

网格环境下基于虚拟组织的调度模型^{*})

刘会斌 都志辉

(清华大学计算机科学与技术系 北京100084)

摘要 虚拟组织是指在个人、机构或团体间一种灵活、安全、平等的资源共享动态集合,这种共享更强调直接对计算机、软件、数据以及其它资源的直接访问。管理虚拟组织资源、协同工作来解决大规模的计算是一个复杂的问题。提出了一个基于虚拟组织的资源分配和任务调度模型,为资源的管理和应用程序的调度建模。网格服务提供者向虚拟组织注册,并发布它能够提供的服务,任务调度器根据提供者属性分解任务并采用特定的算法将任务分配给不同的服务提供者。这个框架的模型侧重研究如何调度任务,根据任务的属性合理分配资源来解决复杂计算问题。

关键词 虚拟组织,模型,网格,调度

Scheduling Model Based on Virtual Organizations for Grid Environments

LIU Hui-Bin DU Zhi-Hui

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Abstract Virtual organizations refer to flexible, secure, coordinated resource sharing among dynamic collections of individuals, organization or institutions. The resource sharing is more concern about interoperation among computer, software, data and other resource. Allocating resource and cooperate scheduling applications to solve large-scale problem are a challenging task. A framework used for model resource management and task scheduled is proposed. Participants register themselves as Grid Service Providers, publish the resources or application services that they interested in sharing, scheduler resolves the task according to the information of registering and assign the task to different provider. This framework aim to study how to schedule task and allocate resource according to properties of the task.

Keywords Virtual organizations, Model, Grid, Scheduling

1 引言

网格^[1]和P2P^[2]技术的发展为解决大规模的分布式计算提供了解决方案。通过创建虚拟组织(VO)^[3]来共享和整合数以百万计的资源。虚拟组织是指在个人、机构或团体间一种灵活、安全、平等的资源共享动态集。共享不仅仅是简单的文件交换,更强调直接对计算机、软件、数据以及其它资源的直接访问,这种需求在工业、科学以及工程界等许多领域都会遇到。网格中每个VO的建立都有其需求和目的,在很多情况下,VO管理的服可以理解为地理位置分布、逻辑位置相邻、属性相似的服务集合,通过这些VO来为用户提供特定的服务。已经开发了一些网格项目来解决资源共享调度问题。例如,SETI@home^[4]项目通过使用互联网上计算机来寻找外星生命,通过使用地理上分布的空闲资源来解决复杂的计算问题。Legion^[5]提供了一种软件基础设施,通过它可以实现异构、地理分布的高性能计算系统的无缝交互。本文提出了网格环境下基于虚拟

组织的调度模型,和其他系统不同,这种框架侧重于根据资源提供者的属性,来调度工作任务,提供了网格节点(资源的提供者)协同工作的机制,合理调度用户的应用,完成大规模计算任务。

2 基于虚拟组织的调度模型的结构

这种框架结构有两个主要的实体:资源登记服务器^[6]和任务调度器。通过这两个实体操作以及和网格服务提供者相互作用实现工作任务的调度。网格的服务提供者按照需求和兴趣选择加入VO,使VO提供大量相似类型服务,解决特定的任务。虚拟组织的各节点向登记服务器注册,声明可提供的服务。资源登记服务器负责记录网络各个节点的可利用的资源的属性,例如,哪些节点能提供高性能计算资源,哪些节点能提供大容量的存储等,并负责把这些节点信息存储到资源库中。任务调度器负责接收用户的应用,向登记服务器查询虚拟组织各节点所提供服务的状态。运行任务调度分配算法将工作任务分配给各个节点。实体的相互作用见图1。

