TGrid 实现的关键算法

林伟伟[1] 齐德昱[1] 陈月红[2]

(华南理工大学计算机科学与工程学院 广州 510640)[1] (广东技术师范学院计科系 广州 510665)[2]

摘 要 TGrid是我们提出基于树形结构的面向高性能计算、面向主题的资源共享和新一代的需求建模的一种新的网格环境。本文在分析TGrid其特点的基础上,讨论其实现的关键问题,给出了树的建立、动态重构、任务分配等关键算法,重点描述了TGrid实现大规模并行处理的动态负载平衡算法,并进行模拟实验和分析,最后给出华南树型高性能计算网格的实例。

关键词 TGrid,虚拟资源,抽象,负载平衡

Key Algorithms for TGrid Implementation

LIN Wei-Wei[1] QI De-Yu[1] CHEN Yue-Hong[2]

(College of Computer Engineering and Science, South China Univ. of Tech., Guangzhou 510640)

(Department of computer, GuangDong Polytechnic Normal University, Guangzhou 510665)[2]

Abstract TGrid proposed by us is a new tree-based grid environment for high performance computing, subject-oriented resources sharing and the next generation requirement modeling. After analyzing the characteristics of TGrid and discussing the key problems of TGrid implementation, the algorithms for building tree and dynamic reconfiguring tree and task distributing are presented in this paper. Then, it focuses on the tree-based dynamic load-balancing algorithm for large-scale applications in TGrid and the experiment results of this algorithm are satisfying. Finally, an example of high performance computation grid, south china grid is showed.

Keywords TGrid, Virtual resource, Abstraction, Load balancing

1 引言

网格是当今的研究热点。网格目标就是把整个因特网整合成一台巨大的"虚拟超级计算机",实现计算资源、存储资源、数据资源、信息资源、知识资源、专家资源的全面共享和协同。要构成虚拟的超级计算机的网格计算环境,其中网格系统的体系结构是我们必须首先解决的问题。

目前,网格的体系结构主要有两大类:五层沙漏结构和开放网格服务体系结构(OGSA)[1~6]。此外,还有织女星网格的体系结构[7]、GridLab[8]、Rajkumar Buyya提出了经济网格及计算经济网格体系结构模型[9]等。网格体系结构研究是当前投入比较多的领域,但目前它仍然有很多不完善的地方:

1)没有考虑网格拓扑:地理上分布的各个资源节点或网格节点以什么样网络拓扑来连通才能更好地实现资源共享和协同计算等;

2)应用开发困难:大都是针对专业用户的,需要使用网格的编程语言进行复杂的编程工作,一般用户难以使用;

3)标准化程度低、抽象性不足:五层结构中,中间小,通用的性较强,但两头大,是面向具体问题的,对不同的网格计算问题,构成也不同,非标准化的成分太多;在OGSA中,具体应用的实现是服务的集成,服务的管理与互操作是标准化的,但具体的服务都是由用户按规范根据具体应用提供的,因而OGSA抽象出来的标准化成分太少;

4)管理问题:网格中关键节点的管理复杂,难以构造大型网格应用;

5)与网格最终目标有差距:无论是五层沙漏还是OGSA的网格体系结构,都距"虚拟的超级计算机"要求很远。

针对上述问题,在文章"一种新的网格环境模型——TGrid"中提出一种基于树型网络拓扑的树型网格环境TGrid,它以树结构来组织网格节点和集成各种资源,实现了自底向上、多级、面向需求的资源抽象和多种资源融合,并从网络拓扑上来解决大型复杂应用负载平衡问题和保证资源的快速定位,而且树型结构符合自然层次组织关系,容易实现网格系统的层次化管理。然而,要实现TGrid,我们需要解决诸如系统、体系架构、实现算法等等问题,因此,在本文中,我们首先概述TGrid,并分析了其特点,然后讨论其实现需要解决的问题,重点给出一些实现TGrid的关键算法和基于树的动态负载平衡算法以及模拟实验结果。

