跨地域高性能科学计算及其可视化的设计和实现*)

Design and Implementation of a Common Platform of High Performance Computing and Visualization Based on Internet

刘 鹏 何 川 李三立 都志辉 黄震春 时培植

(清华大学计算机科学与技术系 北京 100084)

Abstract A new platform, which connects distributed High Performance Computing (HPC) resources in the country and provides a user-friendly Web interface to the general researchers in different fields, is presented in the paper. Users can put forward large HPC tasks through Internet browser, watch the running status of the task, and get the result along with the visualization effect in the end. The system will promote the popularization of HPC and inter-fields research.

Keywords Advanced Computing Infrastructure (ACI), High Performance Computing, Visualization, Cluster, Database Middleware

1 引言

高性能计算(High Performance Computing, HPC)技术对于提高国家科技水平、增强国家实力,都具有重要意义。美国的高性能计算计划 HPCC 和 ASCI[1]分别是在布什和克林顿总统参与下制定的。在科学技术方面,高性能计算是推动大规模科学与工程计算的不可缺少的工具平台;它可应用于新材料分子结构计算、飞行物流体/固体力学的计算、高精度石油勘探的计算、人类基因组计划、分子动力学计算,等等。国际上普遍认为:21 世纪人类面临的所谓"巨大挑战"(Grand Challenge)[2]的科技问题,如海洋循环、黏流动力学、全球准确气候预报等等,没有万亿次 (TeraFlops)以上的高性能计算机系统,是不可能彻底解决的[3]。

目前我国已经具有一定数量的超级计算机,某些超级计算机的峰值速度已达到国际先进水平。然而,超级计算机造价昂贵、不便移动、操作复杂,只能安装在某些特定的机构,供有限的技术人员使用,这严重影响了我国高性能计算技术的普及。因此,我们提出了建立我国的先进计算基础设施 ACI (Advanced Computing Infrastructure)和成立高性能计算联盟的构想。所谓 ACI,是指把地理位置分布的信息处理资源(人员、高性能计算机、仪器、数据库等)用高速网络连接起来,再用中间件软件(操作系统、网络平台、资源管理软件、应用工具等)有机粘合在一起,为各处的科学工作者提供一个单一映像的虚拟计算与处理平台,使大家可共享这个平台上的资源,共同处理科研和教学中的问题[4,5]。

ACI 北京上海试点工程如图 1 所示。其中,北京的集群式计算机[6]THNPSC-2[7]峰值速度达到 1000 亿次浮点运算;上海的集群式计算机"自强 2000"峰值速度达到 4500 亿次以上浮点运算,由此初步构成网上 5000 亿次/秒以上峰速的高性能计算能力;并在华东理工大学和清华大学系级单位建设网络终端系统,使这些单位能共享 ACI 试点工程的高性能计算资源。

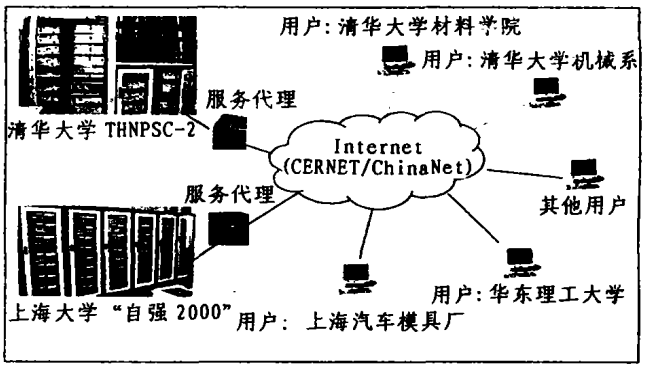

图 1 ACI 北京上海试点工程构成图

作为 ACI 试验床,我们设计并开发了一套基于 Web 的远程计算平台原型,基于 Web 技术实现了一个简单易用的前端用户的可视化使用界面,并开发了后端系统管理和任务调度系统,实现了异地超级计算系统的自动调度。另外,由于高性能科学计算的结果一般都有很大的数据量,直接使用相当不直观,为此我们针对一些典型应用,开发了相应的可视化软件,使得用户可以在异地通过浏览器观察高性能计算的三维动态可视化结果。这个平台使非专业人员远程使用超级计算机成为可能,将推动我国高性能计算的发展。