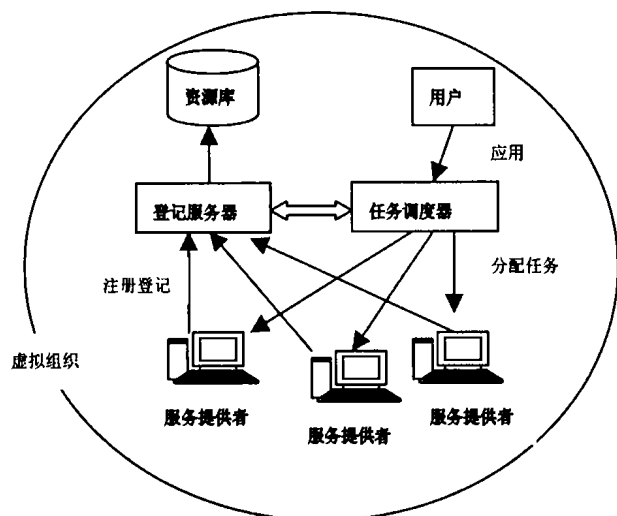


图1 虚拟组织实体的互操作

基于虚拟组织调度模型共分4层,用图2表示。第一层是构造层,指通过互联网连接地理上分布的计算设备。第二层是网格中间件层,包括 Globus^[7]和网络安全组件,Globus 提供了资源的发现和通信等服务。第三层是核心层,由登记服务器和调度服务器组成,提供资源的注册、调度等功能。最高层是用户的应用程序。

第四层	应用层	应用程序
第三层	核心层	登记服务器、任务调度器
第二层	网格中间件	Globus
第一层	构造层	Internet

图2 网络环境下虚拟组织分层结构

3 登记服务器结构

登记服务器结构用图3表示。关键的组件包括注册管理器和查询服务器。注册管理器登记服务提供

者状态信息,包括可提供的资源、资源使用时间等等。查询服务器为任务调度器提供查询服务,任务调度器根据可利用的资源,来分解工作任务并分配给资源提供者来执行。除此之外,登记服务器通过资源库记录服务提供者身份信息和所提供的服务。

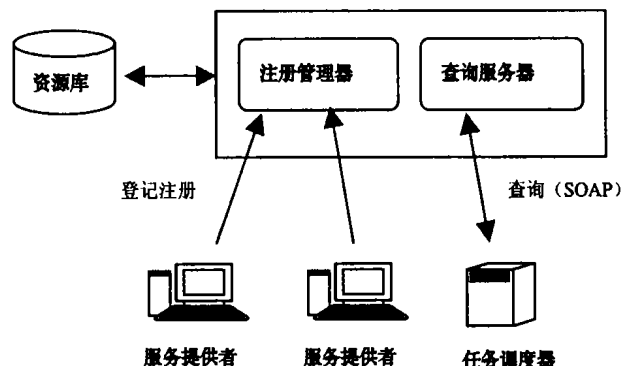


图3 登记服务器的结构

3.1 注册管理器

注册管理器结构用图4表示。它提供了两种不同的访问接口:提供者的管理模块和服务的管理模块。提供者管理模块负责记录、管理提供者身份信息,例如,提供者的名字、口令、联系地址和一些其他信息,并支持服务提供者注册和注销。服务提供者的身份信息在注册时被创建。服务管理模块记录服务提供者所提供服务的属性,由每一个注册的提供者动态地产生,提供者通过服务管理模块来增加、更新和删除它所提供的服务。基本的服务属性包括:服务名、服务类型、服务提供的截止时间、节点主机名和应用路径。除此之外,安全的问题也由注册管理器来管理。服务管理和提供者管理采用一种登录授权机制来识别服务的提供者。服务的修改操作在提交给资源信息库之前必须要经过提供者认证模块认证。

