计算机科学2002Vol.29№.8

一种结合 ORB 与 MOM 的消息服务机制*)

A Message Services Mechanism Integrating ORB and MOM

周西京　朱静伟

(五邑大学　广东江门529020)

Abstract In this paper, the middleware based on Object Request Broker (ORB)and Message Oriented Middleware (MOM)are introduced. CORBA works especially well as a distributed object model because it hides infrastructure and presents distributed programs and data as objects. Message Oriented Middleware is an excellent message transport. It enables asynchronous exchange of data, event notification, persistence and quality of service (QoS). By combining the best of both technologies, an more reliable message services mechanism can be obtained.

Keywords ORB, CORBA, MOM, Message services

中间件作为前端客户机和后端服务器之间的一个中间层,为应用程序处理提供了一个位置,一般包含应用逻辑,负责接收客户端的应用请求,对请求做出相应的处理后将请求交给后端服务器,并负责将服务器的处理结果返回给客户端。其中基于 ORB 的中间件和面向消息的中间件(MOM)在消息服务上各有特色,各有所长。

1. CORBA 和 MOM 的现状

1.1 CORBA 消息服务

基于 ORB 的中间件主要是采用面向对象的技术。ORB 可以看作是语言独立性的面向对象的 RPC 应用。它的成员函数可以采用类似 Object-function()方式调用远端的对象。目前,ORB 存在两个彼此竞争的标准:CORBA ORB 和 DCOM ORB。本文主要讨论 CORBA 标准。

当使用 ORB 时,IDL 用于定义对象之间的接口,类似于 RPC 中的 IDL 定义过程的接口。ORB 特别适于对象接口变化不频繁,不会导致代码经常被重新编译及链接的情况。尽管 CORBA 也有动态调用接口来解决上述问题,但是实际上它们很难使用。

像 RPC 一样,ORB 主要采用同步、点对点通信方式。因此总的说来,CORBA 是一个同步请求—答复协议。用户调用一个方法,然后等候它完成。如果不想等,必须在自己的线程里运行每一个调用。CORBA 1.0版本定义了三个请求调用机制来避免这种限制:1)可以利用 CORBA 事件服务;2)可以使用动态调用接口(DLL)的延迟同步模式;3)可以声明一个方法是单向通信(或数据报)。因此,CORBA 需要可靠的通信层,当网络层不可靠时,它们很难定位错误。

多年来,不少 CORBA 的研究者认为,缺乏对异步消息的强有力支持是 CORBA 规范的一大漏洞,因此,与其它的开放式分布处理标准/规范(如 RM-ODP)比较,CORBA 在支持大规模分布式处理方面已经相形见绌。一些 CORBA 厂商则在自己的产品中增加了一些专用扩展来克服上述缺点基于 ORB 的中间件,主要专注于分布式及异构环境。但是,建立这种分布式应用比较复杂,很少有工具直接支持 ORB 的开发,同时,ORB 没有给开发者提供负载平衡及 Fail Over 的功能。

1.2 基于消息的中间件(MOM)

MOM 提供了一个完整的处理环境,允许开发者及用户连接不同系统之间的数据和代码,或采用一致的界面进行应用处理的互连。MOM 提供了一个高层应用接口,为不同系统提供操作核心。

MOM 主要是通过将信息以消息的方式在程序间传递来完成。MOM 结构如图1所示。

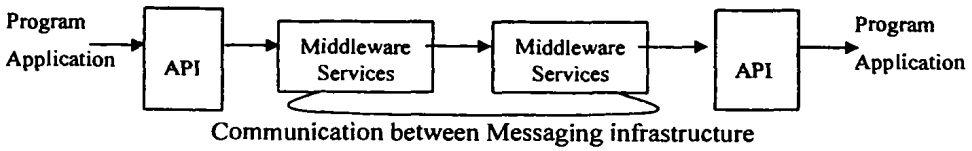

图1　面向消息中间件 MOM 结构图

MOM 一般可以分为两种形式:消息传递(Message Passing)和消息队列(Message Queuing)。消息传递在建立大型的分布式应用中比较常见。其主要的模式是广播/订购(Publish-Subscribe)方式。采用该方式,应用程序既可以订购,也可以广播。

