题需求不同的运行平台, 这势必导致网络管理上的困难。因此要研制一个基于消息的实时中间件, 以实现实时应用与常规应用的无缝集成。

所谓中间件是在计算机硬件和操作系统之上, 支持应用软件开发和运行的系统软件, 具有标准的程序接口和协议, 从而避免了应用系统与具体平台之间的紧耦合, 能够使应用软件相对独立于计算机硬件和操作系统平台。基于消息的中间件(MOM)主要通过消息传递来完成分布式计算环境下数据和控制的传递。由于MOM既支持同步又支持异步通信方式, 因此特别适合于事件驱动的应用场合。

1 实时信息特征的分析

美国的Real-Time Innovations公司在设计NDDS系统时从3个方面对实时信息进行了研究^[1], 即数据流类型、数据流属性、数据流模式。通过对实时信息特征的分析, 我们想知道在后面的设计中应注意哪些问题, 尤其是如何设计实时服务网络模型, 以支持实时信息的传递。

实时信息一般有以下类型^[1]: 信号、命令、状态、事件和请求。不同类型的信息, 甚至同种类型的信息在不同的情况下, 经常对时间紧迫性和可靠性的要求不同。

根据对实时信息特征的归纳, 可以对实时中间件的设计提出以下要求:

(1) 由于存在多种类型的实时消息, 因此需合理地设计消息格式, 以便中间件或应用程序能容易、高效地解释数据。比如, 为了能用一致的编程方法解释数据, 需将系统中所有需要传递的信息, 统一分类、统一设计消息格式。另外, 消息格式的设计可用模块化设计。

(2) 不同类型的实时消息对可靠性、时间的紧迫性要求不同, 这

需要在设计实时服务网络模型时考虑对控制可靠性、时间紧迫性的支持及设计合理的控制策略。

2 实时服务网络模型

本文讨论的实时服务网络模型基于出版订购模型而建立。在出版订购模型^[1,2]中, 出版者只是匿名地发送某种主题的消息, 它不必知道订购者的数目和位置。订购者接收某种主题的消息, 它也不必知道出版者的数目和位置。出版者与订购者之间的耦合很小, 它们之间的通信及差异通过中间件解决。

要实现数据传递的实时控制, 需要对数据的出版和订购加入控制参数, 控制参数的定义如下:

(1) 出版参数

对于出版, 首先必须有主题和类型两个参数, 这是实现出版订购所需要的参数。出版的主题用来标志数据流, 类型则定义数据格式, 用来解释数据。

加入优先级参数, 用来设定同一主题不同出版者消息之间的优先级别, 它代表的是不同消息的重要性级别, 以满足数据量大, 出版者和订购者多的应用的需要。

对于出版数据有效期限的设定, 有两种方案。若能在系统中统一时间, 可以直接在出版数据中加入数据的最终有效时间参数。或者, 使用相对时间, 在出版数据中加入生存期参数。

订购者将根据数据的优先级参数和有效期限参数接收数据。对于同一主题未过期的不同数据, 订购者将优先接收优先级高的数据。这种机制会在最高优先级的出版者和它的所

基金项目: 211重点实验室建设项目; 湖南省科技园入园项目资金资助

作者简介: 余腊生(1969—), 男, 博士生、副教授, 主研方向: 数据库技术, 现场总线与实时通信技术, 结构与算法; 王鲁鹏, 硕士生

收稿日期: 2003-05-17

E-mail: riley@cs.hu.cn

有订购者之间建立比较健壮、有效的通信。

(2) 订购参数

主题必不可少。它用来识别数据流,订购者只接收指定主题的数据。

类型表示一种数据格式。某些情况下只需要接收某主题中特定类型的数据,就需要定义类型参数。可指定一个类型值(如全零值)表示接收所有类型数据。

在某些情况下,比如某主题的数据表示对环境或事物状态的测量值,它是不断测试和发送的,该主题的订购者可能并不需要接收每一条数据,这时就需要定义一个参数—最小接收间隔。若指定了最小接收间隔,订购者接收一条消息后,在其后的最小接收间隔时间内,将不接收数据,经过最小接收间隔时间后,订购者又重新接收数据。若指定最小接收间隔为0,表示订购者将不间断地接收数据。