2 树型网格概述及特点

2.1 树型网格概述

如图1所示,整个网格以树结构拓扑来组织网格节点,网格节点可以动态加入和离开系统,所有网格节点在逻辑上构成一个棵树,它又由若干子树组成,每棵子树的根节点不仅是子树的管理节点,而且也是上层树的子节点,负责向上层传送该子树的相关信息,并由上层树的根节点管理。整个网格系统是一个基于树型的分层管理体系。系统在运行过程中,树的结构可以根据情况改变,即节点可以删除、增加,节点间关系可以动态改变。每个网格节点上的物理资源(比如计算机的CPU、内存、数据库等)被抽象成虚拟资源,并提供给用户或应用访问。虚拟资源即可以是虚拟CPU和主存的分布式JVM(TJVM),用户可以提交一个大型Java程序给网格,然后由网格系统将实现任务分解,并将子任务在TJVM子树中并行分布给各TJVM执行,最后收集结果返回给用户,从而

可以在树型网格中实现面向 Java 应用的高性能计算。而且，为了执行大型的 Java 程序，可以让多个具有 TJVM 资源的节点形成一棵 TJVM 虚拟资源树，以树型网格上实现分布并行计算。网格节点上的虚拟资源也可以是实现数据库级资源集成和共享的多数据库中间件(TDOD)或者是实现其他软件和数据资源共享的 Globus 网格服务(Gservice)。

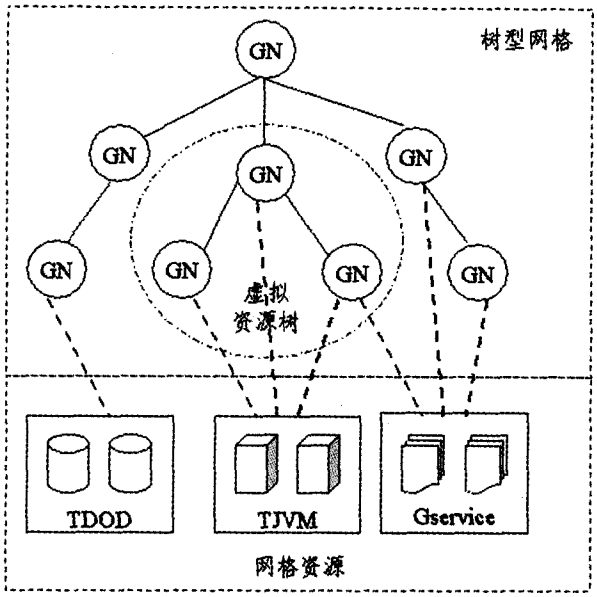

图 1　树型网格结构示意图

2.2　树型网格的特点

树型结构有利于减轻中心节点的负载，实现大规模应用的负载平衡，提高资源查找效率，实现系统的层次化管理。同时，该树型网格为网格用户提供不同程度资源抽象，以实现用户对网格资源各种需求和透明使用。例如，用户可以通过网格提交一个大型 Java 应用来完成一个复杂科学计算任务，也可以通过网格获得某项服务，如查询车票服务。总结它的主要特点如下：

1)树型结构有利于减轻中心节点负载，实现任务负载平衡，提高资源查找效率；

2)实现层次化管理，降低管理复杂度；

3)可以实现自底向上、多级、面向需求的资源抽象和多种资源融合；

4)新加入节点可以根据其网络情况来寻找父节点，提高各节点之间的网络通信性能；

5)大型 Java 应用的分布透明执行，面向高性能；

6)该网格体系结构的每个节点上资源分类专业化。TJVM 虚拟计算机 CPU 和内存资源，GLOBUS 网格服务虚拟各种软件和数据资源，TDOD 虚拟集成各种数据库资源；

7)面向主题的资源共享和多样化资源表现形式。事实上多样性是事物的存在方式，就如同软件的多样性和计算机应用技术的多样性一样。

3　树型网格实现的关键问题

要实现树型网格，需要解决以下关键问题：

首先需要解决网格模型的问题。各网格节点如何组织，采用什么样的网络拓扑，是树型还是图模型，或者星型模型？在树型网格我们将采用动态可动态重构树，那么必须实现这棵树的建立、动态可重构的相关算法，动态可重构树是系统在运行过程中，树的结构可以根据情况改变，即节点可以删除、增加，节点间关系可以改变。引入动态可重构好处是容易实现负载均衡和故障恢复。