Publish-Subscribe 通信模型提供了位置透明性。程序只需简单地将消息以主题方式发送出去,由中间件来将消息传递给所有订购该主题的程序。

MOM 主要通过 agents 技术来实现 Publish-Subscribe 方式应用。当程序广播消息时,首先与一个代理进行连接,之后将消息传递给代理。代理负责路由消息给相应的程序。由于代理可以实现消息的动态路由功能,因此,Publish-Subscribe 方式能够提供较好的容错功能。它比较适合于对容错性和可靠性要求较高的分布式应用,但它缺乏 MOM 的异步特性,不太适合长时间网络断开的情况。

消息队列方式在程序间的通信中一般要借助于队列来实

*)本文得到广东省自然科学基金资助。

现。它允许程序无需直接建立起连接即可发送和接收消息。使用消息队列方式，程序只需简单地将消息发送给消息队列，由消息队列负责消息的传递，对应用程序完全透明。消息队列可以认为是 pull 技术，程序从消息队列中提取消息。在大型企业级应用中，消息还可以从一个队列转发到另一个消息队列中。

消息队列采用异步方式，为信息提供了一个安全的存储方式，特别适用于不是直接连接的应用，如移动用户、发送方或接收方进程可能处于不活动状态的应用。

Message 队列方式的缺点是需要一些配置工作，性能不是很高，而且如果队列丢失，整个系统将受到影响。MOM 提供同步和异步两种方式访问功能，它的异步访问方式更加常用。它将请求保存在队列中，并在应用可用时将请求传递给它们。

MOM 可以克服基于 ORB 的中间件的限制，可以提供基于消息的异步通信机制，因此，MOM API 调用不会阻塞应用程序。同时，MOM 不会占用大量的网络带宽，可以跟踪事务，通过将事务存储在磁盘上，可以恢复系统及网络故障。

但目前 MOM 也存在一些问题：需要开发者学习新的 MOM API 编程；缺乏流行的开发工具的支持；如果使用传统的客户机/服务器开发工具集成 MOM，必须使用 MOM 厂商提供的 DLL、Active X 控件或同工具类库相集成。尽管 MOMA 组织正在进行 Meta Level 的互操作标准化工作，但有关 MOM 的标准还未形成。

1.3 MOM 和 CORBA 的消息服务的比较

比较请求-答复和 MOM 的消息接发排队范例，就象是比较电话呼叫和通过信件通信。使用电话的相互作用是直接的——双方直接交谈，电话交谈结束时一个工作单元就结束了。相反，通过邮件通信允许用户控制工作流，而不是电话。但另一方面，这可能在客户机端阻挠不接收直接反馈。

下表1比较了请求-答复和 MOM 的消息接发排队模型。当然消息接发排队比请求-答复更灵活，耦合更松散，更有耐性。然而，消息接发排队只是在时间上使事情置后，并有可能有它自己的复杂性。消息接发排队促进事件驱动通信模型。用户可以发出多个请求给多个服务器，然后当响应返回时进行接收。应用程序无阻碍的等候响应。相反可以将响应看成事件，当它们发生的时候，用户必须准备好处理。请求-答复系统可以利用线程模仿这种异步的、松散耦合的、事件驱动的行为。

表1 请求-答复和 MOM 的消息接发排队模型的比较

特性	MOM 消息接发和排队	请求-答复
形态	像邮局	像电话
C/S 时间关系	异步。客户和服务器可能在不同的时间，以不同的速度工作	同步。客户和服务器必须同时工作，服务器必须跟上客户
C/S 排序	无固定排序	调用—返回
参与对象是否可用	否	是
负载均衡	可以用简单的队列来实现先进先出或基于优先权的政策	需要分离的对象 TP-Monitor（或智能 POA）
是否支持事项处理	是(MOM 与之间)	是(端到端)
消息过滤	有	无
性能	慢。需要一个中间转发	快

