第34卷 第23期 计 算 机 工 程 2008年12月
Vol.34 No.23 Computer Engineering December 2008
·软件技术与数据库· 文章编号：1000—3428(2008)23—0068—03 文献标识码：A 中图分类号：TP393.09

基于GT2和MPICH-G2计算网格平台的设计

孙 峥，孙瑞志
(中国农业大学信息与电气工程学院，北京 100083)

摘 要：针对某些科学研究领域计算方法复杂、数据量庞大的问题，设计和构建了一个基于Globus的计算网格环境，能够发现可用计算节点，并动态提交计算任务。同时，Globus工具包2为构建网格环境和网格应用提供了可以直接调用的API函数接口，提高了并行计算的效率。通过运行一个典型的并行算法，验证了平台的可用性和高效性，并进行了结果分析。

关键词：网格；并行计算；Globus工具包

Design of Computing Grid Platform Based on GT2 and MPICH-G2

SUN Zheng, SUN Rui-zhi
(College of Information and Electrical Engineering, China Agriculture University, Beijing 100083)

【Abstract】Aiming at complex calculation method and massive data in some science research field, a Globus-based computing grid is designed and implemented, which can discover available nodes and submit computing jobs dynamically. Also, Globus Toolkit 2 provides API functions to construct grid environment and grid application, so the parallel computing efficiency is remarkably enhanced. Through running a typical parallel algorithm, the availability and high efficiency are verified, and a result analysis is given briefly.

【Key words】grid; parallel computing; Globus Toolkit

1 概述

目前，科学研究的许多领域涉及到的数据处理普遍具有计算方法复杂、数据量庞大的特点，超长的计算时间使使用者难以忍受，即使购买超级计算机也不能无限制地提高计算性能。计算网格可以为科学计算提供高性能计算环境，它充分体现了2大优势：(1)提供了很强的数据处理能力；(2)能充分利用网上的闲置计算资源。

计算网格的特点主要是对计算效率的要求很高，而在流行的网格支持平台中，Globus Toolkit 2(GT2)的主要特点是通过API函数提供服务，因为没有其他的开销，适合于支持科学计算任务。MPICH-G2的特点是通信性能高、程序可移植性好、功能强大，两者结合，能够极大提高并行计算的性能。

2 Globus工具包和MPICH-G2

2.1 GT2中间件简介

GT2是Globus Toolkit工具包的第2版，自1997年起成为网格计算的事实标准。它是一款基于社区、开放架构、开放源码的服务集合和软件库，用来支持网格和网格应用。GT2关注和解决的问题包括安全、信息发现、资源管理、数据管理、通信、故障检测以及可抑制性[1]，是基于5层沙漏模型体系结构的典型实现。

GT2对于实现科学计算而言，有其独特的优势：

(1)GT3, GT4虽然封装了GT2的GRAM(Globus Resource Allocation Manager), MDS (Metacomputing Directory Service)等功能，但都是以服务的形式提出来，使用了Java和Web Service的协议栈后，性能大打折扣；GT2直接调用API函数，使得计算效率得到大幅度提高，并且开发灵活、易于控制。

(2)对并行计算环境MPICH-G2的支持性较好。

(3)一般开发高性能计算都是在Linux, Unix平台下，采用C, Fortran来写算法，用的资源管理系统一般也是shell命令方式，在GT3, GT4平台上实现不是很方便。

2.2 并行编程工具MPICH-G2

MPI(Message Passing Interface)是目前比较重要的并行编程工具。它具有移植性好、功能强大、效率高等多种优点，而且有多种不同的免费、高效、实用的实现版本。它其实就是一个“库”，有上百个函数调用接口在FORTRAN 77和C语言中可以直接对这些函数进行调用，后来又进一步提供对FORTRAN 90和C++的调用接口[2]。

MPICH-G2是MPI-1标准的完全实现，并加入MPI-2标准中的客户机/服务器管理功能。它被构造成MPICH的网格化进程管理和通信模块。模块采用Globus工具集机制，克服了在异构多站点环境中安全、高效和透明执行程序所必须面对的困难，包括跨站点认证，处理多个具有不同特征的调度器的需要，协调进程创建，异构通信结构，可执行的分段以及标准输出的整理[3]。

