维普资讯 http://www.cqvip.com

CORBA 事件服务及其在分布式并行计算中的应用*)

CORBA Event Service and it's Application in Distributed Parallel Computing

李金宝[1,2] 李建中[1,2] 于 玲[2]

(哈尔滨工业大学计算机科学与工程学院 哈尔滨150001)[1]

(黑龙江大学计算机科学技术学院 哈尔滨150080)[2]

Abstract The Common Object Request Broker Architecture (CORBA) is a main specification applied to the distributed parallel program design at present. It includes many services, in which Event Service can provide the asynchronous communication model for the distributed parallel program design. This paper introduces the Event Service of CORBA, then designs and implements a distributed parallel model based on Event Service. This model is made to implement overlap of the computing and communication , to reduce the unnecessary time for waiting and enhance the parallel effect. The paper also shows the practical application of the model with an example of the distributed parallel matrix multiplication.

Keywords CORBA, Event service, Distributed parallel computing

1 引言

随着网络技术的飞速发展和日益完善,分布式并行程序设计已经成为软件开发的主流技术之一。在分布式并行计算环境中,异构分布式环境中资源的透明存取和系统中各个异构节点间的通信问题是一个主要问题。OMG 提出的 CORBA 规范[2],能很好地解决异构系统中面向对象的应用之间的通信问题,因此 CORBA 适合作为分布式并行应用程序的中间件[6]。

CORBA 的基本通信模型是同步通信[1],而在分布式并行计算中,同步通信模型不能完全满足高效计算的需求。在分布式并行计算环境中,一方面,客户端也是计算的主要参与者,我们要充分利用客户端的计算资源,避免无效的空闲;另一方面,客户与服务器是相对的,在不同的情况下服务器也可能成为客户,反之亦然。为了避免同步通信模型带来的空闲,有必要为分布式并行计算选择一种异步通信模型,以实现计算和通信的重叠,获得更好的并行性。在 CORBA 提供的许多服务中,事件服务[1~3]是一种比较合理的选择。

本文首先介绍了 CORBA 事件服务的基本原理,然后给出了一个基于 CORBA 事件服务的矩阵乘法的分布式并行计算的实例。

2 CORBA 事件服务的基本模型

在 CORBA 标准中,事件服务将对象分为两种:提供者和使用者。提供者产生事件,使用者处理事件。事件信道是一个标准的 CORBA 对象,驻留在对象请求中介(ORB)上,它允许多个事件提供者和多个使用者异步地通信。一般的事件信道对象并不了解它所传送的数据内容,它仅仅传递属于类型 any 的单个参数,提供者和使用者必须同意公用事件语义[4],此外事件信道对象也允许传递以 OMG IDL 定义的类型化事件信息[5]。

事件信道是事件服务的核心,它支持两种数据传送模型:push 模型和 pull 模型。

·push 模型 事件提供者发出一个事件,调用事件信道上的 push 方法,然后事件信道把数据"推"到使用者对象上,如图1所示。

push 型提供者 PushSupplier 的接口定义如下:

```
interface PushSupplier{
        void disconnect_push_supplier();
};
```

disconnect_push_supplier 用来结束提供者对事件的发送。

push 型使用者 PushConsumer 的接口定义如下:

```
interface PushConsumer{
        void push(in any data) raises(Disconnected);
        void disconnect_push_consumer();
};
```

push 函数用来将事件传递给使用者,函数 disconnect_push_consumer 用来结束使用者对事件的接收。

·pull 模型 事件使用者主动调用事件信道上的 pull 方法,然后事件信道把数据从提供者那里"拉"回来,如图2所示。

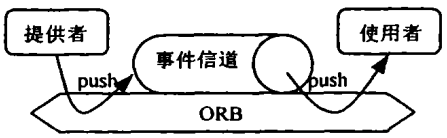

图1 push 模型

图2 pull 模型

*)国家863计划 CIMS 专题资助项目(项目编号2001AA415410);黑龙江省教育厅科技资金资助项目。李金宝 副教授,博士生,主要研究领域为并行计算,计算机体系结构,数据库。李建中 教授,博士生导师,主要研究领域为数据库,数据仓库,并行计算。于 玲 硕士研究生,主要研究领域为并行计算,网格计算,数据库。

