

网格计算系统研究及发展方向

Grid Computing Research and Future Direction

王育峰 杨寿保

(中国科学技术大学计算机科学技术系 合肥230026)

Abstract This article details grid computing characteristic, motivation, architecture, key techniques, system software and application, analyzes the present research, and gives the future research direction.

Keywords Grid computing system, Grid technology, Metacomputing

1. 引言

随着高性能计算应用需求的迅猛发展,解决一些超大规模应用问题所需要的计算能力,已不可能在单一的计算机上获得。因此,将地理上分布、系统异构的多种计算资源通过高速网络连接起来,构建成网络虚拟超级计算机,以此来共同解决大型应用问题,即高性能的元计算(Metacomputing)技术应运而生^[1]。

元计算系统被定义为通过高速网络连接、由地域分布的动态计算资源构成的网络虚拟超级计算机系统^[2]。网格计算(Grid Computing)发展了最初元计算的概念,它通过将地理上分布、异构的各种高性能计算机、数据服务器、大型检索存储系统和可视化、虚拟现实系统等,通过高速互连网络连接并集成起来,形成对用户相对透明的虚拟的高性能计算环境,这个计算环境是一个广域范围内的无缝集成和协同计算环境。

本文将详细讨论网格计算系统的特征、目标、体系结构、关键技术、系统软件及应用,分析网格计算系统的研究现状,对网格计算未来的研究趋势提出自己的观点,最后是小结。

2. 网格计算系统

2.1 网格计算系统的特征

网格计算系统与分布式系统和并行系统相比有很多相同的特征,但与二者又有着非常重要的区别。与分布式系统类似,位于多个管理域下的超级计算机通过不可靠的网络进行连接,并且需要对广域分布的动态资源进行集成,但是网格计算系统对高性能的要求使其编程模型及接口与分布式系统有极大的差别。网格计算系统作为并行系统还需要进行超级计算机之间的通信调度以满足应用对性能的要求,然而由于网格计算系统的异构性以及动态性,现有的并行计算技术不能够很好地适应这种需求。

一般而言网格计算系统具有以下几个特征^[3]:

(1)扩展性:网格计算系统初期的规模较小,随着超级计算机系统的不断加入,系统的规模随之扩大。

(2)系统多层次的异构性:构成网格计算系统的超级计算机有多种类型,不同类型的超级计算机在体系结构、操作系统及应用软件等多个层次上具有不同的结构。

(3)结构的不可预测性:与一般的局域网系统和单机的结构不同,网格计算系统由于其地域分布和系统的复杂使其整体结构经常发生变化。

(4)动态和不可预测的系统行为:在传统的高性能计算系统中,计算资源是独占的,因此系统的行为是可以预测的,而在网格计算系统中,资源的共享会造成系统行为和系统性能经常变化。

(5)多级管理域:由于构成网格计算系统的超级计算资源通常属于不同的机构或组织并且使用不同的安全机制,因此需要各个机构或组织共同参与解决多级管理域的问题。

对于网格计算系统来说,最根本的问题是实时获得系统的结构和状态信息,通过这些信息对网格应用进行配置,并能实时获得计算资源的状态信息。

2.2 网格计算系统的目标

网格计算系统的目标是使用户能够共享其中的计算资源并以合作的方式进行计算,为此有两个层次的工作要做。其一是网格计算前端,主要解决最终用户通过统一的界面来使用广域网上各类计算资源的问题;其二是网格计算内核,主要解决计算任务在广域网中各种超级计算机上协作完成的问题,提供一个完整的程序开发和运行环境。当用户提出计算请求时,计算问题的执行代码在系统内部的计算资源上进行合理的调度和管理,最后得出运行结果并通过网格计算前端反馈给最终用户^[4]。

为实现上述目标,必须重点解决以下三个问题:(1)异构性:由于网格由分布在广域网上不同管理域的各种计算资源组成,怎样实现异构机器间的合作和转换是首要问题;(2)扩展性:要在网格资源规模不断扩大、应用不断增长的情况下,不降低性能;(3)动态自适应性:在网格计算中,某一资源出现故障或失败的可能性较高,资源管理必须能动态监视和管理网格资源,从可利用的资源中选取最佳资源服务。