因此，一个先进的 ORB 应该支持 MOM 和请求-答复两种形式的通信。每一个不同的形式都给出了它的范例，用户终将选择(或许是通过一个服务质量参数)一个最适合自己特定需要的形式。

2. CORBA 与 MOM 结合

2.1 对 MOM 的扩展

CORBA 给 MOM 提供了一个公用对象模型，这个模型包括接口定义语言和一个强大的体系结构框架，应用程序将建立在这个框架上。

第一，ORBA 提供基于对象的统一接口。当前，每一个 MOM 产品提供它自己的非标准应用编程接口(API)。这些 API 大多是过程性的。CORBA 给 MOM 提供标准的对象接口以及对接口定义语言的支持和一个完整的分布式对象基础结构。

第二，为描述消息内容提供技术。目前，由于消息内容对 MOM 是不透明的，用户必须重新发明技术来描述其消息内容，包括数据类型。CORBA 允许 MOM 消息通过接口定义语言进行类型化，还提供一般数据类型。

第三，定义消息交换的端点。目前，每一个 MOM 有自己的描述端点构成内容的模型，一个端点可以是一个队列，一个进程或一个对象。在 CORBA 中，一个端点是一个独一无二的对象标记。

第四，提供可互操作的中间件。每一个 MOM 必须定义自己的专有线路层协议，没有一个产品可以互操作。CORBA IIOP 定义了线路协议，允许多个不同的 MOM 能够互操作。IIOP 协议还支持事项处理和安全性，这样，在 IIOP 协议之上建立的 MOM 将可以互操作。

第五，提供了一个 Internet 基础。CORBA IIOP 协议正成为一个标准的 Internet 协议。这意味着 CORBA 将成为端口、URL 映射和防火墙标准代理。如果 MOM 没有 CORBA 的支持将不得不重新建立这些基础结构。

总之，CORBA 给 MOM 提供了标准接口和对象基础，CORBA 将允许用户编写可移植的和可互操作的 MOM 应用程序。

2.2 具有 MOM 功能的对象请求中介

通过扩展 CORBA 去支持 MOM 的语义学，除了请求-答复之外，对象还应该通过把消息放进队列里和从队列里取出消息，在 IIOP 对象请求中介之上进行通信。使用 MOM 或者利用请求-答复都可以调用相同方法。对象请求中介应该可以根据用户规定的服务质量(QoS)，在运行时决定调用形式。

因此，包含 MOM 功能的 ORB 主要消息服务内容包括：异步消息服务、与时间无关的调用和消息服务的服务质量。

第一，为获得异步消息服务可以采用两种异步请求模型：

·回调：在每个调用请求中，客户提供一个额外的对象引用。当 ORB 收到对象实现的执行结果时，将根据这个对象引用将结果返回给客户方的应用。

·轮询：当客户调用一个操作时，马上就可以获得一个返回数据，该数据就用于单独的轮询。

与原来的延迟同步请求模型不同的是，回调和轮询异步模型可用于使用静态存根的客户。

第二，MOM 允许 CORBA 为流动客户提供与时间无关的排队服务，允许客户请求目前在物理上与客户尚未连接的

对象所提供的服务。MOM 允许客户机和服务器是不可用的。通过在 GIOP/IIOP 中引入"中间路由代理"来完成这一功能，这些代理主要完成存储-转发功能。当目标对象当前不可访问时，代理保存客户发送的消息。当客户和目标对象之间的连接由于异常原因被关闭时，代理也可以保留对象实现的应答。至于请求和应答的生命周期，则由相应的 QoS(Quality of Service)来控制。即允许客户和服务器在不同的时间里运行。客户发出的请求可以不在客户对象的的生命期内完成。这个模型在松散耦合的企业间情况下非常有用，这两方没有必要等着事项处理完成。

