

文章编号:1007-130X(2009)A1-0307-05

# 虚拟化技术在高性能计算机系统中的应用研究<sup>\*</sup>

## The Application Research of Virtualization Techniques for High Performance Computer Systems

钱 磊, 李宏亮, 谢向辉, 陈左宁

QIAN Lei, LI Hong-liang, XIE Xiang-hui, CHEN Zuo-ning

(江南计算技术研究所, 江苏 无锡 214083)

(Jiangnan Institute of Computing Technology, Wuxi 214083, China)

**摘 要:**随着高性能计算机系统规模的不断扩大,体系结构的日益复杂,其管理和使用也变得越来越困难。将虚拟化技术应用于构建高性能计算机系统有助于帮助开发人员高效管理和使用高性能计算机系统,使其发挥最大效能。本文详细分析了虚拟化技术在高性能计算机系统中已有的几个成功应用,总结了虚拟化技术应用于高性能计算领域时所面临的主要问题,最后对虚拟化技术在高性能计算机系统中的应用前景进行了展望。

**Abstract:** With the expanding of the size of high performance computer (HPC) systems, the architecture is becoming more complex. It is quite difficult for users to manage and use such a complex system. Therefore, it is a feasible way to apply virtualization techniques to high performance computing to assist developers to manage and use HPC systems efficiently. In this paper, we analyzed several existing applications of virtualization techniques for HPC systems and concluded key problems in constructing HPC systems with virtualization techniques. At the end of this paper, we prospected the foreground of virtualization techniques for HPC systems.

**关键词:** 虚拟化技术; 虚拟机管理器; 高性能计算机

**Key words:** virtualization techniques; virtual machine monitor; high performance computer

doi:10.3969/j.issn.1007-130X.2009.A1.087

中图分类号: TP303

文献标识码: A

## 1 引言

随着高性能计算机性能的提高,其规模不断扩大,各部件之间的关系日益复杂,对系统的管理和使用也变得越来越困难。为了获得较高的性能和效率,高性能计算机系统软件必须根据体系结构的特点进行定制,高性能计算应用程序也需要根据特定体系结构进行优化。这样不仅大大增加了系统软件和应用程序的开发时间,加大了开发难度,还降低了应用的可移植性。此外,越来越大的系统规模和越来越高的软硬件复杂性也降低了系统可靠性。特别是高性能计算机体系结构向多层次、多粒度异构化方向的发展使得上述问题变得更为突出。如何高效管理和使用高性能计算机系统,使其发挥最大效能,已成为系统软件和应用开发人员亟待解决的关键技术难题。将虚拟化技术应用于构建

高性能计算机系统是解决上述问题的一个有效途径。

虚拟化技术最早出现在上世纪六、七十年代的 IBM 大型机系统中,用于支持高层软件共用底层硬件资源,为用户提供多个应用软件运行环境。到了八十年代,由于多任务操作系统的出现、硬件费用的降低以及个人电脑的流行,虚拟化技术逐渐淡出人们的视野。随着计算机软硬件技术的进一步发展,硬件性能得到了显著的提高,操作系统功能也越来越丰富,但系统可靠性和安全性却变得越来越差。为了提高系统可靠性,系统管理员重新采用以往一台机器运行一个应用的计算模型,但这种计算模型会导致更多的硬件成本和管理开销。而虚拟化技术正好满足了上述需求。将在多台物理机器上运行的程序放置在虚拟机(VM)上运行,可以节约硬件设备成本,提高资源利用率,降低管理的成本,通过虚拟机之间相互隔离,可以有效提高系统可靠性和安全性<sup>[1]</sup>。目前,虚拟化技术在台式电脑和应用服务器

\* 收稿日期:2009-07-13;修订日期:2009-09-10

基金项目:国家 973 计划资助项目(2007CB310907);国家 863 计划资助项目(2007AA01Z117)

作者简介:钱磊(1982-),男,江苏常州人,硕士,助理工程师,研究方向为虚拟化技术和高性能计算机体系结构。

通讯地址:214083 江苏省无锡市 33 信箱 031 号;Tel:(0510)85155628;E-mail:qianlei0808@163.com

Address:P. O. Box 33-031, Wuxi, Jiangsu 214083, P. R. China

领域已经取得了很好的应用效果,拥有如 VMware Workstation、Virtual PC 和 Xen 等一批成功的虚拟化软件<sup>[2]</sup>。硬件厂商也纷纷加入虚拟化的行列,推动虚拟化技术的发展,如 Intel 的 VT-x、VT-i 和 VT-d 技术<sup>[3,4]</sup>,AMD 的 Pacific 技术<sup>[5]</sup>等。

与在服务器领域的广泛应用相比,虚拟化技术在高性能计算领域的应用还相对较少。本文首先详细介绍目前虚拟化技术在高性能计算领域已有的几个成功应用;然后分析总结虚拟化技术应用于高性能计算机系统时所面临的主要问题;最后对虚拟化技术在高性能计算中的应用前景进行了展望。