2 远程计算服务平台的设计和实现

2.1 远程计算任务的提交和处理流程

ACI 系统单个超级计算机结点的逻辑结构如图 2 所示。远程用户通过基于 Web 的人机界面提交任务请求。该请求经过 Internet 和 Firewall 到达 Web Server。Web Server 通过一个中间层进行安全认证和授权,将任务的详细信息提交到 Database Server。后台的 Agent 通过中间件从 Database Server 中将任务取出,进行安全性检查,然后将其放到任务队列(Task Queue)之中进行调度。Agent 同 Supercomputer 进行通信,将任务提交给超级计算机,控制任务的启动、停止、中断和恢复等,并且取回任务的断点和结果,进行一定的处理后,

*)本课题受教育部重点项目"先进计算基础设施北京上海试点工程"资助。刘 鹏 博士生。何 川 硕士生。李三立 中国工程院院士,博士生导师。都志辉 博士后。黄震春 博士。时培植 博士生,主要从事高性能微体系结构、分布/并行体系结构的研究工作。

通过中间件存放到 Database Server 之中。在这个流程中，Agent 处于核心位置，它实际是存在于一台或多台计算机上的若干个服务程序的统称。这些程序合作完成大量计算任务的调度管理、断点和结果的收集、预处理、超级计算机的安全保障、超级计算机的状态监控等一系列功能。

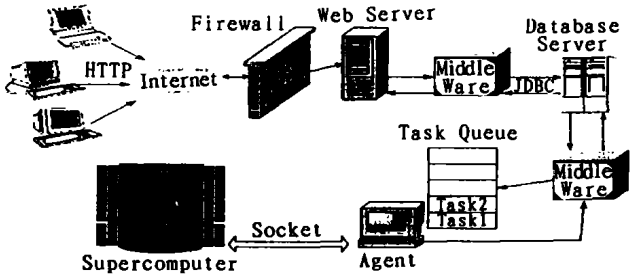

图 2　单个超级计算节点的结构图

2.2　ACI 软件平台的实现思想

在 Web Server 前端的开发上，我们使用了 Servlet＋Applet 技术，利用两种技术在安全性、可移植性、高性能、可视化方面的多种特点，开发出了简单易用的人机交互界面；我们把管理系统运行的数据库系统放在了中心位置，并在对数据库的访问上，以 JDBC 为基础设计实现了一个功能强大的数据库中间件，并提出了数据库中间件分层设计的思想，较好地解决了 Web 数据库的互联问题；在 Agent 的制作中，我们使用了多线程的机制，并对单个任务的生存周期建立了有限状态机模型，使整个系统的核心——Agent 更加稳定并且具有同时处理多种任务的能力；另外，我们使用流套接字（Stream Socket）和 Shell Script 相结合，利用 Shell Script 在 Unix/Linux 系统的强大功能和流套接字的面向连接的稳定通信能力，完成了 Agent 对超级计算机系统的控制和二者之间的通信问题。在此，我们对一些重要的实现思路进行阐述。

2.2.1　*用中间件技术实现对数据库的透明访问*　ACI 涉及到大量的数据信息：因为允许用户在任何地方通过 Internet 用户终端访问使用超级计算机，所以需要处理用户帐户、权限和费用信息；由于要让用户可以远距离查询自己计算结果和系统运行状态（包括处理机利用率、内存利用率和通信开销，等等），从而调整自己并行算法，以取得优化的计算效果，因此海量的计算结果信息和运行状态信息必须易于远程访问；更进一步，为了把高性能计算的结果和经验做成类似建筑业的“预构件”，使后续用户不再从“砖头”开始建房，需要建立高性能计算的应用库，并通过数据挖掘，来提炼高性能计算中的特有和共性数据。

对于这些数据信息，如用户帐户信息、系统配置信息、运行状态信息、科学计算相关信息等，我们的指导思想是尽可能将其置于数据库系统的统一管理之下，让所有的管理和服务工作围绕数据库展开。这样做带来的好处是：易于管理、标准化、与平台无关、可移植性好。

为了简化对数据库的访问，提高数据库访问的效率，ACI 使用数据库中间件来实现对数据库的封装和访问。我们将整个数据库服务体系分为四层（如图 3 所示），每一层向上一层提供接口，为上一层服务：第一层为 Database Connection Layer，使用 JDBC 数据库编程接口提供基础的数据库连接服务，并向数据库发送 SQL 请求和接收结果。第二层为 Abstract Database Layer，使用第一层提供的接口，对关系数据库和数据表进行封装，提供了更易于使用的 API 。第三层是 Service Layer，它使用第二层的接口，含有一系列的服务模块，包括：用于快速生成 HTML 页面的 HTMLFile，进行盘错处理的 Valid，对结果信息进行解析处理的 Parsers 等。第四层是 Application Layer，它包括在 Web Server 和 Agent 之上运行的数据库应用程序，它们调用第三层中的服务模块完成特定的功能。