2.3 网格计算系统的体系结构

网格的体系结构从下到上定义为四个层次:结点及互联层、工具层、中间件层和应用层^[5]。网格的体系结构如图1所示。

(1)结点及互联层:由各类资源及连接这些资源的高速网络组成,这些资源包括异构的各种高性能计算机、数据服务器、大型检索存储系统和可视化、虚拟现实系统等。

(2)工具层:提供给用户的工具,以使用户能够更好地利用网格资源。

(3)中间件层:网格计算的核心,负责提供远程进程管理、资源分配、存储访问、登录和认证、安全性和服务质量等。

(4)应用层:提供系统能接受的编程语言,并配置其它一

王育峰 硕士研究生,研究方向为计算机网络、网格计算、机群系统,杨寿保 教授,博导,研究方向为计算机网络及应用、信息安全与密码学。

些支持工程应用、数据库访问的软件,还可提供 Web 服务接口,使用户可以使用 Web 方式提交其作业并取得计算结果。

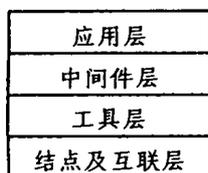


图1 网络计算系统体系结构图

2.4 网络计算系统的关键技术

网络计算系统的关键技术包括以下几个方面:

(1)网络技术 研究提高网络的性能、可靠性和容错性,为桌面机提供一个高带宽的通信系统,高速通信协议和通信服务质量保证技术对网络计算系统的研究都非常重要。

(2)高效可扩展的分布式并行计算系统 为了使用户能够透明地使用网络资源,需要建立一个高效的、可扩展的分布式并行计算环境,并以此为基础提供各种高级的服务。

(3)面向用户的程序设计模型 为了能够充分发挥网络计算系统的特点,需要为网络系统设计有效的程序设计模型来充分利用资源。面向对象技术将会是主要的实现技术之一,基于 CORBA、Java 或者 Web 等技术的分布式对象技术适合于建造复杂的计算对象和移动计算对象。

(4)可视化技术 在网格计算中的可视化技术使用图形、图像等手段来描述网格系统中资源和任务的行为、状态和结构,为网格计算系统的资源管理、作业调度和数据传输提供帮助。

(5)大型科学应用计算研究 网络计算系统的目的是充分利用网络上的各种资源来支持大型的并行分布式计算,因此,对应用的研究也应同时展开,由应用驱动来提出对网络计算系统的功能和技术要求,并验证其技术途径和技术实现的有效性。

(6)高效的程序编译模型和执行引擎 传统情况下,应用程序在执行前需要编译成面向某平台的可执行代码,这些本地码的平台相关性决定了它们不能跨平台执行,这使得程序在异构环境下的执行和资源的使用受到了一定的限制。解决这个问题的方法是使用平台无关的中间代码,由于中间代码通常是解释执行的,因而执行的效率比较低。为了解决这些问题,需要对程序的编译模型和执行引擎进行研究,解决平台无关性和执行效率之间的矛盾。

2.5 网络计算系统的系统软件

建立大规模的网络系统的关键技术是网格系统软件,特别是网格操作系统,其核心技术包括以下几个方面:网格资源的管理、信息优化使用技术、网格中作业调度技术、网络安全技术等^[6]。

网格系统软件必须发挥各网格节点的高性能计算系统和大容量数据信息处理系统的性能和功能,使之成为网络上的可共享使用的国家高性能基础设施,提供多用户远程使用网格节点计算和数据资源,为各行业用户提供具有统一管理、易于远程使用、安全的高性能计算和大容量数据处理的环境。网格系统软件要达到的目标有如下几点:

(1)单一的用户资源管理:全网格的帐号管理和记账系统。

(2)单一的资源目录管理:统一用户界面,一致的命令格

式,全网格和结点的资源状态动态监控和查询,环境资源的动态更新、查询。

(3)单一的作业管理:作业由统一的作业管理系统进行管理,并且作业具有优先级别。提供批处理和交互式作业管理机制,对多用户作业进行有效的调度。可在全网格范围内指定机器类型和结点类型。

(4)系统学习环境:全网格及行业网格环境的人员学习培训和系统联机帮助。

(5)远程使用环境:支持远程用户的并行程序开发,以 Telnet、FTP、TTY、CDE、Xterm 等仿真环境为主,远程程序的执行不因 Telnet 等的异常而中断。支持生产性程序的远程用户使用(应用程序已经开发完毕,用户提供数据便可以在网络结点上进行计算),以 Web 等各种方式提供支持。

2.6 网络计算系统的应用

基于网络计算系统的应用主要可以分为四类:

(1)桌面超级计算:这些与远程超级计算机和数据库相耦合的应用具有高端图形处理能力,这种耦合使用户在获得超级计算能力的同时,与计算资源、应用开发人员以及其他用户保持远程距离。