然后是网格资源管理与分配方面。现有的资源管理模型主要有层次模型、抽象所有者模型和经济/市场模型[10,11]。它是网格的核心功能。针对树型网格的特点，提出适合树型资源管理的模型，在资源管理将引入关键技术：1)采用分层等级结构，通过逐级资源管理处理任务与资源的调度和管理问题，以实现任务负载均衡和分级管理，减轻全局资源管理器负载；2)引入移动 Agent 收集网格资源动态信息，适应网格资源的动态变化。

网格资源信息服务方面。目前资源信息服务按支撑技术的不同主要分为三种，分别是基于 LDAP 目录的信息服务[12]、基于 OGSA 开放架构的信息服务以及基于 RDBMS 关系型数据库管理系统的信息服务[13]。在我们树型网格下，对于大量动态资源信息的管理可以借助移动 Agent、P2P 等技术。

网格资源发现机制。现有资源发现机制主要有集中式、分布式、分层式和基于 P2P 技术的资源发现机制等，在我们树型网格下，适合采用什么样资源发现机制？是否可以采用的是集中式与分布式相结合的分层机制，并结合移动代理在资源发现方法的优势。

网格服务质量方面。目前，在网格中间件与其资源管理系统两者能力存在差距，网格中间件提供一些 QoS 支持，但其资源管理系统只提供有限的功能。然而，为了确保整个系统服务质量，必须在所有层次上支持 QoS，这就要求不仅需要把现有 QoS 功能集成到局部资源管理系统，而且需要更强资源预留机制来确保 QoS。特别是，要实现用户可接受的各种 QOS，必须在用户和资源管理器有灵活的协商机制，我们可以采用基于服务级别协定(Service Level Agreement)来实现。

树型网格安全。研究树型网格的安全体系结构，并实现其身份认证及访问控制的策略及其相关协议。

4　树型网格的关键算法

4.1　树的建立与动态重构算法

1)树的建立

树的建立，就是针对加入网格的物理资源构造并建立相应的网格节点的通信虚拟树。树主干由网格管理员来完成，当有新的节点申请加入时，即新的资源加入，首先选择父亲节点，然后更新父亲节点信息表，同时同步更新所有相关结点的资源信息，该结点即加入了系统。新加入节点有两种方式：一种是由网格管理员加入，网格管理员可以为新加入的节点指定父亲节点；另一种是由系统自动加入，系统根据资源的类型和节点的网络位置选择一个或多个父亲节点加入。新增资源的自动加入是网格环境应该具备的功能，它充分反映了网格环境动态性的特点。当主干确定以后，网格环境中的用户可以自由地将它的计算资源添加到网格环境中进行共享，而不需要管理员的干预，资源的自动加入遵循网络邻近原则[14]。

2)树的动态重构算法

系统在运行过程中，树的结构可以根据情况改变，即节点可以删除、增加和转移，节点间关系可以动态改变。

节点增加算法与树建立的系统新节点自动加入算法相同。

节点删除算法：由于涉及到要重新分配在上面正在执行

的任务，问题变得比较复杂。如果删除的是叶子节点，则告诉其父亲节点将其上的任务重新分配；如果删除不是叶子节点，则必须将与之相关联的子节点重新增加到其父亲节点所在树，并且告诉其父亲节点重新分配删除节点对应子树上的任务。

节点转移算法，节点转移算法是为了更好实现负载平衡而提出来的。当某个节点完成任务，并且其父亲节点没有任务，则可以将该节点先删除，然后按照节点增加算法把该节点增加到负载重的子树中，这样就不需要重新进行复杂的任务分配。

4.2 资源查找算法

资源的查找，主要是为任务分配资源。根据用户需要资源的类型（主要在树型网格抽象部分给出的三种虚拟资源）和对资源要求在网格节点树上进行查找，从某个网格节点出发，在网格节点树上查找到满足查找条件的资源。

