

一种用于网络计算的接口模型 NCIM*

赵振平 杨培根

(南京大学计算机科学与技术系 南京 210093)

(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘要 分析研究了当前网络计算的传统模型,指出了它们在服务资源表示中的不足,提出了新的接口模型,能够精确、全面地描述计算资源所提供的服务,以期增进计算机之间的交互,使之能够自动地发现并共享网络上的软件资源。

关键词 网络计算,语义,服务质量,客户/服务器,远程过程调用。

中图法分类号 TP393

近年来,随着信息基础设施建设的飞速发展,网络计算的时代已经来临。通过网络相连的千万台计算机,能够完成单机无法完成的巨量计算任务。1994年,美国的一批研究人员利用1600台工作站联网计算,只花了8个多月的时间,完成了单机约1000年的计算量,首次破译了由RSA公开密钥加密体制发明人20年前公布的著名密文。^[1]Andrew. S. Tanenboun 在其《Computer Network》第3版^[2]中提到了一个有趣的“Chinese Lottery”方案:在全中国的电视机/收音机内安装一个专用芯片,使用穷举法破译由DES加密的消息,那么,对于用56位长度的密钥加密的密文,仅需1分钟就可以被协力攻破!^[3]

由于网络上的计算机可能具有各种不同的软硬件结构,为了使它们能够相互通信、共同合作,必须分而治之,设计通信协议软件,提供一个网络计算模型,为高层应用程序服务。日前已有许多分布式计算的标准,如DCE, CORBA等,设计了远程过程调用RPC(remote procedure call)、对象请求代理ORB(object request broker)、分布式组件对象模型DCOM(distributed common object model)等机制,为应用程序从单机到分布式发展提供了有力的支持。

但是,当前的标准没有对服务资源作全面描述,客户程序不能了解服务器上的计算资源的真正作用,致使服务资源具有专用性,只有事先知道特定计算资源存在的客户程序才会使用该资源;客户软件也只能使用事先就定好的服务资源。

在分析网络计算的传统模型及机制的基础上,我们提出了对服务资源进行语义描述和质量描述的新方案。计算机能够通过新型的接口发现服务资源,通过分析服务资源的语义信息,客户程序就能够在计算中自动利用新发现的资源;而服务资源的质量信息,使客户程序能够合理地安排参与计算的服务器及其计算量。

1 网络计算的传统模型及机制

网络计算的基本模型,是客户/服务器模型。客户和服务器是分离的逻辑实体,服务器进程是服务的提供者,客户进程是服务的消费者,客户和服务器通过消息传递机制耦合在一起,共同完成工作。根据服务器所提供服务的性质,客户/服务器结构可分为若干类,从最简单的文件服务器、复杂的数据库服务器、事务处理服务器,直到当今流行的组件服务器和对象服务器(如图1所示)。

网络计算涉及到不同体系结构的机器、操作系统和网络等大量复杂的细节,为了方便程序设计,人们设计了许多标准的高层机制,将这些细节隐蔽起来。现在,让我们来研究一下当前主要的网络计算标准中所提供的抽象机制:RPC, ORB和DCOM。

1.1 远程过程调用RPC(remote procedure call)

RPC机制^[4](如图2所示)把客户进程通过网络,向服务器进程发送计算请求,等待服务器的计算结果,这样一些

* 本文研究得到国家“九五”科技攻关项目基金和南京大学计算机软件新技术国家重点实验室基金资助。作者赵振平,1973年生,硕士生,主要研究领域为计算机网络与分布式计算。杨培根,1940年生,教授,主要研究领域为计算机网络与分布式计算。

本文通讯联系人:杨培根,南京210093,南京大学计算机科学与技术系

本文1997-07-07收到原稿,1997-10-20收到修改稿

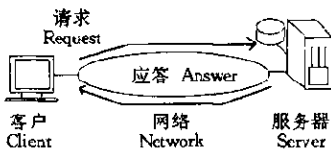


图1 客户/服务器模型

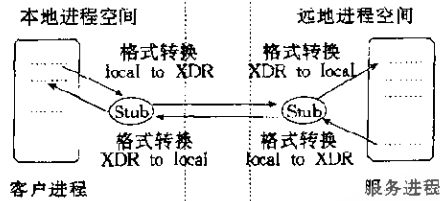


图2 RPC机制

复杂步骤抽象为一个过程调用,只不过该过程的真正执行体不在本地(所以称为远程过程调用),在本地的只是一个由编译器自动生成的Stub例程,该例程负责寻找服务器,向服务器的Stub例程传递请求消息(然后,服务器端的Stub例程启动真正的计算过程),等待它发回计算结果,再象通常的过程一样,将结果返回给本地的客户进程。