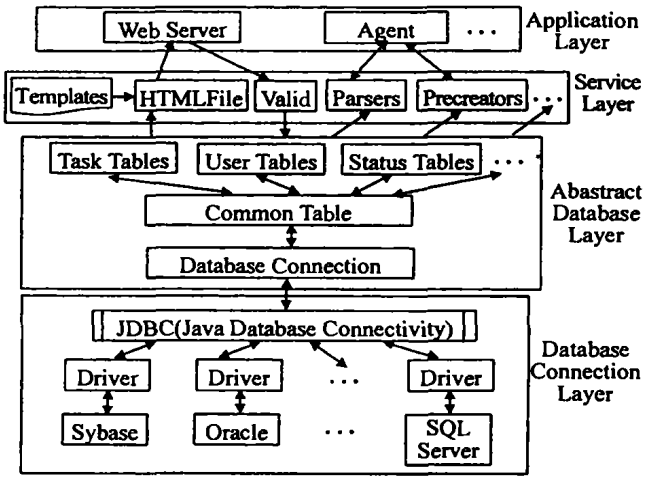

图 3　数据库中间件结构图

采用这种数据库中间件带来的好处是：

易用性：Abstract Database Layer 隐藏了复杂的 JDBC 编程接口、关系数据库结构及 SQL 语法，开发人员只需要使用一些简单形象的接口函数，就可以操作数据库，使得在 Service Layer 上编写服务模块更加简单和高效。

可移植性：Abstract Database Layer 是一个相当通用的模块，它使用 Java 编写，内部使用标准的 SQL 语句对数据库进行操作，只要在 Database Connection Layer 使用不同的 JDBC Driver，就可以应用于不同的关系数据库，在其它系统的开发中也可以重复使用。

可扩展性：由于底层的可靠支撑，我们只需要在 Service Layer 上增加服务模块，就可以为 Application Layer 提供更加强大的服务。

2.2.2　*用多线程机制实现任务调度与运行支持*　ACI 的任务调度与运行支持由 Agent 程序完成，它包括以下几种功能：

1. 环境变量维护：检查程序的环境变量设置情况，如任务队列长度、垃圾清除时间等，随时动态调整 Agent 的工作性能。

2. 垃圾数据清理：定期清除数据库中过期的数据和超级计算机上无用的目录和计算程序等。

3. 超级计算机状态搜集：定期查询超级计算机的内存使用、CPU 占用及各节点通信情况等信息，保存到数据库之中。

4. 任务调度：根据超级计算机的负载情况，使用一定的负载均衡调度算法对提交上来的任务进行动态的调度。

5. 任务断点和结果收集：收集正在运行的任务的断点和已经完成的程序的结果文件，经过一定的处理之后，保存到数据库之中。

由于 Agent 程序需要同时完成上述多种事务工作，因此必然要采用多线程机制。多线程是指在单个程序中使用多个运行线索，这些线索在同一时间并发运行，执行不同的任务。

