

## 开放分布式处理中的交易服务\*

龚俭

(东南大学计算机科学与工程系 南京 210096)

TP301

**摘要** Trading is the fundamental function of open distributed processing systems to compose an open cooperative system. It is a kind of improvement of network directory service and binder service in RPC, and has more comprehensive formal semantic description capability. The basic functions of trading service is introduced in the paper, and its working mechanism and required operations are discussed as well.

**关键词** ODP, Trading, Trader.

开放分布式处理, 交易, 并行过程

大规模互连网络的出现及其与并行工程、远程教育等新型分布式应用的结合, 已开始迅速地影响现实的生产活动和社会活动。这种结合促使分布式应用的设计与实现从全局模型方式转向集成方式,

即分布式应用的处理元素以标准的形式进行交互合作, 并支持一致的语义, 这就是开放分布式处理的含义。开放分布式处理(ODP)的基本参考模型为开放的分布式应用提供了一个体系结构框架, 以支持系

获得较大的性能提高, 所以关键在于如何选择最优的循环展开向量。

**2.2.4 如何确定最佳循环展开向量** 确定一个最优展开向量  $U = (u_1, u_2, \dots, u_k)$  这本身是一个组合优化问题, 虽然问题空间是  $k$  维离散的, 但穷尽搜索是不可能的, 所以必须利用一些优化启发信息。不管使用哪一种优化搜索方法, 关键在于选择目标函数, 以及如何选定  $U$  的初始值。

如果把  $D_{max} = (g_1, g_2, \dots, g_n)$  定义为一个循环任务图的最大距离向量, 其中  $g_i = \max\{d_i\}$  ( $d_i$  为 DP 中  $D$  的第  $i$  个分量), 那么  $D_{max}$  也是展示所有隐藏的循环传递相关的最小展开向量, 所以可把  $D_{max}$  作为最优搜索的循环展开向量初始值  $U_0$ 。

设  $x_i$  为执行第  $i$  层循环迭代一次所需时间, 那么  $x_n$  为执行一次  $G_n$  的时间,  $x_1$  为执行一次最外层循环所需时间。设  $y_i$  为第  $i$  层循环的两个相继迭代之间的通讯延迟。Rewini<sup>[20][21]</sup> 给出了目标函数  $F = \left( \prod_{i=1}^n \frac{1}{u_i + 1} \right) (x_n + \sum_{i=1}^n y_i)$ 。至于参数  $x_i$  和  $y_i$ , 把任务图  $G_n$  调度到某一目标机器上时,  $x_i$  就是所获调度结果的调度长度。另外, 根据调度所产生的 Gantt

图, 我们可得到第  $i$  层循环的相继迭代之间的通讯延迟  $y_i (1 \leq i \leq k)$ 。

## 参考文献

- [1] 陈华平、李京、陈国良, 并行分布计算中的任务调度问题(一), 《计算机科学》, 24(1)1997
- [2] 陈华平、陈国良, SMPD 模式下的任务调度问题, 《计算机科学》, 23(6)1996
- [3] D. Towlsley, Allocating programs containing branches and loops within a multiple processor system, IEEE Trans. on Soft. Eng. SE-12(10) 1986
- [4] H. El-Rewini and H. Ali, Scheduling conditional branching using representative task graphs, Journal of Combinatorial Mathematics and Combinatorial Computing, 10, 1991
- [5] H. El-Rewini and H. Ali, Scheduling parallel programs containing conditional branching, Journal of Parallel and Distributed Computing, 1994
- [6] V. Saletore et al., Scheduling non-nmform parallel loops on distributed memory machines. Proc. HIC-SS-26, Maui, Hawaii, 1993
- [7] H. El-Rewini and H. Ali, How many times should a loop be unrolled?, Technical Report, Dept. of Computer Science, University of Nebraska at Omaha, 1993
- [8] H. El-Rewini and T. Lewis, Schedule-driven loop unrolling for parallel processors, Proc. the 24th Hawaii Intl. Conf. on System Science, Kauai, Hawaii, January, 1991

\* 本文的研究内容受江苏省青年科技基金课题(BQ94002)资助。

统的集成<sup>[1]</sup>。

按照 ODP 的概念,分布式处理的各处理元素可以不予预先指定,而是根据一定的合作条件动态地集成在一起。这些合作条件通过 ODP 的企业观点、信息观点和计算观点来刻画,而它们的使用(定义、检查和协商)则由 ODP 的交易功能来实现,交易功能以增量的、动态的和开放的方式提供“服务提议(Service Offer)”和“发现服务提议”的手段。本文将在介绍 ODP 交易功能基本概念的基础上探讨交易服务的实现原理和实现结构。