资源查找算法描述为：(1)首先判断当前节点是否存在满足要求的资源，有满足要求则结束查找；(2)否则，将查找需求发送到父亲节点。由于父亲节点存储了整个子树的资源情况，所以如果存在满足要求的解，则必然在该子树中，即找到需要资源；(3)如果父亲节点也没有满足要求的解，则继续向上一级父亲节点发送查找请求；(4)直到查找过程进行到根结点，则查找结束。

资源查找算法时间复杂度分析：在最坏的情况下，查找步数为树的深度 $\log(N)$，因此其时间复杂度为 $O(\log N)$，平均情况下的时间复杂度也为 $O(\log(N))$。

4.3 并行处理的动态负载平衡算法

为了充分利用分布在网络的多个网格节点资源，高效的任务分配算法是非常重要的。负载平衡常用作任务分配的方法。负载平衡算法能分配计算任务到各个网格节点资源以实现资源最大利用率和网格系统最大吞吐量。现有的网格计算大都是以主从（Master/Slave）模式来实现大规模并行计算的，在这种计算模式中，这种模式中，一个节点作为 Master，它负责指挥与之相关联的一个或者多个 Slave。由于 Master 主机必须负责任务分配与协调、收集和合成任务执行结果、任务负载平衡等，这样容易造成 Master 主机负载过重，特别是随着 Slave 数量的增加，Master 主机将成为整个系统的瓶颈。这里的动态负载平衡算法是基于大规模的粗粒度任务的分配算法，所以，下面首先给出分配算法。

大规模粗粒度任务的分配算法

1)树的根节点在收到一个大任务后，首先选取一些子树，并分配任务给这些子树的根节点；2)子树的根节点收到任务，将其任务分配给儿子节点；3)如果某个树节点没有儿子，则执行任务，并返回结果给父亲。

动态负载平衡算法

在树型网格中，由于整个网格系统由树来组织个网格计算节点，整个网格就是一个棵树，而这棵树又由若干子树组成。因此，可以采用一种集中与分布式方法结合的分级方法[15,16]来实现任务的负载平衡，即通过子树来实现多层次管理。在我们提出的层次方法中，首先按照上面的任务分配算法执行，当某个叶子节点空闲时，动态重新分配该叶子节点到负载重的子树，以便它更快完成任务。在这种方法中，不需要重新分配任务，也就不需要移动任务。由于移动任务的开销相对较大，而重新分配树的叶子节点只需要更新对应父亲节点的儿子信息表，所以这种动态分配计算节点的方法比重新分配任务的方法能更好地实现任务的负载平衡。

在计算过程中为了收集中间结果，必须设置各个节点的同步，我们定义一个判断同步点函数 syn()，当其取值为真时表示需要同步。动态负载平衡算法的形式描述：

```
Procedure root_task_assign()//根节点任务分配过程
  While (not syn())
    Select_task(Queue)//从任务队列中选择一分配任务
    Assign(idle(Snode))//分配任务给空闲儿子节点
    Put_task(Queue)//将新任务放到队列中
  End While
  If (syn()and taskfinish())//到同步点且任务完成
    Send(STOP,Snode)//发送停止消息给儿子节点
  Else//发送空闲消息给请求任务的儿子节点
    Send(IDLE,Request(Snode))
  End If
End
Procedure Snode_task_assign()//分支节点任务分配与执行过程
  While (Receive_task())//当从父亲节点收到任务
    Put_task(Queue)//将新任务放到队列中
    Execute_task(Queue)//执行一个任务
    Assign(idle(Snode))//分配任务给空闲儿子节点
    Gather_result()//收集子节点的执行结果
    Send_result()//发送结果给父亲节点
  End While
  If (Receive()=IDLE)//收到空闲消息
    Reassign_node()//重新分配儿子节点
    Update_node()//更新儿子节点
  Else IF(Receive()=STOP)//收到停止消息
    Send(STOP,node)//发送停止消息给儿子节点
  End If
End
Procedure node_task_assign()//叶子节点执行任务过程
  While (Receive_task())////当从父亲节点收到任务
    Execute_task(Queue)//执行一个任务
    Send_result()//发送结果给父亲节点
  If (Is_Snode())//被分配给新的父亲节点
    Update(Snode())//更新父亲节点
  End If
  End While
End
```