我们将一个任务可能的状态分为六种：等待、执行、中断、

完成、结束、清除。图 4 给出一个任务从等待到清除的有限状态机描述，从状态机图中可以清晰地看出 Agent 中多线程的工作流程和管理实施情况。

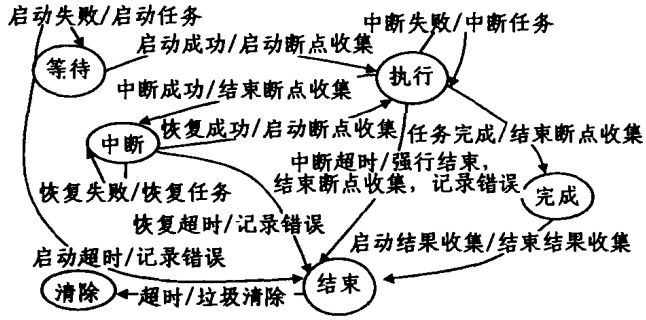

图 4　任务生存期的有限状态机

在多线程程序的设计过程中，我们还特别注意了以下几点：

1. 充分的出错处理。例如，在调度失败时随即释放占用的资源，所以不会出现任务永远不结束的情况。

2. 服务线程在不工作时处于睡眠状态，释放不必要占用的资源。

3. 由于多个线程需要同时读写数据库，因此存在临界区的冲突问题，需要采取同步机制对多线程读写进行控制。

2.2.3　*用流套接字和 Shell Script 完成对超级计算机的控制*　套接字(Socket)是在网络上运行的两个程序之间双向通信的一种机制。常用的 Socket 类型有两种：流式和数据报式。流式 Socket 是面向连接的 Socket，针对面向连接的 TCP 服务应用；数据报式 Socket 是无连接的 Socket，对应于无连接的 UDP 服务应用。在 ACI 中，Agent 同超级计算机之间的通信使用了流套接字，图 5 显示了流套接字的工作流程。

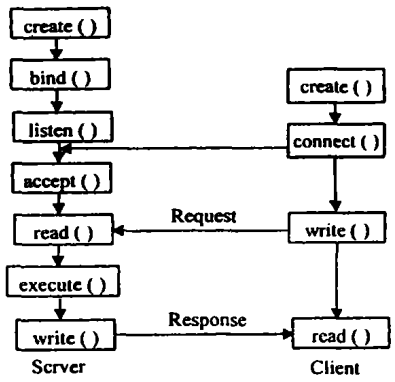

图 5　流套接字的工作流程

使用流套接字和超级计算机进行通信的好处在于：

1. 平台无关性：由于各种不同的操作系统都提供了流套接字的编程接口，而我们编写 Agent 所用的 Java 语言是平台无关的，这样就可以实现 Agent 与底层操作系统的平台无关性，从而同一个 Agent 程序可以对运行在异构操作系统上的不同超级计算机进行控制。

2. 便于管理和控制：由于流套接字是一种面向连接的稳定的通信方式，Agent 可以通过它和超级计算机实现稳定的交互，不会出现通信不稳定的情况。

3. 安全：因为可以使用固定的端口进行控制通信，故破坏者突破了 Firewall 之后不能从 Agent 所在的机器对超级计算机进行攻击，增强了系统的安全性。

我们在超级计算机上的控制模块使用 Linux Shell Script 编写。我们编写了对超级计算机进行控制的一些基础模块，如 Start，Stop，Break，Resume 等。Agent 程序调用这些基础模块，实现任务的启动、停止、中断、恢复等功能。

3　跨网可视化系统的设计和实现

可视化是用户界面友好的重要标志，它使系统容易被用户接受，从而增强系统的实用性。在 ACI 中，我们实现了系统运行状态和运行结果的可视化，并由 Web 方式提交给异地 Internet 用户浏览。

3.1　运行状态的可视化

ACI 利用上述的 Web Server＋Database Server＋Agent 的服务框架提供可视化服务。在这个框架之上，系统需要提供三个可视化功能模块：数据采集、数据存储、图形生成。

数据采集可以通过一个后台的 Daemon 程序实现。它不断扫描机群系统的资源情况，并把原始数据存入数据库。数据采集的过程与数据本身的类型相关。我们把需要采集的信息分为三类：(1) 静态信息，是指不会随时间而变化的信息，比如机群系统的处理机、操作系统、最大存储容量、最大磁盘容量等。静态信息是在 Daemon 启动时，一次采样得到的；(2) 集合信息，是指需要由 Daemon 按照一个固定的间隔时间采样的信息。它的内容是一个随时间轴变化的集合，如处理机忙碌程度、内存占用程度、排队等待计算的任务数、可用磁盘空间等，通常是负载指标；(3) 快照信息，是指系统在某一个时刻瞬间的信息，与集合信息的自动生成不同，这个信息只有在用户请求时才由 Daemon 生成。

数据存储则需要先定义这些数据在数据库中的表结构，然后按照数据表的格式，把数据存入数据库。在 ACI 中，图形生成采用 Java Applet 的方式：用户通过浏览器访问嵌入特定 Applet 的 Web 页面，然后这些 Applet 从 Web Server 获取数据库中的数据并绘制出相应的图形。由于 Applet 不是在服务器端绘好后再传给客户，而是用数据控制客户端的绘制过程，大大减少了网络数据的传输量，并能够支持与用户的动态交互。