(2)精密仪器:这些应用将用户和远程超级计算机系统上的望远镜、显微镜以及卫星接收装置相连以获得准实时的数据处理。

(3)协同环境:第三方应用将多个虚拟环境组织在一起,不同地点的用户可以和其他用户和超级计算机模拟系统进行交互。

(4)分布式超级计算:这些应用将多个超级计算机组织在一起,解决一些单机难以处理的问题,或者将问题分解到多台计算机上并行处理。

3. 网络计算系统的研究现状

网络计算环境可以连接广域范围内不同标准的异构“孤岛”,形成庞大的全球性计算体系,是 Internet 发展的高级形式,因此,受到世界各国和组织的高度重视,已经开展了许多论坛(<http://www.gridforum.org>)、实验环境和研究项目,它们大多数都是在网上共享的,极大地方便了人们研究和利用有关资源。较有代表性的网络计算项目包括:实验床、Globus 项目、Legion 项目、Globe 项目、NetSolve 项目、Javalin 项目等。这些项目可简单地分成有代表性的两类:

(1)Globus 项目。提供基础的软件,集成分散的异构资源,形成一个单一的计算环境。其核心是 Globus 网络计算工具包,这是一个构筑网络计算环境的中间件,提供基本的资源定位、管理、通信和安全等服务。该计算工具包是模块化的,允许用户按自己的需要定制环境。

(2)基于 Java 的网络计算。Java 语言和相关技术成功地解决了困扰网络计算的几个关键问题,如异构性和安全性,另一个重要的优势是 Java 程序的最小执行环境可以在 Web 浏览器中执行而不需要另外安装软件,理论上全球任意一台装有 Web 浏览器的机器都可以进行全球计算。尽管 Java 平台还存在效率低等问题,但它无疑将大大影响网络计算模式的发展,使实现全球分布式计算已不再是一个梦想。

当前的研究普遍停留在网络计算前端上,并有不少可投入使用的系统。这些系统的普遍模型是:以某一个或几个超级计算机作为计算资源,在这些超级计算机上有一些预登记的计算应用程序;在整个广域网上的用户和这些超级计算机之

间提供一个中间件,使得使用不同硬件环境、操作系统环境的各个用户能够透明而安全地提交自己的计算请求;由中间件将这些计算请求转换成超级计算机上预登记应用程序所要求的形式,交给超级计算机进行计算。

在上述网格计算模型中,基本上没有针对网格计算内核做什么工作,随着 Jini^[7]技术的出现,前端技术的研究变得更为简单,而内核技术的研究则相对落后。高性能计算由局域网扩展到广域网时缺乏适合广域网计算并行程序编程模型和统一的、与编程模型相协调的计算资源管理模型。

针对网格计算系统的特点及目前研究的实际情况,一些研究单位将网络计算的前端与内核统一看待,研究网格计算系统资源的异构与动态特性、计算资源与并行任务充分匹配的资源管理模型,提出并研究一种不依赖用户身份的网格计算系统,并且给出安全、可靠、低开销的可定制资源保护策略,使网格计算系统资源管理与资源保护更加实用、科学。

4. 网格计算系统的研究趋势

分析网格计算系统的特点及目前研究的实际情况,网络技术未来的研究可能集中在如下几个方面:

(1) 动态自适应研究 研究网格计算系统的动态自适应性,使得网格计算系统能够自动适应环境的变化。网格计算系统中某一资源出现故障或失败的可能性较高,系统的资源会不断扩大、应用会不断增长,系统的整体结构和整体性能会不断地发生变化,并且随时有不可预测的系统行为发生,这就要求资源管理程序能够动态监视和管理网格资源,从目前可利用的资源中选取最佳资源服务,尽量减小由于这种故障或失败、整体结构和整体性能发生变化或不可预测的系统行为等问题对网格整体性能的影响。

(2) 安全管理研究 研究网格计算系统的安全管理机制,

确保网格计算系统管理和使用的安全性。建立全网络的帐号管理和记账系统,使得任何用户能够从任何连在网络的计算机上,安全登录并有效使用网格资源;确定适用于网格计算系统的信息加密机制和信息传输机制;确定网格计算系统的管理层次体系,将管理域按照区域层次划分,并且决定管理信息流的流向;能够为不同级别的系统管理员提供强有力的工具或界面监视系统资源和系统的运行情况。

(3) 应用计算研究 研究可使用网格计算系统进行计算的新应用,充分利用网络上的各种资源来支持大型的并行分布式计算,由应用驱动来提出对网格计算系统的功能和技术要求,并验证其技术途径和技术实现的有效性。