4.4 动态负载平衡算法的模拟分析

模拟实验系统的建立。模拟实验系统是在 Windows 和 Linux 平台下采用 Java 语言实现的。实验共有 10 台微机，它们的配置有一定差别（可以模拟计算节点的计算能力的差异）。首先选择一台机器作为树的根节点，它负责生成多个可并行的计算任务（注意，这些任务必须具有局部可并行性和整体上具有串行性的特点），也称为全局管理节点。将另外 9 台机器连接上根节点，作为根节点的儿子，负责任务分配和收集儿子节点执行结果，也称为局部管理节点。然后在 9 台机器上生成多个 Java 多线程程序，一个多线程程序模拟一个计算节点，即树的叶子节点，它完成由分支节点提交的计算任务。实验比较三种任务分配方案下并行处理效果。

三种任务分配方案。1)无负载平衡方法：每当根节点生成一批并行任务后，平均分配任务到各个节点（模拟的线程程序），等到所有线程执行完后，根节点再分配下一批任务。2)静态负载平衡方法：每当根节点生成一批并行任务后，按照节点的计算能力分配一定数量任务到各分支节点，即静态负载平衡分配；3)基于树的动态负载平衡方法：先将任务按照各分支树的计算能力分配一批任务到各分支节点，然后分支节点将任务平均分配到各线程。并且在任务执行过程中如果某些线程程序空闲可以立即加入到其它负载重的分支节点，作为它的儿子参加计算。

实验结果与分析。模拟实验结果如图 2 所示，由图可以得出下面结论。在系统执行的任务数量不大的情况，静态负载平衡方法和基于树的动态分配方法的执行时间相对无负载平衡方法较短，这是因为任务数量少时，系统基本上不需要进行动态负载平衡。在系统执行的大量任务时，基于树的动态分配方法的执行时间明显优于静态负载平衡方法和无负载平

衡方法，这是因为任务数量大时，在基于树的动态负载平衡方法中的各个分支节点间可以进行计算资源的协调，可以加速每一批任务执行的数度，进而提高整体的效率。

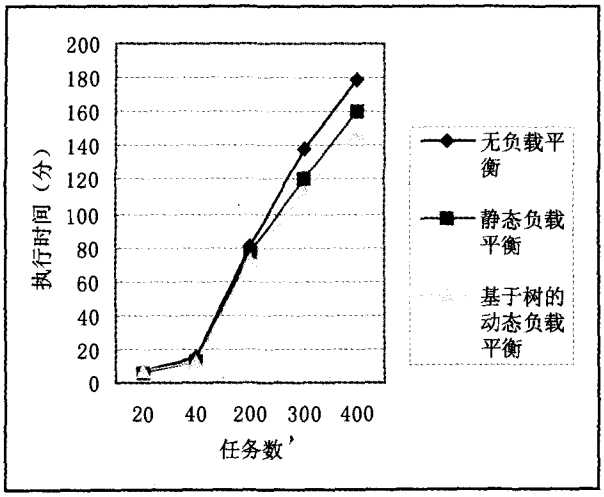

图 2　负载平衡模拟实验结果

5　树型网格实例

我国一个特色是行政体制是从中央到省市再到县镇的一个树型结构，相应地，我国的网络建设和信息化建设也形成了这种树型结构。因此，组成一个层次分明、树型结构的网格系统将更加有利于共享网上所有硬件和软件等资源。而且，在具有中国特色的树型网络结构下非常有利于建立中国特色的树型网格。

以建立 CERNET 华南高性能计算树型网格为例，网格系统管理员指定 CERNET 华南网络中心为整个网格系统的根节点，并在其上安装树型 JVM 网格软件，然后把各个高校作为根节点的子节点资源。各高校、各研究机构的管理员可以指定本单位各系各所的资源作为下一级子节点，或者下载并安装树型网格软件自动加入树型网格作为其中一个节点。图 3 描述了设想中的 CERNET 华南树型高性能计算网格的主干结构。