对高性能计算机运行状态的可视化实现了以下功能：

1. 系统全局动态资源的检测与分析：分析和显示处理机利用率，内存占用比例，外存剩余空间等，如图 6 所示。

图 6　系统处理机使用记录

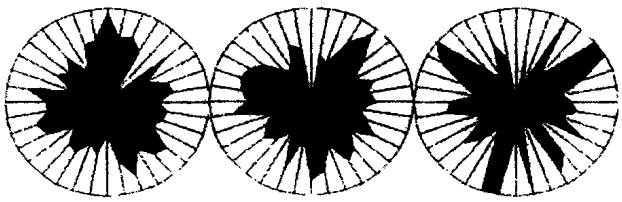

图 7　处理机利用率雷达图

2. 与应用程序相关的资源检测分析：它包括动态与静态

的分析。动态的分析指按照时间轴的变迁，以图形的方式显示出这个应用程序对系统资源的占用程度；静态的分析，指对整个应用程序而言，各种资源占用的总的比例。

3. 应用程序运行过程的重播：指通过容易观察的方法把一个应用程序的执行过程回放出来。例如对基于MPI的应用程序，可以直观地看出通信的处理机动作及消息的流向。

3.2 运行结果的可视化

高性能计算结果的可视化可以分为两大类：

1. 物理意义的可视化：应用程序通过科学计算后，可得出一系列有一定物理意义的计算结果。将它们用一定手段给用户直观显示，可帮助用户进行定性分析，例如图8所示模拟退火算法的物理意义。

图8 模拟退火算法的物理意义图

2. 真实状态的模拟：在绝大多数情况下，进行高性能计算的科技工作者都希望能在计算机屏幕上，看到所研究对象（如分子结构、大气运动、机械铸件等）在客观世界中的真实存在状态。更有意义的是，对非常态条件下（例如，超高温或超高压）对象的模拟，可以使用户直观地观察在实践中难以观察到的物质状态。

ACI对实现科学计算结果的可视化提出了较高的要求：一方面，客户是通过Internet观察这些结果的，没有事先安装任何软件；另一方面，高性能计算结果的可视化数据规模又非常大，必须用动态三维图像展现。显然，如果采用OpenGL等传统的可视化手段，只能在Web服务器端生成图像，再一幅幅地传给客户端显示，在当前的Internet传输能力下，是不可能满足网络环境下高性能计算的可视化要求的。因此，传统的可视化手段难以同时满足这两个目标。

而SUN公司新推出的Java 3D[8]能够很好地解决这个问题：它是面向Internet的三维开发环境，继承了Java的代码可传输性，使得用来生成可视化场景的小巧的Applet可以方便地从服务器传给客户端，然后在客户端本地运行。也就是说，传输的不是图像本身，而是控制三维图像生成的程序和数据，因而大大节省了网络传输数据量。相比另外一种Internet三维可视化工具VRML，Java 3D的功能和可编程性更强：它本身就用Java书写，与在网络环境下的多个计算机群体之间实现了无缝结合；它像Java一样，书写一次，就可以跨平台运行；它具有VRML所没有的形体碰撞检查等功能；另外，可以充分利用Java语言的强大功能，编写出复杂的三维应用程序。

因为采用Java 3D，我们实现了从Internet自动安装用户端运行环境，并能通过传输控制三维模型的代码，在本地快速生成可视化图像；同时，结合Applet与Servlet交互技术，在用户观察当前可视化结果时，分解传输有效数据来控制三维模型后续的运动，使跨网络的动态可视化效果接近于本地运行的水平。

4 ACI应用举例

例1 *高分子材料微观结构的模拟* 研究高分子对胶体分散体系的稳定作用和絮凝作用，在涂料工业、高分子复合材料、石油二次开采、煤液化、粘合剂等工农业领域具有重要意义。在对高分子链的研究中，用工作站计算16个分子链，负担已经过大。使用了ACI后，研究对象的规模比以前扩大了很多，其可视化结果如图9所示，它表现了分子链的动态微观结构。目前正在研究"高分子膜"的计算，计算量极大。