维普资讯 http://www.cqvip.com

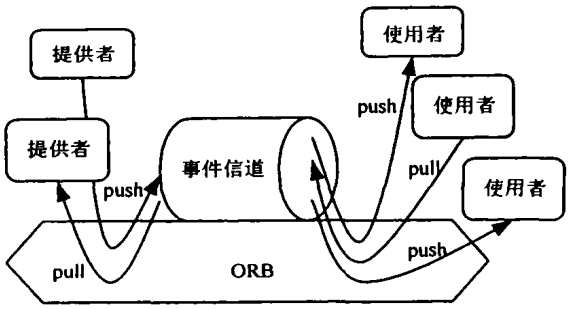

图3 pull 与 push 混合模型

pull 型提供者 PullSupplier 的接口定义如下：

```
interface PullSupplier{
        any pull() raises(Disconnected);
        any try_pull(out bollean has_event) raises(Disconnected);
        void disconnect_pull_supplier();
};
```

pull 函数用来获得事件，pull 函数以阻塞的方式进行，直到事件已经准备就绪或者发生异常为止。try_pull 函数用来周期性地询问事件，以非阻塞的方式进行，如果事件已经准备好，have_event 返回 true，事件作为结果返回，否则 has_event 返回 false。disconnect_pull_supplier 函数用来结束提供者对事件的传递。

pull 型使用者 PullConsumer 的接口定义如下：

```
interface PullConsumer{
        void disconnect_pull_consumer();
};
```

disconnect_pull_consumer 函数用来结束使用者对事件的接收。

事件信道可以提供 push 和 pull 混合通信方式，即事件信道可以用一种通信方式与提供者通信，又可以用另一种方式与使用者通信。事件信道可以让使用不同模型的多个提供者和多个使用者之间进行数据传递，提供者和使用者并不需要知道信道上其它对象具体使用哪种通信模型，如图3所示。

在事件服务中，提供者和使用者并不直接交互作用，而是通过从事件信道获得的代理 proxy 对象进行事件传递。事件信道的代理提供者对象分配器 SupplierAdmin 为提供者分配代理提供者，代理使用者对象分配器 ConsumerAdmin 为使用者分配代理使用者。

由于 push 模型相对于 pull 模型更节省通信资源，所以使用得更普遍。下面以 push 模型为例，简述提供者通过事件信道向使用者发送事件的过程，该过程如图4所示。