(4) 高效的程序编译模型和执行引擎研究 研究解决平台无关的中间代码执行效率低下的问题,对程序的编译模型和执行引擎进行研究,解决平台无关性和执行效率之间的矛盾,最终能够提供一种或几种解决方案,使得程序能够顺利地异构环境下执行和使用系统资源。

参考文献

- 1 Smarr J, Catlett C. Metacomputing. Communication of the ACM, 1992, 35: 44~52
- 2 Foster I, Kesselman C. Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications, 1997, 11(2): 115~128
- 3 <http://www.origincom.com.cn/article/20010627-1.htm>
- 4 <http://www.ccw.com.cn/html/produ/topic/01-6-8-13.asp>
- 5 <http://www.dl.ac.uk/TCSC/UKHEC/GridWorkshop/Tanaka/ninf/img2.htm>
- 6 <http://www.grid.org.cn/html/overview/info.htm>
- 7 <http://www.jini.org/whatisjini.html>

(上接第146页)

语义的修改,这两种修改操作都可改变抽象逻辑结构图 t_i 的状态,但不会改变其结构。若逻辑结点的修改操作引起概念结点类型和概念结点语义的变化,则会导致抽象概念结构图 t_c 的状态变化。由于逻辑结点的修改不会引起抽象逻辑结构图 t_i 的结构改变,因而逻辑结点的修改不会导致抽象概念的结构变化。

假设 a' 是抽象逻辑结构图 t_i 上将被修改的逻辑结点,下面我们依据 a' 是导出结点还是扩展结点而分成两种情况来讨论:

(1) 如果 a' 是逻辑导出结点,即 $a' \in A_{id}$,则在抽象概念结构图 t_c 的概念结点集 A_c 中存在一个概念结点 a ,有 $a' = f_{id}(a)$, $a = f_{id}^{-1}(a')$ 。修改逻辑导出结点 a' 的逻辑结点类型,必然引起抽象逻辑结构图 t_i 的状态变化而导致修改后的抽象逻辑结构图 t_i 与抽象概念结构图 t_c 中结点 a 的概念结点类型与 a' 的逻辑结点类型不一致,必须对导出结点 a' 的概念源结点 a 的概念结点类型进行更新。这时只需使 $a = f_{id}^{-1}(a')$ 满足逻辑结点到概念结点的逆向映射规则 R_{id}^{-1} ,以确定 a 结点的逻辑结点类型 $a[Tcn]$ 。若对结点 a' 的修改有效而保证修改后的抽象逻辑结构图 t_i 的有效性,我们前面给出的定理就能够保证对 a' 的概念源结点 a 的上述更新方法的有效性,从而使新的抽象概念结构图 t_c 与新的抽象逻辑结构图 t_i 保持一致。对逻辑导出结点 a' 的逻辑结点语义的修改以及由此而引起的概

念结点 a' 的概念语义的一致性更新属于语义范畴,主要由人工进行维护。

(2) 如果 a' 是逻辑扩展结点,即 t_c 的概念结点集合 A_c 中不存在任何概念结点 a ,使 $a' = f_{id}(a)$ 。因修改逻辑扩展结点不会对抽象逻辑结构图与抽象概念结构图的一致性产生任何影响,所以对 a' 结点的逻辑结点类型和逻辑结点语义的修改均不会引起对抽象概念结构图 t_c 的任何一致性更新操作。

结束语 本文提出的逻辑结点到概念结点的逆向映射规则以及双向映射函数的有效性定理为抽象逻辑结构图编辑操作引起的抽象概念结构图一致性更新问题的解决提供了理论基础,是抽象逻辑结构图可视化程序设计方法学中的重要内容,对于维护抽象概念结构图与导出抽象逻辑结构图的有效性与两者间的一致性具有重要意义。文中证明的双向映射函数的有效性的定理保证了两个规则集的无矛盾性和映射的封闭性。双向映射体系的构建,使程序员在设计过程中既可对抽象概念结构图进行编辑,又可对导出抽象逻辑结构图进行修改,使抽象概念结构图和抽象逻辑结构图对程序设计过程的支持变得非常灵活。

参考文献

- 1 刘建宾,郝克刚,龚世生. 抽象概念结构图到 JAVA 过程蓝图的平滑过渡及一致性. 计算机科学, 2001, 28(8)
- 2 刘建宾. JAVA 过程蓝图. 计算机科学, 2000, 27(7)