在订购者开始接收数据后,却并无相应主题的数据到来,订购者不可能无限等待,应该在等待超出某个时间之后,通知订购程序,有订购程序决定下一步如何进行,因此需定义一个参数——等待期限(deadline)。

在实时系统中,不同的数据对传送的可靠性和时间紧迫性的要求是不同的。而可靠性与时间紧迫性的实现是相冲突的,因此应该用对两者的权衡值来表示对两种性能的要求。只需提供3种权衡值就可很好地满足各种要求:严格的可靠性、完全时间紧迫性和资源适应的可靠性。

3 面向消息的实时中间件分析设计

消息中间件一般有消息传递和消息队列两种实现方式^[3]。从消息传递和消息队列的工作方式来看,消息传递更适合实时应用,并且易于对消息传送、消息传送的可靠性和时间特性等方面进行控制。消息队列适合简单的信息交换,难以对信息的交换进行复杂的控制。所以对于设计实时中间件来说,选择消息传递方式更合适。

面向消息的中间件有一些通常需要包含的功能,对于一些主要的功能,可以采用如下的设计方法:

(1) 触发器功能的设计

在某个事件发生时,希望能够马上进行特定的处理,这就需要触发器功能,触发器机制在实现同步异步的机制中也会用到。实现触发器机制可以用以下方法:

- 1) 设置标志,并进行轮询。
- 2) 使用操作系统提供的事件触发机制。

(2) 数据格式翻译功能的设计

因为各种数据类型在不同平台上的格式可能不一致,所以当消息数据在不同平台之间传送时,需要对数据格式进行翻译。对数据格式翻译的设计由两个方案:

方案1:将每一种平台上的数据格式翻译成一种中间格式,接收到数据后,再将中间数据格式翻译成该平台上的格式。

方案2:发送数据时,不进行翻译,并在传送的数据中包含平台类型(平台类型信息的表示需要用简单数据类型,比如用消息数据的第一个字节表示,或者用方案1传送、解释平台类型信息);接收数据时,根据数据中的平台信息将数据翻译成该平台的数据格式。

要设计能适用于各种大型分布式系统的中间件,方案1更为合适,因为这种方式在设计新平

台的中间件版本时,不需要对其他的版本进行修改。

(3) 数据广播功能的设计

根据对实时服务网络模型的讨论,通过出版订购模型很容易实现数据广播。只要所有应用程序都订购某主题的消息,该主题的消息就可实现数据广播。

(4) 调试功能的设计

对消息中间件及其应用程序的调试主要是对消息传送的跟踪,可以通过输出中间件接收和发送的消息及中间件中消息队列中的消息提供调试功能。

4 面向消息的实时中间件在Windows下的实现设计

4.1 应用程序与中间件的通信设计

Windows提供了很多种单机进程间通信的方法,这些单机进程间通信方法的最低层机制都是内存映射文件。因此,如果要求达到较高的性能和较小的开销,内存映射文件是最好的选择。

4.2 共享内存区域的使用

能否合理利用共享内存存在应用程序和中间件间进行信息交互是影响系统效率的一个因素。根据应用程序与中间件间共享或传送的不同信息,可将共享内存划分为不同的区域。

通信对象(将出版者和订购者统称为通信对象)的部分属性可以放在共享内存中,以便中间件查询。因此,需要在共享内存中划分出通信对象控制块区(通信对象控制块指存放通信对象属性的内存区域)。

为提高效率,避免数据的不必要的复制,可以在共享内存中划分出专门存放消息数据的区域。通信对象可以将数据直接发送到该区域或直接到该区域接收数据。消息的处理也在该区域进行,不进行移动。