由于不同的CPU使用不同的数据结构(如整数的长度和字节顺序不同),为了使远程过程调用保持机器独立,必须有一个标准的数据格式表示方法。SUN的RPC使用了一种称为外部数据表示(XDR^[5])的中性格式。客户端的Stub例程自动将客户进程调用远程过程的参数转换为XDR表示,发往服务器,服务器端的Stub例程将这种格式转换成服务器的本机格式,计算完成后,把结果转为XDR格式发回,客户端的Stub例程再将其转换为客户端的本机格式。返回结果给客户进程。程序员只要使用RPC系统提供的接口定义语言IDL为自己的计算过程定义接口,其中给出了远程过程的名称和用XDR表示的参数,编译器会自动根据IDL文件建立客户端与服务器端的Stub例程。图3给出的是一个数学库的IDL文件。

```
[uuid(2654873F-FE65-3986-44DF7835F752),
  version(1.0)
]
interface MathLib
{
  void GetFactor([in] long A, [out] long factor);
  void GetGCD([in] long A, [in] long B, [out] long gcd);
  const BUFSIZE = 256;
  void root ([in] float Coefficients[(BUFSIZE+1) * 2],
            [out] float aRoot[2]);
}
```

注:其中声明了求因数、最大公约数和复系数代数方程的一个根3个过程

图3 一个数学库RPC服务程序的IDL文件

RPC机制及其提供的工具使程序员可以很容易地将单机处理的程序移植到分布式环境下。

1.2 对象请求代理 ORB(object request broker)

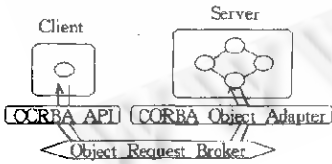


图4 ORB机制

对象式计算是当今的主流计算模式。ORB是对象管理集团(OMG)的CORBA规范中的一部分。ORB的目的是要在异构型的分布式环境下,使不同的语言生成的、运行在不同的硬、软件条件下的对象之间能够交互(如图4所示)。

同RPC的实现类似,ORB机制也提供了IDL用于定义服务端的对象。另外,ORB机制还提供了一种动态请求对象的办法,允许客户端在运行时给出所要请求的类、所执行的方法以及参数、动态联接对象。这样,客户端就可以查询服务器上的类,更加灵活地使用对象。

1.3 分布式组件对象模型 DCOM(distributed common object model)

Microsoft的DCOM(如图5所示)也是一个通用的软件对象互操作的体系结构模型。它也允许不同语言、不同平台的对象可以交互,合作完成计算。现在,该模型已经发展成为ActiveX参考模型。

DCOM规定对象之间的交互要通过COM接口进行。接口是一组相关的函数(也即对象的方法)的声明,它定义了一套服务;每个对象可以(并且通常)实现多个接口,以提供多套服务。

每个COM对象都必须实现IUnknown接口,该接口有一个方法——QueryInterface。通过该方法可以查询其他接口。如图6所示,一个对象实现了IStack接口,其中有Push、Pop方法。客户进程先通过IUnknown接口中的QueryIn-

terface 方法查询 IStack 接口,获得了 IStack 接口后,再通过 IStack 接口联接调用其 Push,Pop 方法.

DCOM 同样也提供 IDL 为服务器上的对象定义接口,由分别驻留在客户端和服务器的 DCOM 运行时的支持库来做参数转换等工作.