## 1 交易的基本概念<sup>[2]</sup>

在一般的意义下,交易是双方或多方经协商(讨价还价)而达到一致的过程,它基于某种需求,用口头或书面的形式表述,具有一致同意的含义,并可能需要中介介绍人和具体的交易地点。ODP 的基本参考模型中借用了交易的一般概念,按照 ODP 的观点,交易的目的是把不同的 ODP 系统结合起来以形成一个针对某个特定应用的共同体或联邦。

参加交易的基本元素称为活动,它是一个遵从 ODP 概念的独立系统的抽象,也是构成整个分布式处理的基础。共同体是活动的集合,是一个完整的系统,完成特定的分布式处理需求。共同体可以递归定义,所有包含共同体的共同体称为联邦。共同体和活动均可抽象为可嵌套定义的域,因此交易可用域及域中所包含的元素(称为对象)作为讨论的基础。对象是资源的抽象,它可以代表数据,也可以代表处理功能(服务)。按照交易的要求,域中的对象应分为三类:(1)私有对象,仅属于共同体中的某个活动(称为用户域);(2)系统对象,可移出和移入的对象,它们构成交易域的元素;(3)共享对象,从交易域中移入的(外来)对象。移出的对象同时属于用户域和交易域,然而对象的控制权总是属于用户域,因为实际移出的只是对象的描述,而不是对象本身。

域中的交易功能由交易者(trader)提供,通过交易者移出或移入它们的服务。移出时活动向交易者提供服务的描述(类型定义)和服务的位置(通过一个特定的接口);移入时活动向交易者提出所需服务的特性,交易者将其与可提供的服务描述进行比较,选取适合者并将其位置通知申请的对象。

在大规模网络中,交易者会有多个,每个交易者都有自己的一批用户,形成一个交易共同体,并受一个交易策略控制。当用户的需求在自己的共同体中得不到满足时,可直接或间接地(通过本共同体的交

易者)访问其他共同体。用户只固定地与本共同体的交易者联系,跨共同体的交易必须通过交易者,由其代理进行或通过它透明进行。因此交易功能可概括为:移出服务,移入服务,交易者之间的交互工作。

交易共同体是递归定义的,共同体中的活动可担任下列角色(可兼任):

- (1)交易者:负责接受服务的移出请求并响应服务移入请求的对象。
- (2)移出者:向交易者移出服务的对象。
- (3)移入者:从交易者处获得服务移入的对象。
- (4)交易管理者:贯彻交易策略的对象。
- (5)交易策略制定者:制定并管理交易者的交易策略的对象。
- (6)服务提供者:服务描述的维护者。

整个交易环境中可包括多个交易共同体,因此一个活动可同时属于多个交易共同体,一个交易共同体中的交易者可同时是另一个交易共同体中的移入者或移出者。一个共同体可跨越多个域。移出者可移出在自己的接口提供的服务提议,也可移出别的服务提供者的服务提议。移入者可为自己或为别人移入服务。交易共同体中的交易管理者对自己的交易者有完整的控制权。在移入服务时要求进行计算接口调用结构类型的检查,另外还可对其子类型或超类型关系、行为兼容性和环境变量进行检查,交易者有责任及时清除过时的服务提议,例如可在服务提议接受策略中要求服务提议规定失效时间,或者在移入动作策略中要求服务提议的有效时间。

发起组成共同体的活动在自身用户域的基础上,根据需求并利用交易功能在交易域中寻找合适的系统对象,然后在一个交易协议的支持下与拥有这些系统对象的活动协商使用权,从而形成合作关系,并一起构成相应的共同体。这个共同体中各活动之间合作关系的正式化表述记为一个合同,它定义了共同体存在的条件和共享对象的使用约束。每个活动都有发起组成共同体的能力,因此一个活动可以为响应另一个活动组建共同体的请求而发起一个新的组建共同体的请求,其结果是构成联邦。

## 2 交易的依据

交易是从目录服务演化而来的。传统的域名服务器将一个正文名映射成一个物理地址;若将正文名抽象为接口定义,则需使用 Binder 来描述服务可提供的操作以及操作的变量和结果(这为远程过程调用的实现中所经常使用)。交易服务可看成是

Binder 的进化,它允许用户在移入服务时具有一定的灵活性,即不要求精确的匹配,同时使用更为复杂的交互协议。交易是分布式计算与面向对象技术相结合的产物,后者的封装性、抽象性和多态性也是交易的重要性质。