凡是需要用在应用程序和中间件通信或共享的数据都可以在共享内存存放。除了上面所说的两类数据,全局的控制信息和节点中的所有通信对象的主题信息都应该在共享内存中存放。

4.3 中间件大体框架

通过上面的设计,我们可以用图1简单的表示中间件的大体结构:

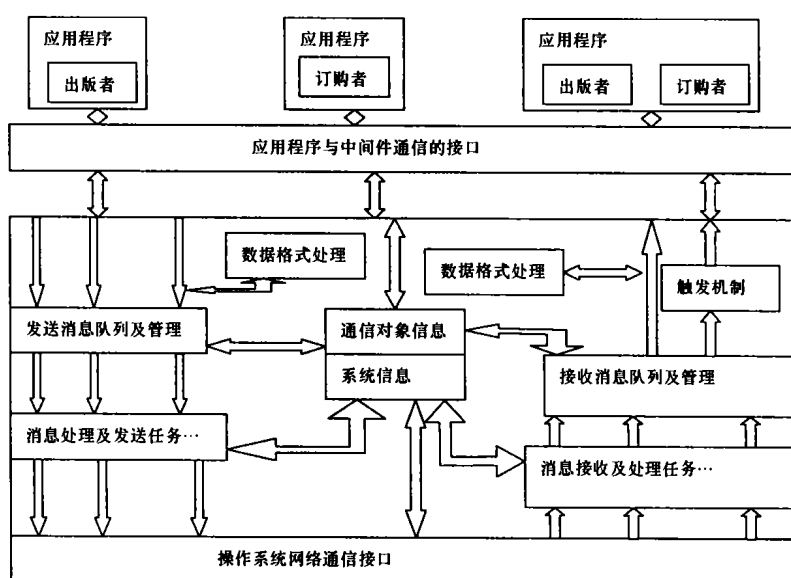


图1 消息中间件结构示意图

从图1可以看出,每个节点上运行的中间件需要保存有本节点中通信对象的信息和系统中其他节点的相关信息。为了保持系统信息始终是最新的,同时保持系统的灵活性,可以采用各节点对等广播方式或令牌管理方式。应用程序通过中间件通信的运行过程大致如下:

应用程序创建出版主题为A的出版对象,出版对象直接将出版的数据放入共享内存区,中间件将对共享内存区的消息数据进行格式处理、有关实时控制和其他控制的处理,中间件中的发送任务负责将处理完毕的消息发送出去。

系统中的其他节点接收到消息后,若判断消息有效(假设该消息主题为A),则根据本节点是否有订购主题为A的订购对象决定是否继续处理该消息。若本节点有订购主题为A的订购者,将消息处理以后,加入相应队列,供所有订购主题为A的订购者读取。