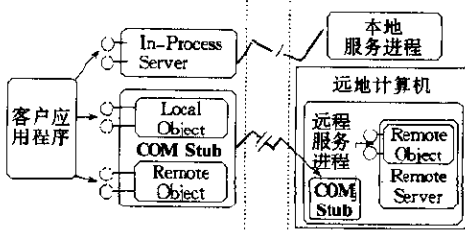


图5 DCOM机制

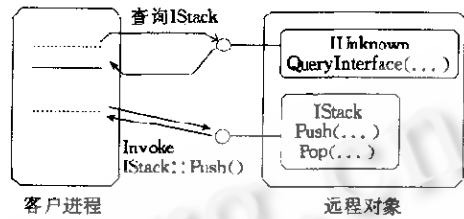


图6 DCOM中对象的动态联接

2 网络计算的新型接口模型

对上述3种广泛使用的机制仔细分析,我们发现它们有一个共同的特点:它们都以利用网络上机器的硬件资源为主,而对软件资源的自动利用没有提供足够的支持.类、方法、函数的名称及参数的名称只对人类程序员有用,因为编译器根据IDL定义文件自动生成的接口时,这些名称均已被转换成相应的GUID(全球唯一标识),客户端的程序无法从这些接口中了解服务器上所提供的服务的精确作用;即使对于程序员,名称也只能起到提示的作用,要求程序员查阅服务程序开发者提供的文档,才能领会所提供服务的的作用.例如,仅根据参数无法确知一个排序过程到底是顺排序还是倒排序,这样,驻留在服务器上的计算过程或对象基本上是为特定客户而准备的.ORB或DCOM中虽然能够由程序动态指定类名、接口名、方法名来联接对象完成计算,可若不是事先就知道某个类、接口的具体作用,这样的动态查询一个新类然后请求联接其方法的过程似乎是漫无目的的.当然,专门测试对象行为的程序除外.

由于服务方提供的接口中没有提供服务的语义描述,所以,客户端就不可能主动发现服务方的软件资源而加以利用.软件的开发耗费大量的人力,是人类智慧的结晶.软件资源的共享,从某种程度上讲,比硬件资源的共享更加重要.现代操作系统,如UNIX系统中,提供了大量独立的小巧工具,用户熟练掌握这些工具后,就可以通过组合它们来完成绝大多数日常任务,不必为特定任务而重复开发,从而大大提高了系统的使用效率.Internet的飞速发展,为人们提供了丰富的共享信息资源.如果发展一种方法,让计算机之间也能够互相理解对方所提供的服务的语义,从而自动地共享软件资源,就能使全球的计算机相互沟通,取长补短,构成一台计算能力超强的虚拟机,其前景将非常广阔(如图7所示).

为此,我们提出一种方案,把语义描述和质量描述引入服务的接口定义,相应地在本地操作系统中增加推理模块,就可以利用形式语义学和计算机软件的形式化开发方法的既有成果,实现主动、灵活的网络计算.

程序的语义就是指程序的作用,它是计算机语言的形式语义学所研究的内容,通过用数学的方法来描述程序的语义.它分为操作语义、指称语义、公理语义、代数语义4大流派.例如,从程序正确性验证的基础上发展起来的公理语义,给出了一种方法,能在给定的前提下,验证程序对于某种特定的性质是否成立.程序的这种前、后置条件能够精确地刻画程序的语义.如图3中求整数的因数的程序GetFactor,其前后置条件如图8所示.

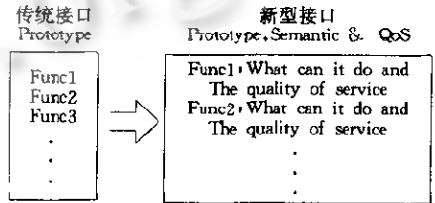


图7 新型接口模型

```
void GetFactor([in] long A, [out] long aFactor)
    precondition: {A>1}
    postcondition: {(aFactor>1) && (A mod aFactor==0)
        && (if (b>1) && (A mod b==C)
            then b>=aFactor
        )
    }
```

图8 前后置条件定义的GetFactor的语义

根据前后置条件可知,GetFactor 将求出输入 A 的第 1 个大于 1 的质因子.

对于一个庞大的程序来说,要用形式化的方法推导出它的语义是极其困难的.幸好,在这里我们不需要推导服务资源的语义,因为采用形式化的科学方法开发程序时,从一开始,就要给出过程的前、后置条件,它们就刻画了过程的语义.

服务的语义描述中需要指出计算过程或方法的返回结果与输入参数之间的关系,以及计算过程或方法之间的关系.在接口中引入语义描述就可以很好地同程序开发的形式化方法结合起来.

其他的语义表示方法还有λ演算、组合子、π演算等.如同已经建立的参数表示的标准一样,形式化的语义描述也应当建立一个标准,这样,计算机之间的相互理解才有可能.可以选择现有的形式描述语言,如 Z 语言等加以扩充,用作接口的语义描述语言.为了简化语义的描述,需要给出基本概念的定义和符号表示的约定,例如:逻辑关系“或”、“与”、“蕴涵”的符号表示;几何学中的基本关系“属于”、“介于”、“合同”的记号表示以及公理都应作为国际标准给出,这样才能据此进行推理.语义描述的标准本身应当可以从网络上获得,可以参照现有 Internet 上用于定位主机的域名系统 DNS 来设计语义描述标准的查找系统;为了方便检索,语义标准中可以划分成一些域(如图 9 所示).它不应当是静态的,而是不断地发展,增添新的领域、新的概念和经过证明的定理,连网的计算机可以随时获取最新的信息.