第三，在 CORBA 平台上，ORB 向客户方的应用屏蔽了许多与分布式计算有关的细节，如对象的定位、网络连结的建立和请求的发送等。使在实现 CORBA 规范时有很大的自由度，并使 CORBA 平台的易用性体现得非常充分。但是，过多地屏蔽实现细节使应用难以控制底层 ORB 所提供的服务的质量。CORBA 消息服务通过允许客户说明它所需的服务质量来弥补这一漏洞，如客户可以提出对消息分发、消息排队和消息优先级的要求。CORBA 的 QoS 策略是，让 QoS 可以在多个层面上发生作用：在 ORB 层，它影响所有的请求；在线程层，它影响该线程发出的所有请求；在对象引用层，它影响所有发给这一对象的请求。

第四，允许服务器决定何时从队列里检索出消息。服务器对象既可以在先进先出的基础上从队列里检索出消息，也可以依据某种优先权或负载均衡模式从队列里检索出消息。服务器也可以使用消息过滤器仍掉它们不想处理的消息，或可以把这些消息传送给其它的服务器。队列既可以是持久(记录在磁盘上)的，也可以是非持久的(在存储器里)。队列的类型取决于用户规定的服务质量。

第五，允许客户提出无阻塞请求。MOM 允许对象提出不要阻塞客户执行线程的异步请求。这就意味着用户的客户机没有必要是多线程的。这样它们更易于编写和维持。

MOM 风格的通信使 CORBA 更灵活，更有耐力。

3. 怎样实现基于 CORBA 的 MOM

有下列几种方式实现 CORBA 和 MOM 结合：

·客户端和服务器端两端都可以通过一个服务质量属性规定消息接发形式。利用服务质量，客户和服务器都可以规定请求或响应的形式。QoS 可以包括以下属性：确认等级、存活时间、优先权、成本、可靠性、路由选择、事项处理支持和持久性等级。

·客户机可以在单个调用的基础上设置 QoS。客户机可以使用 current 准对象逐个调用地控制 QoS，管理器也可以设置一个缺省 QoS。在规定了 QoS 之后，客户机可以进行一个普通的 CORBA 调用。

·客户机规定一个回叫对象去处理响应。延迟响应被发送

(下转第144页)

(上接第149页)

用互斥对象编程的时候最复杂，但也是控制共享资源最为强大的方法，它不仅能够在一个进程内的线程之间实现资源的完全共享控制，而且可以在不同的进程的线程之间实现。

我们在本模块的实现中，可以用事件对象来实现各个线程的同步。在没有预定事件时，输出处理线程、输入处理线程和其他处理线程挂起以消耗尽量少的资源，监视线程检测到有预定事件时，用一事件对象通知主线程。请求相应的处理。主线程接收发送来的消息，自己处理或唤醒相应的线程处理程序，使信息得到实时处理。这种方法比用轮询法效率要高得多，编程也更为雅致。

用 VC++6.0实现的部分代码如下：

```
if ((m_hPostMsgEvent = CreateEvent (NULL, TRUE, TRUE,
    NULL))==NULL)
  return FALSE;
memset(&m_osRead, 0, sizeof(OVERLAPPED));
memset(&m_osWrite, 0, sizeof(OVERLAPPED));
if ((m_osRead.hEvent = CreateEvent (NULL, TRUE, FALSE,
    NULL))==NULL)//为重叠读创建事件对象
  return FALSE;
if ((m_osWrite.hEvent = CreateEvent (NULL, TRUE, FALSE,
    NULL))==NULL)// 为重叠写创建事件对象
  return FALSE;
  ClearCommError(pDoc->m_hCom,&dwErrorFlags,&ComStat);
  if(ComStat.cbInQue){
   WaitForSingleObject (pDoc-> m_hPostMsgEvent, INFINITE);
   // 等待 WM_COMMNOTIFY 结束
  ResetEvent(pDoc->m_hPostMsgEvent);
  PostMessage(pDoc->m_hWnd,WM_COMMNOTIFY, EV_RX-
CHAR,0);//通知视图
  continue;
}
  dwMask=0;
if(!WaitCommEvent(pDoc->m_hCom, &dwMask, &os)){
  if(GetLastError()==ERROR_IO_PENDING)
     GetOverlappedResult (pDoc-> m_hCom, &os, &dwTrans,
TRUE);// 无限等待重叠操作结果
    Else{
      CloseHandle(os.hEvent);
      return (UINT)-1;
    }
}
```