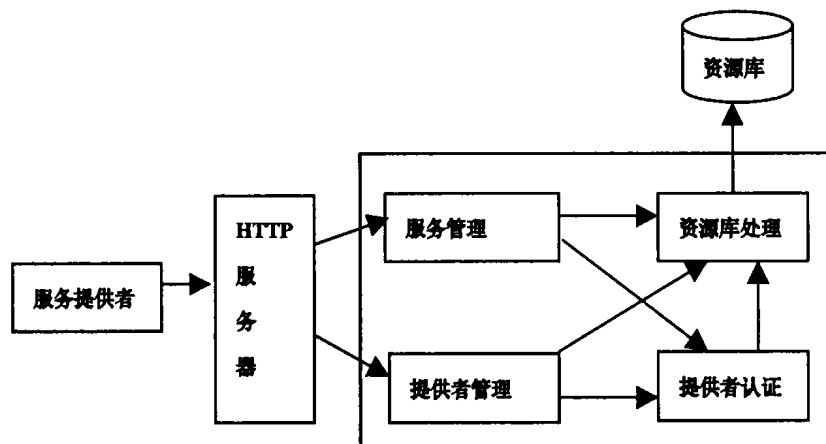


图4 注册管理器的结构

3.2 查询服务器

查询服务器^[6]为任务调度器提供了查询服务。

查询服务是使用 SOAP(Simple Object Access Protocol)来建立的。使用 SOAP 主要的好处是支持多

种语言,许多程序语言支持 SOAP 协议,例如 C++、Java 和 VB 等。查询服务器由两个模块组成:查询处

理模块和资源库处理模块。

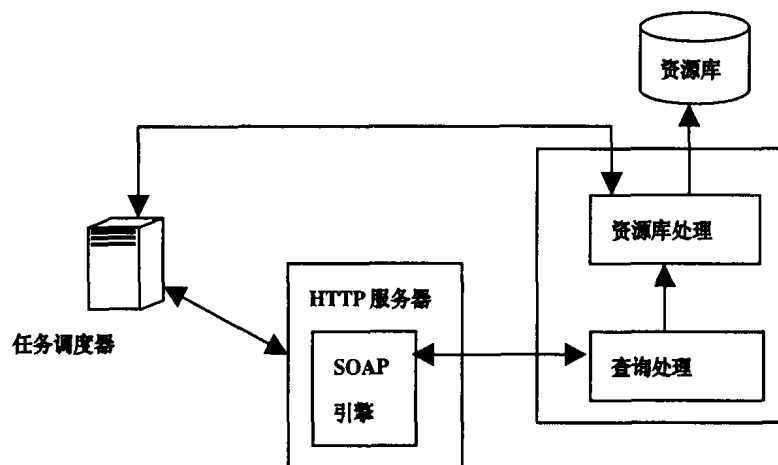


图5 任务调度的查询操作

查询过程用图5表示。任务调度器和查询服务器的通信使用 XML 格式。XML 格式的查询信息封装在 SOAP 信息中,用 HTTP 协议传输。SOAP 引擎获得查询信息并传送给查询服务器查询处理模块。查询处理模块处理查询信息并根据信息的内容分类和向资源库查询,把查询结果返回给任务调度器。