3 计算网格平台的设计

Globus Tookit 2工具包仅仅提供了比较底层的API函数调用来实现网格应用中各种服务模块的功能，这种模式灵活、高效，但同时也不可避免地给开发过程带来困难，它要求项目开发者必须既精通计算机技术、网格计算技术，同时又要具备某些科学领域的专业背景。为了解决计算网格应用普及困难的问题，设计了一个比较通用的并行计算网格平台(Parallel Computing Grid Platform, PCGP)，为使用者提供一个比较透明的接口和友好的应用界面。

基金项目：国家“863”计划基金资助项目(2006AA10Z237)
作者简介：孙 峥(1984－)，女，硕士研究生，主研方向：计算机网络及应用，网格计算；孙瑞志，教授、博士
收稿日期：2008-04-05 **E-mail：**szh_cau0918@126.com

3.1 平台的体系结构

PCGP 能够屏蔽掉网格底层的相关细节内容，对计算网格有需求的用户只需要自主地上传并行算法和数据文件，然后等待计算结果的返回，其他的都交给网格平台管理体系进行处理。这样，网格应用开发人员可以专心于系统功能的设计和实现，使用者可以专心于专业研究领域的算法实现。通过 PCGP 的中间桥梁作用，使得实现网格应用对开发者和应用者来讲都不再是个陌生、棘手的事情。

为了方便实现和管理，设计采用传统的客户端/服务器模式。不同的是，中间要经过 PCGP 进行交互，PCGP 屏蔽了网格环境底层细节，实现了网格环境的各个管理控制模块。客户端是一个普通的 PC 机即可，对于机器的硬件性能没有特殊的要求，它是用户与网格平台进行可视化交互的一个环境。服务器端是计算网格的主要组成，包括 CA 认证中心、网格环境中计算节点池、数据服务器。

CA 认证中心：平台采用 Globus 提供的功能自建 CA 认证中心来管理网格环境，在一定程度上实现对加入网格环境中的计算节点进行身份验证和权限控制，每一个要求提交计算任务的节点都需要向 CA 认证中心申请获得有效授权的安全证书才可以与网格环境进行通信。

计算节点池：使用闲置的计算资源，要求节点能够提供比较稳定的计算能力。考虑到节点的动态加入和撤出的情况，设计一个计算节点池，能够将所有计算能力透明地提供给计算任务，而对于节点的匹配调度等均在计算节点池内部透明地进行处理。

数据服务器：科学计算往往具有庞大的数据量，因此单独设计了一组数据服务器。数据的表现形式也可以是文件系统、各种关系型数据库、XML 型数据库等。数据服务器采用数据集成技术对异构数据源进行合成、处理和统一访问。

整体的系统设计如图 1 所示。

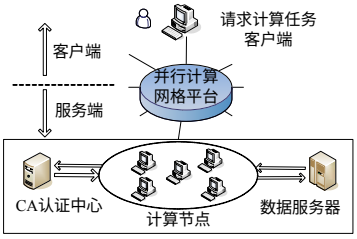

图 1　整体结构

3.2 计算网格平台的功能结构

平台的功能结构如图 2 所示，为了便于理解，图中给出了与 GT2 的 5 层沙漏体系结构的对应关系。

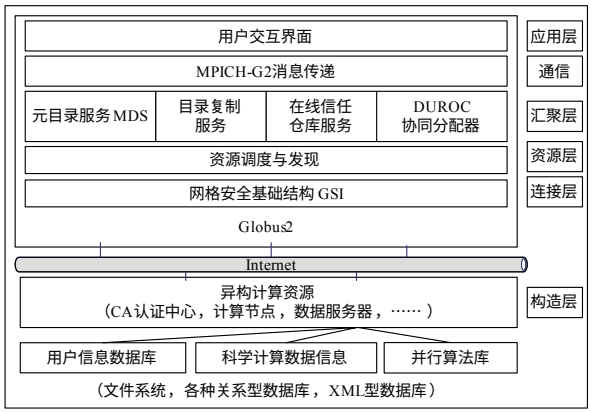

图 2　层次设计图与 5 层沙漏结构的对照

PCGP 基于 GT2 和 MPICH-G2 开发工具包，与异构计算资源之间通过 Internet 相连，并提供了简单、友好的用户交互界面。其中，构造层的计算资源在 3.1 节的整体设计中已经介绍过了，用户界面只需用户提供 MPI 并行算法和数据信息文件的路径，并上传至数据服务器即可，也比较容易理解。下面着重对其他主要功能模块进行说明。