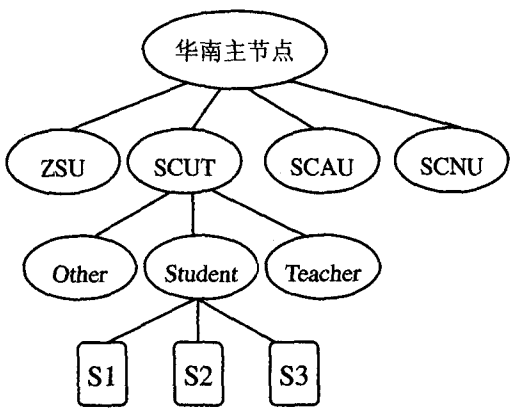

图 3　CERNET 华南树型高性能计算网格环境

该树型网格是一种面向 Java 应用的分布并行计算的树型网格系统，它使多线程 Java 应用和多 Java 作业均可在此网格系统下实现高效、透明的分布并行运行，并且能实现大规模任务的负载均衡。该树型网格的每个网格节点上计算机 CPU 和内存资源的虚拟部件 JVM 是在标准的 JVM 上的扩展。多线程 Java 程序中的各线程、多 Java 作业中的各独立 Java 程序都可自动地被分配到网格中的适当节点执行，实现负载均衡、并行执行、任务协作。

结论　树型网格 TGrid 采用“类集”的思想集成资源，构造网格，使网格资源组合为各种不同的功能类，共享各种可用资源，从而实现面向主题的资源共享和高性能计算。本文是继上一篇文章“一种新的网格环境模型——TGrid”之后的系列文章之一，重点分析讨论树型网格 TGrid 的实现关键问题。然后给出树型网格的几个主要算法：树的建立与动态重构算法、并行处理的动态负载平衡算法和资源查找算法，最后，描述了建立华南树型高性能计算网格的实例。当前我们正在对树型网格的相关实现算法和协议进行模拟实验和分析，并在开发树型网格的原型系统——华南树型高性能计算网格环境。

参 考 文 献

1　Foster I, Kesselman C. (Eds.), The Grid 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2004

2　Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid, International Journal on Supercomputing Applications, 2001, 15(3):200～222

3　Foster I, Kesselman C, Nick J M, Tueche S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 2002

4　Introduction to Grid Computing with Globus[EB/OL], http://www.ibm.com/redbooks, 2003

5　高全泉. 网格：面向虚拟组织的资源共享技术. 计算机科学，2003，30(1):1～5

6　The Globus Project [EB/OL]. http://www.globus.org/, 2004

7　徐志伟，李伟. 织女星网格的体系结构研究. 计算机研究与发展，2002，39(8)

8　Allen G, Davis G K, Dolkas K N, Doulamis N D, et al. Enabling applications on the grid: A Gridlab overview, International Journal of High Performance Computing Applications, 2003, 17(4): 449～466

9　Buyya R, Abramson D, Giddy J. A Case for Economy Grid Architecture for Service-Oriented Grid Computing. In: Proceedings of the International Parallel and Distributed Processing Symposium: 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), April 23, 2001, San Francisco, California, USA, IEEE CS Press, USA, 2001

10　Buyya R, Chapin S, DiNueei D. Architecture Models for Resource Management in the Grid [C]. The First IEEE/ACM International Workshop on Grid Computing (GRID 2000), Springer Vedag LNCS Series, Germany, Bangalore, India: 162～182

11　Buyya R, Abramson D, Giddy D, et al. Economic Models for Resource Management and Scheduling in Grid Computing [J]. Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, 2002.

12　Czajkowski K, Fitzgerald S, Foster I, Kesselman C. Grid Information Services for Distributed Resource Sharing. In: Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), 2001

13　DataGrid. DataGrid Information and Monitoring Services Architecture: Design, Requirements and Evaluation Criteria: [Technical Report]. 2002

14　杨广文，武永卫，朱晶. 一种全局统一的层次化网格资源模型. 2003，40(12)

15　Foster I, Kesselman C, Lee C. A Distributed Resource Management Architecture that Supports Advance Reservation and Co-Allocation. In: Proceedings of the International-Workshop on QoS, 1999. 27～36

16　Leff A, Rayfield J T, Dias D M. Service-Level Agreements and Commercial Grids. IEEE Internet Computing, 2003, 7(4): 44～50