4 任务调度器的结构

任务调度器负责获取资源状态,用调度算法分配任务。

4.1 任务调度器组件

任务调度器的主要组件包括:授权认证模块、任

务的输入控制、资源查询、调度分配算法、任务的分配等。图6说明任务调度器的结构。

授权认证模块一对输入的工作任务的用户进行身份验证,确认是否有访问权限。

任务输入控制一通资源的查询模块获得可用节点状态信息,根据状态信息将输入的任务进行分解。

资源查询模块一向登记服务器查询获得可用资源的状态。

任务分配控制—根据网格节点可提供服务的性质和截止时间,用调度分配算法将工作任务分配给各个节点。

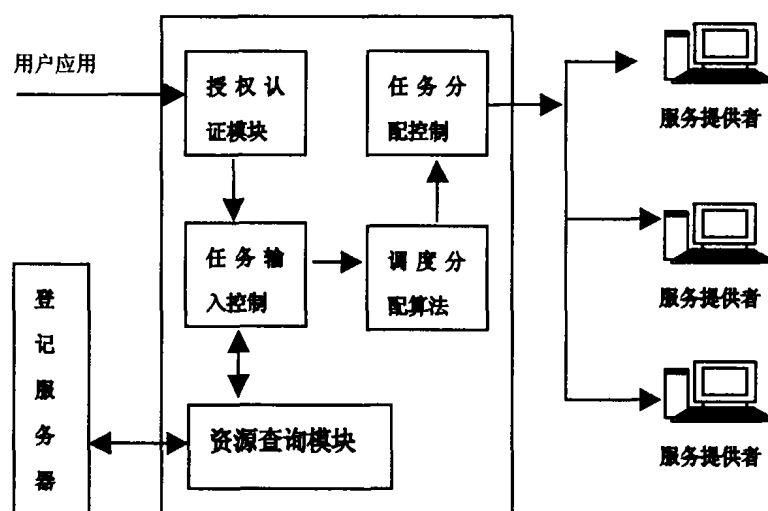


图6 任务调度器结构

4.2 任务调度的过程

任务调度器接受用户的工作任务,通过身份认证后,送到任务输入模块。任务输入模块向登记服务器查询各节点所能供服务的性质,包括,服务名、服务类型、服务提供的截止时间、节点主机名等等,然后运行调度分配算法,调度分配算法根据工作任务

的性质和节点提供服务的属性来对用户的任务分解。例如,A 节点能提供较强的计算能力,B 节点能提供较大存储,任务分配控制模块把需要较快运算能力的任务分配给 A 节点处理,需要大的存储的任务分配给 B 节点处理,由服务提供者进行处理,并

(下转第101页)

XML 文档集合对应的数据图为 G , 索引图为 IG , 要删除的文档对应的数据图为 H , 索引图为 IH , 更新后的数据图为 G' , 索引图为 IG' 。对 IH 中的每一个索引节点 A , $\text{ext}(A)$ 中的每一个数据节点, 在索引图 IG 中仍然属于同一个等价类, 也就是说在 IG 中存在某个节点 B , 使得: (1) $\text{ext}(A)$ 真包含于 $\text{ext}(B)$, 即在 G 中存在某个不属于 H 的节点与 $\text{ext}(A)$ 中的节点属于同一个等价类, 或者 (2) $\text{ext}(A) = \text{ext}(B)$, 即 $\text{ext}(B)$ 中的节点都是 H 中的节点。

当从一个 XML 文档集合上删除一个文档时, 我们从上面分析中提到的两种情况来说明索引的维护操作: 对 (1) $\text{ext}(A)$ 真包含于 $\text{ext}(B)$, 则 $\text{ext}(B) = \text{ext}(B) - \text{ext}(A)$; (2) $\text{ext}(A) = \text{ext}(B)$, 则可以删除节点 B 及其所有的子孙节点, 因为其子孙节点的 extent 也只由 H 中的数据节点组成 (如果两个节点不满足 bisimilarity 关系, 那么它们的子孙节点也不会满足这一关系)。

算法2

```

procedure del-file( $G, H$ )
begin
step1 计算数据图  $H$  的索引图, 令为  $IH$ , 为  $IH$  的根节点 (即文档的根元素对应的节点) 增加一个父节点, 作为新的根节点
step2  $IH$  的根节点入队列1
step3 令数据图  $G$  的索引图为  $IG$ ,  $IG$  的 root 节点入队列2
while (队列1不空)
step4 取队列1的队头节点  $A$ , 取队列2的队头节点  $B$ 
step5 求出  $A$  的所有子节点, 求出  $B$  的所有子节点
step6 对  $A$  的每个子节点  $Im$ , 在  $B$  的所有子节点中找出节点  $In$ , 使得  $(\text{ext}(Im) \text{ 真包含于 } \text{ext}(In))$ 
      则  $\text{ext}(In) := \text{ext}(In) - \text{ext}(Im)$ ;  $Im$  入队列1;
       $In$  入队列2;
      或者  $(\text{ext}(Im) = \text{ext}(In))$ 
      则在索引图  $IG$  中删除  $In$  及其所有子树;
end of while
end//end of procedure