3.2.1　网格安全基础结构 GSI

与集群系统不同，网格环境可以跨越不同的地域、单位、社区，甚至一些虚拟组织。网格系统可以动态扩展到互联网的各个角落，因此，安全通信问题是首先需要考虑的。

PCGP 直接采用了 GT2 提供的 GSI 实现网格安全控制。任务主机将自身安全证书的 subject 传给执行主机，通过 grid-mapfile 文件中设置的映射二元组，将被认可的用户安全证书的 subject 与远程计算节点上的账户 username 进行映射，得到远程计算节点的使用权，其权限范围与 username 账户的权限保持一致。

3.2.2　资源调度与节点发现

如何发现网格环境中的可用节点以及如何向可用节点动态地提交并行计算任务，是计算网格功能实现的核心内容。

在网格动态性和多样性的环境下，资源发现的最佳状态是：在查询资源信息的过程中，根据计算任务的不同，获取到不同的过滤条件，从而检索到与计算任务合理匹配的计算资源。

资源调度算法的设计，可以归结为 2 种方式：(1)找到最佳匹配的资源，即最优化方式；(2)找到一个可用的资源即可，即随机化方式。第 1 种方式中包括蚁群算法、模拟退火算法和遗传算法等[4]。但其实现起来比较复杂，算法本身效率不高。本文采用了后一种方式，是对FIFO调度算法的一种改进，即随机获得一个计算节点，看其是否满足条件，这是通过检查一个过滤条件变量filter来实现的。在过滤条件中，主要考虑了CPU主频大小、内存大小、网络带宽、存储系统大小等主要因素。这种方式的好处是实现简单，性能可靠，不占用计算资源，避免了处理复杂调度算法引起的调度延迟。

3.2.3　元目录服务 MDS

本文引入 GT2 提供的 GIIS(Grid Information Index Service)和 GRIS(Grid Resource Information Service)服务解决网格节点动态增多、检索耗时的难题。实现方法是将环境中的每个网格节点注册 GRIS，然后在某几台节点上为这些 GRIS 注册 GIIS，依次进行下去，形成了一个树型目录层次结构。将所有网格节点统一起来，形成一个集中的资源镜像。在检索节点资源的过程中，只需要向注册 GIIS 的最上层节点提交请求即可访问到所有子分支上的节点信息，简化了资源检索的方式。

本文的工作暂时不涉及目录复制服务和在线信任仓库的内容，以后将对此接口进行功能扩展。

4　基于 MPICH-G2 与 GT2 的任务调度

平台中的任务调度包括了运行程序的自动分发、任务的监控和执行、结果的获取等几个步骤。

4.1 MPI 程序的自动分发

MPI 程序的运行需要将生成的可执行程序分发到各个计算节点上，才能实现多台机器间的进程并行计算。以往，这个工作通常是手动完成的，MPICH-G2 与 Globus 的结合可以解决这个问题。利用 GT2 提供的 GridFTP 文件传输功能，

依据可用节点信息，实现了并行执行程序的自动分配。当计算任务完成之后，再将分配的计算程序进行动态删除，避免了硬盘被大量占用。利用 GridFTP 传输可执行程序到达目的节点后，程序权限为不可执行状态，GridFTP 不能控制远程计算机运行 shell 命令更改其权限，利用在提交传输文件任务的同时提交一个执行脚本的方式，实现了更改远程计算文件执行权限的目的。

4.2 动态提交任务

在 MPICH-G2 中，提交任务是通过 mpirun 命令完成的，mpirun 通过指定进程数目、mpi 可执行文件，向本地机提交并行计算任务。然而，该命令不能根据动态获取到的主机名向远程节点提交计算任务。相反，GT2 的 globusrun 可以根据 RSL(Resource Specification Language)文件的描述内容控制任务的动态提交，功能十分强大。为 mpirun 命令生成一个 RSL 文件，然后交给 globusrun 进行处理，即可实现根据发现可用节点的主机名称，远程提交任务的功能。

之后，MPICH-G2 通过调用 GT2 中的分布式协同分配器 DUROC(Dynamically Updated Request Online Coallocator)，在动态指定的计算节点上调度和启动应用。DUROC 连接库通过 GRAM 的 API 和协议启动自身，随后在每一台机器上管理一个子计算集。任务提交之后，利用任务的状态控制功能，实现对任务运行状态的实时监测，包括计算任务是否在正常执行，执行是成功还是失败，判断任务的挂起、激活等状态，并且在任务执行结束后，实现任务的撤销等功能。