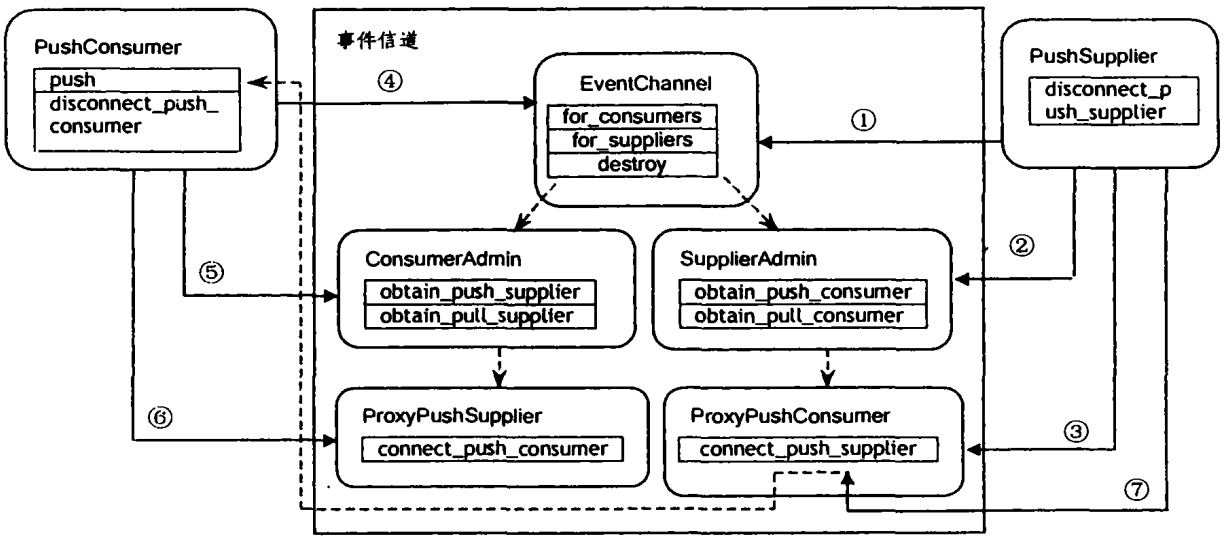

图4 push 模型工作过程

①提供者调用事件信道对象上的 for_suppliers 方法，获得一个代理提供者对象分配器 SupplierAdmin 对象。

②提供者调用 SupplierAdmin 对象上的 obtain_push_consumer 方法，获得一个代理使用者 ProxyPushConsumer 对象。

③提供者调用 ProxyPushConsumer 对象上的 connect_push_supplier 方法，与代理使用者对象建立连接。

④使用者调用事件信道对象上的 for_consumers 方法，获得代理使用者对象分配器 ConsumerAdmin 对象。

⑤使用者调用 ConsumerAdmin 上的 obtain_push_supplier 方法，获得一个代理提供者 ProxyPushSupplier 对象。

⑥使用者调用 ProxyPushSupplier 上的 connect_push_consumer 方法，与代理提供者对象建立连接。

⑦提供者调用 ProxyPushConsumer 上的 push 方法，事件信道将调用注册代理的所有使用者上的 push 方法，进行事件发送。

3 应用事件服务的分布式并行计算

CORBA 事件服务的异步特性使其在分布式并行程序中能得到很好的应用。在一个分布式并行计算程序中，采用事件服务的 push 或 pull 模型，调度程序通过事件信道为节点发送数据，数据的传送过程是非阻塞的，为一个节点分配任务后，不必等待返回结果，调度程序可以继续为另一个节点分配数据，这样就可以减少调度程序的不必要的等待时间，从而提高了整个程序的效率。

在一个具有 N+1个节点的异构分布式并行计算环境中，我们用其中的一个节点做客户(client)，其余 N 个节点做服务器(server1～serverN)。客户端要完成一个大规模计算任务。首先客户端将整个任务划分成 M(一般 M>N)个子任务，然后通过调度程序，将 M 个子任务分配给 N 个服务器(server1～serverN)来并行完成计算任务。调度程序的主要任务是为各服务器节点分配任务，并负责回收各个节点的计算结果。

维普资讯 http://www.cqvip.com

调度程序控制多个线程，其中包括一组数据发送线程(DataSendThread1～ DataSendThreadN)和一个结果接收线程(ResultReceiveThread)。数据发送线程通过事件信道(DataChannel1～DataChannelN)为对应节点分配任务，结果接收线程通过事件信道(ResultChannel)接收各节点返回的计算结果。每个节点上的工作进程控制三个线程：数据接收线程(DataReceiveThread)，计算线程(ComputeThread)和结果发送线程(ResultSendThread)。数据接收线程通过事件信道接收由调度进程发送的任务，然后交给计算线程，计算线程计算结束后再将计算结果传到结果发送线程，由发送线程将结果通过事件信道送到调度进程。数据传送过程采用 CORBA 事件服务的 push 模型。如图5所示。