```

该算法的主要流程是遍历索引图 IG 和 IH , 其时间复杂度为 $O(nIH * \log(nIG))$, 这 nIG 、 nIH 为索引图 IG 和 IH 的节点数。而如果重建索引的话,

时间复杂度将会是 $O(mG' \log(nG'))$, mG' 为删除文档后数据图 G' 的边数, nG' 为其节点数。算法2要比重建索引快得多。

下面来说明该算法的正确性。假设原 XML 文档集合的数据图为 G , 索引图为 IG , 要删除的文档的数据图为 H , 索引图为 IH , 更新后的数据图为 G' , 索引图为 IG' , 算法2得到的图为 I 。我们来说明 $I = IG'$ (这里, 图的相等是指同构)。对任意两个数据节点 $u, v \in G$, 而不属于 H , u, v 在 IG 中属于同一个等价类, 算法2并没有破坏这一条件, u, v 在 I 中也属于同一个等价类, 而 u, v 在 IG' 中当然也在同一个等价类中, 所以, I 是 IG' 的一个细分; 反之也成立, 因此有 $I = IG'$ 。这就证明了算法2得到的图就是我们所要求的更新后的索引图。

总结 在本文中, 我们讨论了在一个 XML 文档集合上增加多个文档和删除一个文档时的结构索引的维护操作, 提出了用于这两种情况下的更新索引的算法, 它们比我们已知的其他方法要更好。这些算法不只适用于远程监控系统, 对于其他的类似的应用也是适用的。其实, 这些算法也不只适用于以文档集合形式存放的 XML 数据, 对于其他形式的 XML 数据, 比如存放在关系数据库中的 XML 数据也是适用的。

参考文献

- Graves (美), 尹志军等译. XML 数据库设计. 机械工业出版社, 2002, 8
- Connolly (美), 宁洪等译. 数据库系统-设计、实现与管理. 电子工业出版社, 2004, 1
- Paige R, Tarjan R E. Three partition refinement algorithms. SIAM Journal on Computing, Dec. 1987
- Kaushik R, Bohannon P, Nauhgonand J F, Shenoy P. Update-forstructure indexes. In: Proc. of the 28th VLDB Conf. 2002

(上接第67页)

将结果返回给用户。达到合理的分配任务, 提高了资源的利用率。

结论 通过网格能创建虚拟组织或虚拟企业, 组织、商业机构和个人通过共享一部分或者所有的资源参与虚拟组织。本文提出了网格环境下基于虚拟组织的调度模型, 建立模型的目的是合理分配、调度资源来解决科学、工业、商业广泛存在大规模的计算问题, 此模型主要由资源登记服务器和任务调度服务器构成。登记服务器记录虚拟组织节点所提供服务的属性。任务调度服务器接受用户应用并分配调度工作任务。它根据资源提供者的属性和用户任务的性质通过调度分配算法来分解任务, 分配给各节点处理, 使虚拟组织各节点协同工作, 共同完成复杂计算任务。

参考文献

- Foster I. What is the Grid? A Three Point Checklist. July 20, 2002. <http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf>
- Foster I, Iamnitchi A. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In: 2nd International Workshop on Peer-to Peer Systems
- 都志辉, 陈渝, 刘鹏. 网格计算. 北京: 清华大学出版社, 2002
- Sullivan W T, Werthimer D, Bowyer S, Cobb J, Gedy D, Anderson D. A new major SETI project based on Project Serendip data and 100,000 personal computers. In: Proc. of the Fifth Intl. Conf. on Bioastronomy, Italy, 1997
- Chapin S, Karpovich J, Grimshaw A. The Legion Resource Management System. In: Proc. of the 5th Workshop on Job Scheduling Strategies for Parallel Processing, April 1999. (<http://legion.virginia.edu>)
- Yu Jia, Venugopal S, Buyya R. Grid Market Directory: A Web Services based Grid Service Publication Directory. Jan 2003. <http://www.gridbus.org/papers/gmd.pdf>
- Globus Web Site. <http://www.globus.org/>