由交易共同体构成的交易环境中包含有:

(1)服务提议集,即交易域。服务提议由服务提议标识符加以区分,它们是唯一和无二义的,因此同一个服务提议若移出给不同的交易者,则应使用不同的服务提议标识符予以区分,从而可对不同的使用者定义不同的使用权限。

(2)交易者(称为结点 node)集。它们是交易操作的执行者,并各自拥有一个服务提议子集。

(3)弧集。表明结点之间的合作关系,这些有向弧构成交易图,对这些弧可分别定义不同的性质以控制查找(即移入)的传播。

交易的依据有移出者提供的服务提议、移入者提供的查找请求、共同体的交易管理策略、交易者使用的各种标准和约束等。

服务提议包括服务描述、服务提议标识符和接口标识符,还可包括一些服务提议性质。服务描述为一个接口调用结构类型,还可包括一组服务性质。服务提议标识符在交易共同体内是唯一的。服务提议性质描述了服务提议的特性,以及 binding 行为的规范。

查找请求是运用于特定查找动作的移入者策略规范,它可包括移入者的匹配、优先和范围标准,以及所需服务的接口调用结构类型。所有满足移入者匹配要求的提议的接口调用结构类型必须是指定接口调用结构类型的子类型。

共同体的交易管理策略包括:与服务移出有关的移出策略;与服务移入有关的移入策略;在交易活动期间对冲突的规则进行仲裁的仲裁策略;限制交易者接受服务提议的服务提议接受策略;与类型规范和类型间关系有关的类型管理策略;标识放置新服务提议的位置的提议放置策略(即如何实现服务提议的分布);以及用以指导在交易系统中寻找合适服务提议的查找策略。

交易者使用的各种标准和约束有:匹配标准,指出筛选时必须满足的条件;优先标准,定义在满足条件的服务提议中确定选择次序的规则;范围标准,定义匹配标准筛选范围的规则;匹配约束、优先约束和范围约束,分别定义由交易系统的策略对匹配标准、优先标准和范围标准所施加的约束。

### 3 交易的操作

交易功能包括两个方面,交易的实施和交易的管理。交易的实施通过交易操作来实现,而交易的管理则为交易操作提供环境支持。交易的实施至少应包含下列操作:

(1)移出 将新的服务提议加入至服务提议集,与一个结点相联系,并由交易者产生一个服务提议标识符。

(2)移入 从交易者获得一个(或一组)服务提议。该操作应分两步:首先根据查找请求在范围标准的限制下对交易环境进行查找,返回满足匹配标准的服务提议集;其次按照优先标准对返回的服务提议集进行排序,排序的标准是移入者的优先标准和交易者的约束条件(如交易者优先约束、仲裁策略等)的结合。

(3)撤消 从服务提议集中撤消指定的服务提议,并解除它与结点原先存在的联系。

(4)修改服务提议 根据服务提议标识符,用新的服务性质和新的服务提议性质来修改原有的服务和提议的性质。由于要保留服务提议标识符,所以该操作不是简单的撤消和移出操作的组合。

根据如上所述的交易原则,与交易管理有关的操作应包括:

(5)加弧 在交易图中指定的两个结点之间加一条弧,并定义该弧的性质,以表示一方可使用另一方所拥有的服务提议集;这种使用是单向的。

(6)撤弧 从交易图中移去一条弧及其相应的性质,即解除两个交易者之间的合作关系。

(7)改弧 修改一个指定弧的性质,即调整两个交易者之间的合作关系。

(8)加点(初始化操作) 在交易环境中增加一个新结点,即增加一个新的交易者。为保证交易管理操作的简单性,新增加的结点在一开始不包含任何服务提议,且与其他结点无合作关系(即是图中的一个孤立点)。

(9)撤点(结束操作) 从交易环境中删除一个结点,要求被删除的结点已没有服务提议,且不存在与其它结点的合作关系,理由同(8)。

根据 ODP 基本参考模型的规定,活动在与交易者进行交易操作之前应首先建立彼此的联络(Liaison),即建立起共同的应用上下文,因此,交易操作的实现需要一个专门的交易协议的支持。另外这个

(下转第 65 页)