结束语 通过以上的讨论，我们可看到多线程技术能很好地解决各种逻辑并发性和物理并行性问题，改善系统的性能如吞吐量、计算速度、响应时间等，提高程序的执行效率和资源的利用率，同时也提高了程序的可读性和稳定性，在实际的开发中有着非常广泛的应用。多线程有着诸多的优点，但同时也带来了一些新的问题，这随着系统规模的增大而更加突出，主要表现在：

(1)多线程结构比原来要复杂，线程规模越大，程序设计的复杂性也越大，可维护性也越差；

(2)多个线程的调度和管理是要耗费系统资源的，线程越多、调度越频繁，耗费越大，甚至会降低系统实际可得到的性能。

从另一方面，线程规模太小又不足以体现多线程技术的优点，不能充分提高系统的资源利用率。如何解决多线程在设计、运行时的复杂性，以及在资源利用率、用户作业周转时间、系统吞吐量等诸多方面达到最佳，以确保在任务完成的基础上达到整体最优，是多线程技术应用研究中面临的一个新的课题。

参 考 文 献

1 张宏莉，田耕，胡铭曾. 多线程技术与并行运算，计算机科学，1999,26

2 李春华，徐明，周兴铭. 多线程的软件实现，计算机工程与科学，1999,21(4)

3 Denver A. Serial Communications in Win32 Microsoft Windows Developer Support

4 陈章龙，陈泽文 编著. IBM-PC 机软硬件接口及试验. 北京：人民邮电出版社

3. 计算机辅助网络工程规划设计系统

计算机辅助网络工程规划设计系统主要由以下几个功能模块组成：

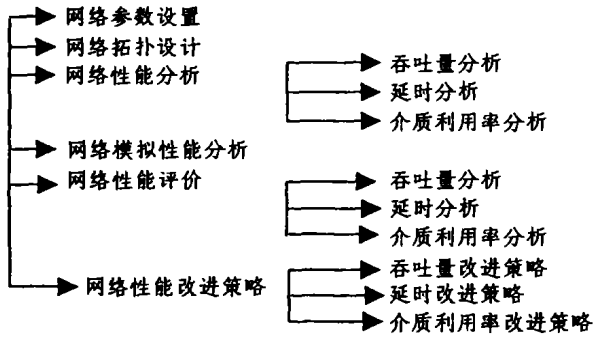

表1所示为令牌环的部分模拟结果。根据模拟得到的数据绘出的吞吐量-平均延迟曲线如图5：

表1

包的最大个数	平均延迟(ms)	吞吐量(packets/s)
100	0.128	436
200	0.139	439
300	0.140	442
400	0.141	458
500	0.141	465
1000	0.140	482