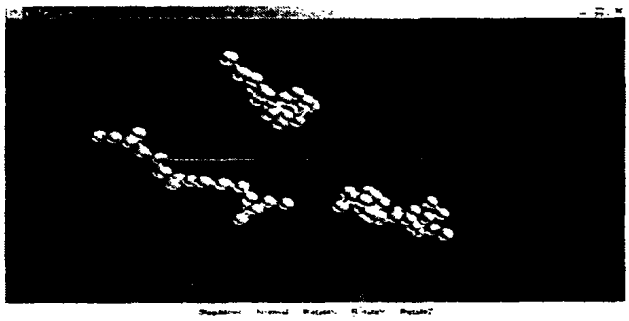

图9 高分子材料微观结构的可视化结果

例2 *石油化工催化的超临界化学反应工程* 在极高温、极高压条件下的超临界反应实验很难做。为了尽量少做实验，或做较易实验，现在通过高性能计算机大量计算，用概率统计方法逼近，研究临界相物质内的一些特殊性质，其可视化结果如图10所示。通过动态的可视化过程可以观察到分子的分布状况随时间动态变化，在某一瞬间达到临界值而发生突变，随后处于变化较小的相对稳态。根据这个结果，就可以在给定的高温、高压及催化剂条件下找到材料的临界点。

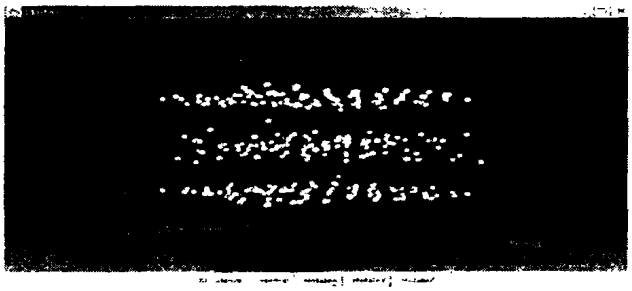

图10 超临界化学反应的可视化结果

例3 *模具压铸过程计算机模拟技术* 采用数值模拟技术求解压铸充型过程中液态金属流动和传热的偏微分方程，获得成形过程中任意时刻流体的自由表面形貌、速度场、压力场和温度场信息，可以实现理想的型腔充填和模具的热平衡状态，从而提高铸件质量和延长模具的使用寿命（如图11所示）。

图11 模具铸造过程的三维可视化图像

（下转第13页）

之一，因此如何改善设备驱动程序、在协议中加入错误控制等特征是进一步研究的方向之一。

今后还可以对用户级通信软件做进一步的优化和实用化工作，使得它既能充分发挥通信硬件的性能，又能方便地用于工具软件和应用程序的编程。另外，可考虑在将来的硬件设计中，改进网络接口的功能，如：实现 FIFO 的流水线工作方式，增设通信处理器以及提供 DMA 机制等。

为了在商业计算机组成的网络基础上建立高效的服务器，今后可以进一步探索把网络协议部分或全部移至用户层的方法，以此来减少协议开销。总的框架是实现一个用户层的数据传输协议，并使其与网络接口较好地结合起来。这样可以高效地在共享虚拟内存和消息传递等模型上实现已有的高层协议，从而把硬件的大部分性能传递给用户通信，同时提供高层 API 的全部功能。共享虚拟存储模型是用户比较容易接受的编程方式，研究通信系统如何高效支持这种编程模型，解决通信瓶颈问题是非常有意义的。

机群系统的发展对通信技术提出了更高的要求，要解决上述问题，仍需我们不断的努力。

参 考 文 献

1 Bruck J，Dolev D，Ho Ching-Tien. Efficient Message Passing Interface (MPI) for Parallel Computing on Clusters of Workstation. Journal of Parallel and Distributed Computing，1997，40：19～34

2 Pakin S，Karamcheti V，Chien A A. Fast Message(FM)：Efficient，Portable Communication for Workstation Clusters and Massively-Parallel Processors. IEEE Concurrency，1997，5(2)：60～73

3 Lauria M，Chien A. MPI-FM：High Performance MPI on Workstation Cluster. Journal of Parallel and Distributed Computing，1997，40：4～18

4 Lauria M，Pakin S，Chien A. Efficient Layering for High Speed Communication：the MPI over Fast Message (FM) Experience. Cluster Computing，1999，2(2)：107～116

5 Jacunski M，et al. Low Latency Message-Passing for Reflective Memory Networks. In：CANPC'99，LNCS 1602，1999. 211～224

6 von Eicken T，et al. Active Messages：a mechanism for integrated communication and computation. In：Proc. of the Intl. Symposium on Computer Architecture，1992. 256～266

7 Rodrigues S，Anderson T，Culler D. High-performance local-area communication using Fast Socket. In：Proc. of the USENIX 1997 Technical Conf. San Diego，California，USENIX Association，1997