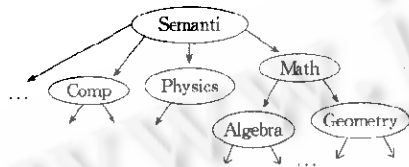


图9 语义描述的领域划分

服务的质量描述(QoS)中可以指出计算的复杂度、计算结果的精度或可靠程度等.客户获得了服务质量的信息,就可以估计计算开销,然后与通信开销权衡,合理地安排参与计算的服务器,均衡它们的负载,使整个计算能更快、更好地完成.对于高可靠性要求的计算,可以分别交由不同的服务器独立进行,客户方校验其结果,提高计算结果的可信度.

操作系统中的推理模块进行形式逻辑计算,该模块是新型接

口模型能发挥作用的键(如图 10 所示).它由谓词逻辑计算部分和知识库构成.系统空闲的时候,该模块就可以在网上传询,和其他计算机交换信息,了解新的标准及周围可达的服务器所能提供的新的服务.所获得的信息被分类存储于知识库中.由于可以精确地了解服务的语义,推理模块将能够自动作一些灵活的处理,例如,解决服务的计算实质符合客户要求,但参数并不匹配的问题;还可以利用管道连接若干服务,实现新的功能.如同人类通过互连网络获取信息一样,计算机通过这种方式了解世界,不断增强自身处理问题的能力.设想一下,一台计算机通过网络了解到 DeepBlue 具有非凡的下国际象棋的能力后,普通人要想再胜过它就很难了.

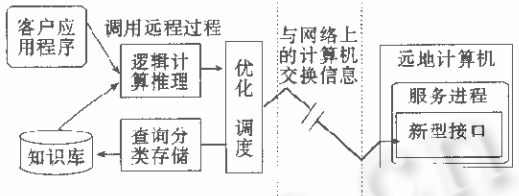


图10 OS中的推理模块

3 实验研究

根据上述思想,我们在国产操作系统 COSIX, UNIX SVR4. 2, Windows NT 和 Windows95 四种操作系统平台上进行了初步的实验.

我们把系统的 RPC 服务加以扩充,构成 XRPC 服务.XRPC 在 RPC 服务的基础上,扩充了接口定义语言,增加了 Sem2Proc 语义查询系统,运行库提供了语义注册、可扩充的语义查询和任务分解调度支持,其结构如图 11 所示.

驻留在服务方主机上的 xrpcbind 程序接受服务方过程的语义注册和 Sem2Proc 查询系统的查询,除此之外, xrpcbind 还负责收集本机的运行时信息,如负载率、空闲内存等,响应运行库的查询.

Sem2Proc 查询系统的设计类似 Internet 上的域名系统,由客户机上的 Resolver 和查询服务器上的 Server 构成. Server 部分收集本子网的信息,并和其他服务器通信交换信息.查询部分可动态装入用户扩充的分析库,现已实现的基本分析库可以检索到语义描述形式相同,而输入参数顺序可能不同的所有服务,将服务标识、参数调用顺序和服务的提供主机返回给查询者.

扩充的接口定义语言 XIDL 将服务的语义描述加入其中.我们采用置标的方法,如 Web 上通用的 HTML 语言那样,描述体系可以用 Tag 标记,语义分析程序遇到未知的 Tag,就把该 Tag 括起的内容略去.XIDL 预处理器抽取语义描述,生成注册代码,然后调用 RPC 的 IDL 处理程序,客户方和服务方过程的代码与这些处理程序生成的代码联编,

生成相应的可执行程序。服务方程序启动初始化时，将向 xrpcbind 注册；客户方程序请求执行远程过程时，XRPC 运行时库根据所请求的服务方标识号，向 Sem2Proc 系统查询。当查询返回多个可用的服务方(包括多个主机提供同一服务的情况)时，XRPC 运行时库将能够把计算任务分派给这些服务者。XRPC 中可动态地装入任务分派库。现已实现的基本分派库分析语义描述，若发现对整数域的不完全穷举搜索(语义描述中为 semiscan from V_1 to V_2 result $V_3 \{ \dots \}$, V_1 , V_2 为输入变量; V_3 为 Boolean 型输出变量, 指示是否搜索到解), 就可以将搜索域划分成