4.3 进行多线程的访问控制

科学计算任务计算量庞大，执行时间较长。一旦提交运行后，就会占用大量的机器资源，容易造成机器对任何操作的不响应，同时影响对任务状态的实时监测。故应用异步调用机制进行线程控制。

采用异步调用有 2 大优点：(1)函数能够立即返回，所以执行一个计算量大的Globus任务时可以在短时间内得到反馈；(2)可以使应用处于一个积极的地位，而让Globus体系去管理任务状态的检测操作，应用只需与Globus环境中的任务保持联络即可[5]。然而，这种任务提交模式会造成多个线程对同一变量、数据文件、计算资源的同时访问，从而不可避免地导致错误的出现。为了对其进行互斥控制，需要通过P, V原语并设置警示变量的方法，避免线程间的死锁和竞争问题。

4.4 任务调度流程算法

整个任务调度流程算法描述如下：

(1)获得用户上传的 MPI 算法源程序路径，将该文件上传到数据服务器上，加入并行算法库。

(2)对算法源程序进行编译，若编译失败，则提示错误，返回(1)；成功，则继续执行。

(3)获得计算所需的数据文件路径，将该文件上传到数据服务器上，加入科学计算数据信息库。

(4)根据计算任务的特点和给出的资源限制条件检索出可用节点。

(5)向参与计算的节点分配 MPI 可执行程序和数据文件。

(6)任务与计算资源之间进行匹配，并动态提交计算任务。如果任务执行失败，则删除失败节点上的算法程序和数据文件，返回(4)；成功，则执行计算任务。

(7)返回计算结果和任务执行过程中的状态检测信息。

5 平台验证及结果分析

为验证平台资源调度算法的可用性，运行了一个典型的计算π值算法。该近似计算方法的原理，是在 0~1 上对函数 $f(x)=4/(1+x^2)$进行积分得到π值，进而转化为计算$f(x)$图像下面从 0~1 之间的面积。随着所划分小矩形数量的增多，精确度提高，然而计算时间却随之增加。该算法进程间交互不频繁，有很好的并行性，适合作为验证该平台的实例。算法中利用了MPICH-G2 提供的组通信的广播和归约操作实现了进程的并行[2]。图 3 显示了进程数(np)为 1~13 时，执行该算法所花费的时间。

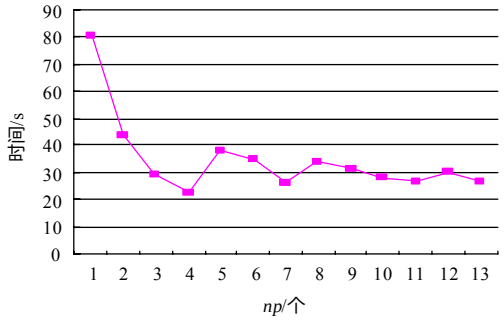

图 3 并行算法执行时间折线图

实验结果表明，该并行计算网格平台在进行多进程并行计算时，性能得到了很大的提高。但是，可以看到，在计算时间相对不长时，随着进程数的增多，机器之间的网络延迟和进程间的通信延迟将显著影响计算性能，因此，需要在进程数目和每个任务的计算量上寻找平衡点，以达到网格平台总体计算效率最高。

6 结束语

本文分析了利用GT2.4+MPICH-G2搭建并行计算平台的优势，阐述了设计思想和关键技术的实现，并应用一个典型的 MPI 并行计算程序，验证了平台的可用性和高效性。以后的工作中将重点完善各个功能模块，考虑进一步改进调度算法以实现更强大的调度策略，以及完成异构数据集成部分和用户角色权限控制等内容。

参考文献

[1] Foster I, Kesselman C. 网格计算[M]. 金 海, 袁平鹏, 石 柯, 译. 北京: 电子工业出版社, 2004.

[2] 都志辉. 高性能计算之并行编程技术——MPI 并行程序设计[M]. 北京: 清华大学出版社, 2001.

[3] 都志辉, 陈 渝, 刘 鹏. 网格计算[M]. 北京: 清华大学出版社, 2002.

[4] 须文波, 张 涛. 网格计算资源调度算法研究[J]. 计算机工程, 2006, 32(14): 95-97.

[5] Girard J Y. Use Globus Toolkit 2.4 and C++ Classes to Submit Grid Jobs[EB/OL]. (2003-05-01). http://www-128.ibmcom/developerworks/grid/library/gr-cglobus/index.html.