### 4.3 界面

用户界面采用多窗口、多级菜单的形式。主菜单的选择项应是环境的各子系统。每个子系统也应有相应的用户界面。

软件界面要解决与其它语言或系统的沟通问题,有许多技术细节。作为 Tuili- I 程序的组成部分的过程基,其内容是外部语言 C 定义的函数集。通过 C 语言,又可调用诸如系统命令、窗口、图形等软件系统。因此,软件界面的部分功能已由 Tuili- I 实现系统得到了解决。需进一步考虑的是拓宽沟通方式,建立与一些常用数据库及图形、窗口软件的接口,以及在 Windows 环境下,别的软件系统调用 Tuili- I 系统的问题。

### 4.4 应用系统和原型系统集

应用系统(或应用原型)与实际领域密切相关,有的系统本身就是一个独立和完整的工作,每个应用系统离不开与领域专家的合作。

多媒体水库、水电信息系统将用于管理我国大型水库和水电资源。数据库中,不仅有关于水库和水电站的各种数据,而且对每个水库、水坝,都有它的外观彩色图、平面布置设计图、断面图以及相关的关系图表。这样,在表示一个水库或水电站的数据元组里就包含了图象和图形这样的多媒体信息。系统提供多种查询方式(如,按政区、流域、库容、坝高等)和统计方式以及其它多种功能。查询结果(包括图形、图象)和统计报表均可输出。显示图形和图象时,用户还可对其进行放大和缩小。除了用于管理和决策外,也为水电领域的科研和工程技术人员的具体设计工作提供了有力支持。这一系统的原型系统已经完成。

气象预测预报 ES 是一个有一定适用性的降水预报系统。众所周知,对天气做出准确预报并非易事,及早预报灾害性天气意义重大。气象预报涉及许多专业知识,既有一般性规律,又有很强的地域特

点。我们研制的这一系统基于与北京气象学院及有关气象部门的以往合作结果。经过适当修改某些知识库,或者通过询问用户方式获得有关的地域情况及其特点的知识,能在一定范围内作因地制宜的降水预报。

动物分类系统是一个原型系统<sup>[10]</sup>。分类本是框架/基于对象的特长。在表 1 中,我们看到它们在已有的一些 ES 中占有一定的比例(约 28%)。近年推出的一些工具和环境,也都突出了在面向对象方面的特性。动物识别分类系统的设计运用了 Tuili- I 的属性描述和属性继承特性,从中也可看出 Tuili- I 语言系统在描述概念的层次结构和分类判定树方面的独到之处。

五子棋对奕属于智能系统,它由陆汝钫教授用 Tuili- I 写成,取名《五朵金花》。系统有二百多条规则,二十四条规则基。由于是与人对弈,并且要进行多个回合,所以对系统的时间和空间效率要求较高。一方面,该系统能为开发复杂应用系统提供一个学习样品,另一方面,以轻松和人们较为感兴趣的方式,展示 Tuili- I 系统及其环境的特点。现正在 Tuili- I 及 MIDE 环境下将其实用化。

结束语 多媒体推理环境 MIDE 的实现工作目前已取得重要进展。由 MIDE 环境各子系统的基本功能不难看出,对环境中一些主要的子系统的实现,一个可行、有效的途径是充分利用 Tuili- I 的系统资源,编译和运行支持系统中的某些组成部分可直接或经改造后间接用于它们的实现。其它子系统的实现工作,如多媒体数据库管理系统的实现已经完成,界面系统和应用系统等,现正在良好的基础上进行。

致谢 衷心感谢陆汝钫、周龙骥教授多年来的帮助和支持。感谢本室陆维明教授、柴兴无、徐建礼博士以及北京气象学院陈永义教授、水电部北京勘测设计院研究所刘永川高工对本文工作的贡献。(参考文献共 20 篇略)

(上接第 30 页)

协议还应能支持交易的不同特性,如交易的可靠性、安全性等。结点或弧的撤销是否会影响共同体的存在也应由有关的活动根据共同体存在规则和交易协议,按具体的合同内容判定。限于篇幅,交易协议的具体内容不在本文中讨论。

结论 ODP 的交易功能是支持开放合作的基本功能。本文探讨了交易服务的依据和应提供的操作。交易服务适合于在一种开放的和广泛连网的环境中动态构造各种虚拟组织以支持 CSCW 应用,如

虚拟学校、虚拟企业或虚拟实验室。进一步的工作包括研究 ODP 计算观点下活动之间的接口规范语言,为交易服务所需的服务描述、接口调用结构描述、服务的移入/移出策略、仲裁策略等提供形式化的描述工具,从而为交易服务的实现打下基础。

#### 参考文献

- [1] 龚俊,开放分布式处理的基本参考模型,《计算机科学》, No. 4, 94 年
- [2] ISO/IEC DIS 13235-ODP Trading Function, 1996 年 2 月