一些子域, 把原先的任务分解成规模较小的子任务, 分派给不同的服务者并行执行, 汇总结果。对不完全搜索, 只需将最先搜索到的结果返回; 仅当各个子任务都没有搜索到解时, 才将无解作为整个任务的结果返回。

进行任务分派时, XRPC 运行时库向可用服务所在的主机发送消息, 这些主机上的 xrpcbind 将本机的状态, 包括所请求的服务过程是否就绪、系统负载等, 返回给任务分派者。若任务可以分解, 基本分派库根据已就绪的服务方的系统负载分配任务量; 若任务不能分解, 则挑选最佳服务者执行。

XRPC 系统与现有的 RPC 系统相比, 具有如下优点: (1) 当客户方程序所请求的服务方未就绪时, RPC 系统计算会失败, 而 XRPC 系统可以寻找能够替代的服务方进行计算; (2) 通过分析服务的语义, 如果识别出可以并行的成分, 就能够将任务分解, 交给多个服务者执行, 提高计算的效率。

XRPC 系统相对于 RPC 系统的缺点是, 进行计算的初始通信开销增大, 需要查询服务语义以及要求服务方报告其所在主机的系统状况。考虑到网络计算的计算量往往较大, XRPC 系统还是可以接受的。

我们当前实现的 XRPC 还很简单, 语义查询和任务分派都比较粗糙。如前所述, 一个完善的系统, 要具有服务质量描述、语义描述标准本身的动态更新等; 任务分派时还必须了解承担任务的机器的硬件性能指标; 另外, 语义描述的认证问题也非常重要。要构成实用的系统, 有待形成一个国际标准, 这样才能够使语义描述被全球的计算机所认同。必须综合应用人工智能、知识工程等相关学科的成果, 才能更好地应用语义信息, 充分地利用软硬件资源。

4 结 语

据《PCWEEK》报道, BellCore's Internet Architecture Research Lab. 的首席科学家 Christian Huiteman 进行了一项调查, 结果显示: 截至 1997 年 9 月, 以永久连接方式接入 Internet 的主机已达 2 600 万。这些机器的平均空闲时间按 8 小时计算, 将能够提供惊人的计算能力。只要通过在服务的接口定义中引入完善的语义描述和服务质量描述, 任何一台联网的计算机将能够通过接口了解周围的计算机世界, 充分利用全球计算机的庞大计算能力, 为人类更好地服务。

参考文献

- 1 Stang D J, Moon Sylvia (著), 程佩青等(译). 计算机网络安全奥秘. 北京: 电子工业出版社, 1994 (Cheng Pei-qing et al(translation). Secrets of Computer Network Security. Beijing: Publishing House of Electronic Industry, 1994)
- 2 Andrew S Tanenbaum. Computer Network(3rd Edition). Englewood Cliffs: Prentice-Hall International Inc., 1996
- 3 Quisquater J J, Girault M. Chinese Lotto as an exhaustive code-breaking machine. IEEE Computer Magazine, 1991, 24(11): 14 ~ 22
- 4 RFC1831. Remote Procedure Call Protocol Specification Version 2. 1995
- 5 RFC1832. XDR: External Data Representation Standard, 1995

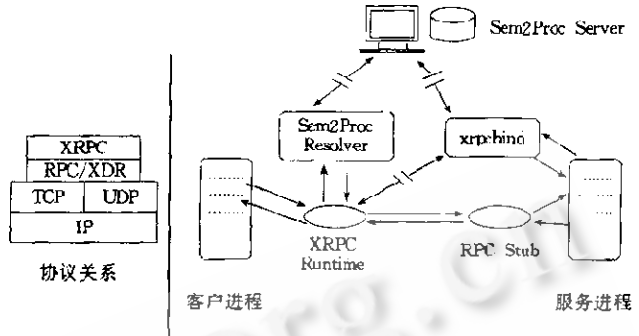


图11 XRPC系统结构图

An Interface Model for Network Computing

ZHAO Zhen-ping YANG Pei-gen

(Department of Computer Science and Technology Nanjing University Nanjing 210093)

(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

Abstract After the modern models for network computing are analyzed, their weak spots of the representation of serving resources are pointed. And a new interface model which can specify a computing resource more precisely and comprehensively is designed in order to improve the interaction of computers themselves so that the software resources can be searched and shared automatically.

Key words Network computing, semantic, QoS, client/server, RPC(remote procedure call).