8 Yocum K G，et al. Cut-through delivery in Trapeze：an exercise in low-latency messaging. In HPDC-6，Portland，Oregon，August 1997

9 von Eicken T，et al. U-Net：A user-level network interface for parallel and distributed computing. In：Proc. of the 15th ACM Symposium on Operating Systems Principles. Dec. 1995. 40～53

10 Dubnicki C，et al. VMMC-2：efficient support for reliable，connection-oriented communication. In：Proc. of Hot Interconnects V. IEEE，Aug. 1997

11 Tezuka H，Hori，A，Ishikawa Y. PM：A high-performance communication library for multi-user parallel environments：[Technical Report TR-96-015]. Tsukuba Research Center，Real World Computing Partnership，Nov. 1996

12 Prylli L，Tourancheau B. Protocol design for high performance networking：a Myrinet experience：[Technical Report N. 97-22]. LIP，Ecole Normale Superieure de Lyon，July 1997

13 Leffler S J，McKusick M K，Karels M J，Quarterman J S. The Design and Implementation of the 4.3 BSD Unix Operating System. Addison-Wesley Publishing Company，Inc.，Redding，Massachusetts，1989

14 董迎飞. 分布式存储多机系统中通信技术的研究：[清华大学博士学位论文]. 1995

15 陈国良. 并行计算-结构. 算法. 编程. 高等教育出版社，1999

16 见网址：http://www.myri.com/myrinet/overview/index.html. Myrinet Overview

17 见网址：http://www.myri.;com/scs/index.html. The GM API

18 申俊，郑纬民，王鼎兴，沈美明. 提高工作站机群系统通信性能方法的研究. 小型微型计算机系统，1997，18(6)：8～13

（上接第 4 页）

结论和展望 ACI 试验床通过高速网络将分布在各地的超级计算机连接起来，形成整体高性能计算能力，为各个地方、各个学科专业的用户提供高性能计算服务，促进高性能计算在国内教育与研究领域的应用。

由于 ACI 的根本目的是推动高性能计算在我国的普及，为此我们在系统与用户的接口方面做了大量的工作。一方面把整个系统的使用环境建立在 Web 之上，使得用户可以通过 Internet 提交、观察和控制任务并得到运行的最终结果；另一方面，我们大量使用了可视化技术，以直观简洁的方式反映运行的状态和结果，消除非计算机专业人员使用高性能计算机的障碍。不同学科的科技人员，已经利用 ACI 完成了一些实际课题的研究工作。它充分体现出了这个平台的优越性。

我们今后的工作是将 ACI 系统建设成为一个全国范围的虚拟机房，建立共同的安全与授权标准、分布式调度、计算模块库和计费管理系统，提供海量存储能力和更加方便的用户接口等。随着网络由窄带向宽带、有线向无线的扩展，以及内嵌处理器、传感器和摄像机等各种先进设备的不断加入，ACI 系统平台将进一步成长，必将推动高性能计算技术在宇宙学、天文物理学、地球科学、流体力学、工程学、生物学和化学等更多学科中的应用。

参 考 文 献

1 ASCI Project. http://www.lanl.gov/projects/asci/asci.html，http://www.llnl.gov/asci/，http://www.sandia.gov/ASCI/，http://www.sandia.gov/ASCI/，http://www.lanl.gov/asci/，http://www.lanl.gov/projects/asci/asci.html

2 Grand Challenges：High-Performance Computing and Communications. Committee on Physical，Mathematical，and Engineering Sciences，National Science Foundation，Washington，D. C.，1991

3 Bell G. Ultracomputers：A Teraflop before Its Time. Communications of the ACM，Aug. 1992. 27～47

4 Foster I. Internet Computing and the Emerging Grid. Nature，7 December 2000. http://www.nature.com/nature/webmatters/grid/grid.html

5 Foster I，Kesselman C. Computational Grids：The Future of High-Performance Distributed Computing. Morgan Kaufmann Publishers，1998

6 Gutmann M J. Cluster PCs for Power. BYTE，July 1993. 57～64

7 Li Sanli，Li Yamin，et al. Q-Net：A Fast Interconnection Network Switch for NPCS-1 Parallel Systems. In：The 10th Intl. Conf. of Parallel and Distributed Computing Systems，Las Vegas，Nevada，USA，2000. 28～31

8 The Java 3D API Specification. Sun Microsystems，Inc.，2000