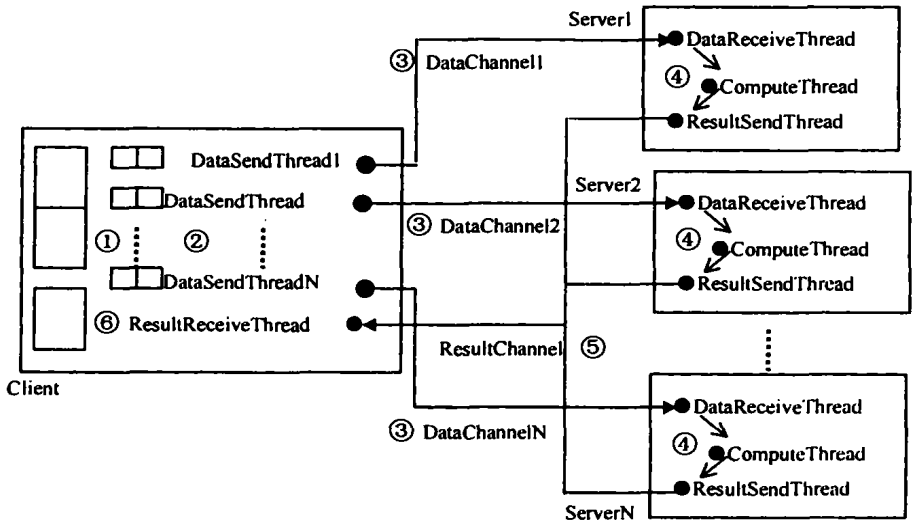

图5 分布式并行计算模型

数据分布过程中，调度程序中的 DateSendThread*i* 是 push 模型中的提供者，服务器 *i* 上的 DateReceiveThread 是使用者。数据回收过程中，服务器 *i* 上的 ResultSendThread 是 push 模型中的提供者，调度进程的 ResultReceiveThread 是使用者。push 方法中的 any 类型将使用数组类型。

下面以矩阵乘法为例，说明了 CORBA 事件服务在分布式并行计算中的具体应用。如图5所示。

①调度程序将待计算的两个矩阵分块，构成子矩阵集合；

②调度进程不断地取出子矩阵，依次轮流交给数据发送线程 DataSendTread1～DataSendTreadN，直至没有数据可分配；

③DataSendThread*i* 通过事件信道 DataChannel*i* 将数据 push 给 server*i* 的 DateReceiveThread；

④DateReceiveThread 接收到全部数据后，计算线程 ComputeThread 开始进行计算，此时，DateReceiveThread 可以接收下一块的数据；ComputeThread 计算结束后，将结果交给 ResultSendThread 进行发送，ComputeThread 可以计算收到的下一块数据；

⑤ResultSendThread 通过事件信道 ResultChannel*i* 将数据 push 给 ResultReceiveThread；

⑥调度进程将 ResultReceiveThread 收到的结果合并，并得到最终结果。

结束语 本文的分布式并行计算模型已在一个由7台异构 PC 机组成的局域网上实现，操作系统使用 Windows 2000 Server，CORBA 采用 IONA 公司的 ORBacus 4.1.3[3]，编程语言使用 C++。

实验采用了 CORBA 事件服务的 push 模型，该模型采用的是一种异步通信方式，所以数据的整个分布过程和结果的回收过程都是非阻塞的。实验结果表明，本文的分布式并行计算模型减少了不必要的等待时间，实现了计算和通信的重叠，并提高了计算的并行性。

CORBA 的事件服务为应用程序提供了异步方式的通信方法，弥补了同步方式在分布式并行程序方面的不足，但事件服务不能对服务质量做出保证[7]，也不能对事件作出区别，这在一定程度上限制了它的应用。

参 考 文 献

1 Aleksy M, Korthaus A. A CORBA-Based Object Group Service and a Join Service Providing a Transparent Solution for Parallel Programming. 2000 IEEE

2 OMG. CORBA Services Specifications. http://www.omg.org

3 IONA. ORBacus 4.1.3 Document OB-4.1.3.pdf. http://www.orbacus.com

4 (美)Orfali R, Harkey D, Edwards J 著，陈章渊，等译. 智能 CORBA. 北京：电子工业出版社，1999.11

5 刘辉，沈钧毅，林欣. 用 CORBA 创建电子商务系统. 北京希望电子出版社，2000.12

6 Otte R, Patrick P, Roy M. 著，李师贤，等译. CORBA 教程：公共对象代理请求体系结构. 清华大学出版社，1999.10

7 (日)Slama D, (美)Garbis J, (澳)Russell P 著，李师贤，等译. CORBA 企业解决方案. 北京：机械工业出版社，2000.1