4.4 出版对象发送数据

出版对象发送数据是将数据插入对应出版主题项的消息队列中,出版对象发送数据的过程如图2所示。

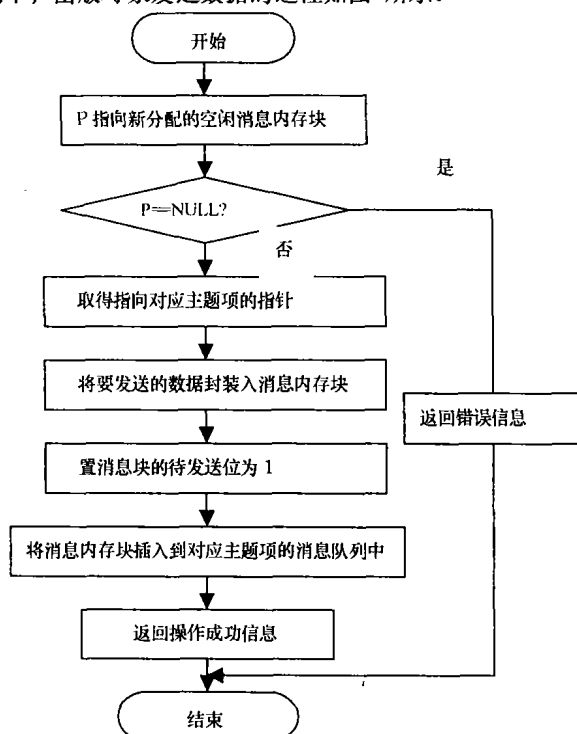


图2 出版对象发送数据的过程

(1) 分配空闲消息内存块,用来存放要发送的消息。若无空闲块,需要返回出错信息。

(2) 找到对应的出版主题项,为了减少临时查找的时间,出版对象中有一个指向对应主题项的指针,这个指针是在出版对象建立时赋值的。

(3) 封装数据。将数据从开发者使用的结构中提取出来,按设计的消息格式封装存放,并放入到分配的消息内存块中。

(4) 将消息内存块的首部的待发送位置1。该位为1表示消息未发送或未成功发送。如果该消息没有被放弃发送,则该消息内存块一直到待发送位为0,即消息被成功发送时,才能被释放。

(5) 将消息内存块插入对应主题项消息队列尾部,并修改主题项尾部消息指针。

(6) 返回操作成功的信息。

4.5 订购者接收数据

订购者接收数据的过程流程图(略),大致过程如下:

(1) 订购对象的控制块中有一个指针pNextMsg,该指针指向对应主题项的消息链表中该订购者下一个应取的消息。订购者根据该指针判断是否有消息可取,并用该指针取消息。

(2) 若指针不为空,订购者由指针找到消息所在的内存块。

(3) 订购者根据消息头计算数据大小,申请一块缓冲区,并读取消息内容,将消息中包含的数据按定义的格式解释到申请的数据缓冲区中。

(4) 将消息内存块的Count字段减1,如果Count为0,则释放消息内存块。

(5) 将pNextMsg后移。

(6) 返回数据缓冲区指针。

5 结论

本文就实时中间件的主要技术进行了研究,通过在实时信息的特征、实时服务网络模型、面向消息的中间件设计3方面对现有研究成果进行的分析、选择,发现消息传递比较适合实时应用中间件的设计,而出版订购模型是一种很灵活的结构,在它的基础上可以实现对实时信息的传送和控制。但基于出版订购模型的实时中间件一般适合于信息复杂的、大型的分布式系统,对于某些小型的、信息不复杂并且对实时性要求较高的系统,可能有较多的时间浪费在为了通用性而设计的实时控制处理上。

参考文献

- 1 Pardo-Castellote G, Schneider S, Hamilton M. NDDS: The Real-time Publish-subscribe Network. Real-time Innovations, Inc. White Paper <http://www.rti.com>, 1999
- 2 余腊生. 基于IP的实时出版订购协议. 计算技术与自动化, 2002-03
- 3 王 柏. 分布计算环境. 北京: 北京邮电大学出版社, 2000
- 4 Young R M. Real-time Protocol Strategies for Mission-critical, Distributed Systems. PhD Dissertation, University of Witwatersrand, 1996-07
- 5 [美] Richter J. 王建华译. Windows核心编程. 北京: 机械工业出版社, 2000
- 6 Real-time Innovations, Inc. RTPS Wire Protocol Specification Version 1.0, Draft Document Version: 1.17. 2002-02
- 7 Tanenbaum A S著. 熊桂喜译. 计算机网络. 北京: 清华大学出版社, 1998
- 8 International Middleware Association. Frequently Asked Questions. <http://www.imwa.org/utilities/faq.html>.
- 9 Pardo-Castellote G, Schneider A S. The Network Data Delivery Service: Real-time Time Connectivity for Distributed Control Applications. IEEE International Conference on Robotics and Automation, San Diego, CA, 1994
- 10 Open Horizon Inc. Building Mission-critical Financial Applications with Ambrosia. <http://www.openhorizon.com/html/FinancialWP.html>.