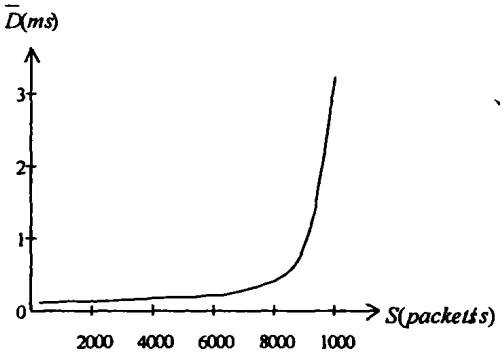

图5 令牌环平均延迟与吞吐量的关系曲线

CSMA/CD 和令牌环系统分析结果如图6：

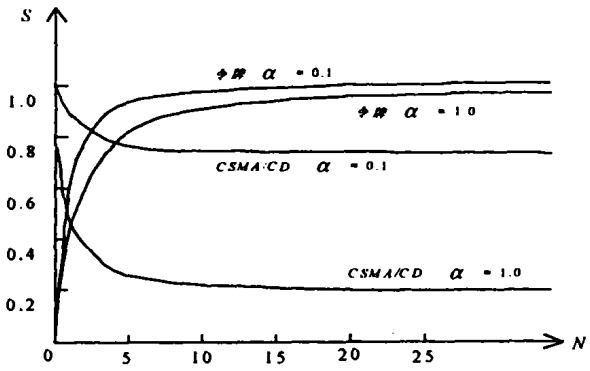

图6 CSMA/CD 和令牌环的 N-S 曲线

网络工程辅助设计系统的优点是：大大缩短网络工程建设周期；减少了人力和财力的浪费；提高了工程的可扩展性和适应性；未建网前即可预测网络的性能；为科学规划网络工程提供了理论依据。其不足是：对具体网络设备的考虑上欠缺些；所建立的网络模型并不能完全代替物理系统；系统对高层的协议性能考虑得少些；要想使计算机网络工程辅助设计系统几乎完全接近物理系统，还需要做大量的工作。随着软件技术的迅速发展，相信这种要求是能够逐渐实现的。

参 考 文 献

1 黄叔武，杨一平. 计算机网络工程教程. 北京：清华大学出版社，1999. 100～120
2 王宝济. 网络建设实用指南. 北京：人民邮电出版社，1999. 160～170
3 [美]Breger R，Riley S，肖文贵，肖丹等译. Swithed and Fast Ethernet Second Edition. 电子工业出版社，1999. 150～167
4 钟嘉强译. J. L. Hammond Peter J. P. O'Reilly Performance Analysis of Local Computer Networls. 人民邮电出版社，1986. 230～245
5 Tobagi F A，Hunt V B. Performance Analysis of Carrier Sense Multiple Access with Colission Detection. Computer Network，1980(4)：240～255
6 何诚. 计算机局部网络结构与性能分析. 北京：中国科学技术出版社，1989. 240～266
7 Tobagi F A，Hunt V B. Performance Analysis of CSMA/CD. Computer Networks，1980(4)：160～180

（上接第130页）

到一个回叫对象——通常这是一个管理客户机上或网络服务器上持久队列的对象。

·有队列识别的服务器必须提供一个 MOA。服务器必须提供一个由 POA 衍生而来的消息对象适配器(MOA)。MOA 允许有队列识别的服务器控制它处理请求的顺序。

·消息可以是对象。利用 CORBA3.0按值传送以及象查询和持久性之类的对象服务，可以存储，查询，检索和传送这些消息，在周围传递这些消息。

CORBA MOM 最有可能建立 IIOP 在协议之上，但必须保证 CORBA 和 MOM 的结合是动态的。这为分布对象增加了大量新的可能性。例如：MOM 能与 CORBA 服务——比如事件，查询和交易器一类的服务集成，为 Object Web 提供非常强大的 Publish-Subscribe 系统。

参 考 文 献

1 Orfali R，Haarkey D，Edwards J. Instant CORBA. 1997. 202～206
2 O MG. The Common Object Request Broker Architecture and Specification. 1998，7
3 OMG. CORBA Messaging. Joint Revised Submission. Document orbos，1998(5)：5
4 Otte R，Patrick P，Roy M. Understanding CORBA. New Jersey：Prentice Hall，1995
5 Vinoski S. CORBA. Integrating diverse application within distributed heterogeneous environments. IEEE Communication Magazine，1997，35(2)：45～55
6 Object Management Group. CORBA successful stories. http://www.corba.org/
7 刘锦德，苏森. CORBA 技术的新发展. 计算机应用，1999，19(5)：5
8 Craig T，Gil H. Current web architecture. Object services and Consulting，Inc. 1996