## 2 虚拟化技术在高性能计算机系统中的应用研究

目前,国外一些公司和研究机构已经成功地将虚拟化技术应用到了高性能计算中,并取得了一些研究成果。这些应用包括了异构资源整合、系统容错、虚拟环境构建以及并行编程等多个方面。

### 2.1 整合异构资源——Cray 的自适应超级计算

在异构资源整合方面,Cray 公司启动了一个 Cascade 计划,该计划融合了虚拟化技术的思想精髓,提出自适应超级计算(Adaptive Supercomputing)的设想,目标是将现有的几种不同的超级计算技术整合到一个统一的高性能计算平台中<sup>[6,7]</sup>。如图 1 所示。Cray 公司认为,高性能计算的将来将各种异构资源集成到同一系统中。为了得到较高的应用程序处理性能,也为了满足用户的高生产率的要求,高性能计算系统必须综合利用各种类型的处理技术。

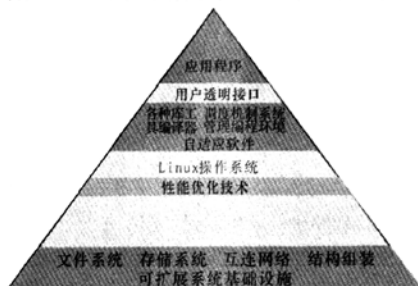


图 1 自适应超级计算

Cray 采用了自适应超级计算技术把标量计算、向量计算、多线程和可重构计算技术都整合到同一个高性能计算平台上去。多种类型的处理技术共享全局寻址存储。采用符合工业标准的 Linux 操作系统,用户可以将整合计算平台作为单一系统来使用。通过虚拟化技术对低层进行抽象,用户可以在抽象的系统架构之上,使用标准的并行编程语言来编写应用程序。当进行程序编译时,由编译器先对代码进行分析,确定哪一种类型的计算技术最适合这段代码,然后再对这段代码进行相应地编译。程序开始执行的时候,调度软件自动将代码配置到合适的节点上运行。自适应超级计算使得系统能自动适应用户代码,而不是让程序员改变代码来适应特定系统。这就意味着能够在同一个

高性能计算平台上支持更多不同领域、不同特点的高性能计算应用程序,用户不必再为了在能够某个特定系统更有效地运行而修改应用程序。

到 2010 年左右,Cascade 系统的原型机将把各种类型的计算技术紧密集成到单一系统中去,在每一个机柜都集成了标量、向量、多线程和硬件加速等计算技术,并提供自适应软件来使节点层的异构处理技术对用户透明。

### 2.2 资源管理——虚拟化集群系统

在资源管理方面,Ohio State 大学和 IBM Watson Research Center 的 Huang 和 Abali 等人提出一种虚拟化集群系统框架结构<sup>[8]</sup>。该框架结构采用模块化方法进行设计,通过多种优化技术提高虚拟化集群系统的性能。

#### 2.2.1 虚拟化集群系统框架结构

图 2 所示为虚拟化集群系统框架结构。

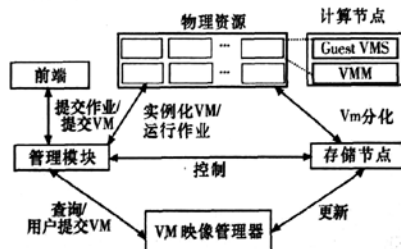


图 2 基于 VM 的虚拟化集群系统框架结构

虚拟化集群系统框架结构主要包括了以下五个主要模块:

(1)前端:前端是用户与虚拟集群系统交互的界面,用户可以在前端节点上定制系统环境,提交批处理的作业。

(2)物理资源:物理资源是通过高速互连网络连接起来的计算节点,由虚拟机管理器进行管理。为了获得最佳性能,一个物理节点上不会运行超过其上的 CPU 或核数目的虚拟机,以避免频繁地切换状态所带来的性能损失。

(3)管理模块:管理模块是整个框架的核心,其主要任务是根据提交作业的要求查找和调度物理资源,实例化 VM 映像以满足作业运行的要求。当获得足够多可用的空闲虚拟机时,管理模块提交并执行作业。

(4)VM 映像管理器:VM 映像管理器管理所有的 VM 映像信息。这些映像信息包含诸如 VM 核心版本、OS 类型、特殊的库等,被保存在 VM 映像数据库中。VM 映像管理器负责查找用户作业需要的 VM 环境,然后用于创建对应的 VM 映像。

(5)存储节点:存储节点负责存储 VM 映像和用户生成的各种数据。

#### 2.2.2 支持集群系统虚拟化的性能优化

在构建虚拟化集群系统时,Huang 等人采用了多种技术对虚拟化集群系统进行性能优化。

(1)减小虚拟化带来的额外性能开销。

虚拟化集群系统中通过以下两种技术来减小虚拟化带来的性能开销:

①VMM-bypass I/O 技术:该技术允许 VM 通过 DMA 操作直接访问底层网络设备。这种访问方式只产生非常小的 I/O 虚拟化开销,可以使得高性能计算应用程序获得接

近原始的通信性能。

②定制 OS 技术:支持高性能计算应用程序运行所需最少的操作系统服务。通过裁减不必要的内核模块和系统服务来定制 OS,最大限度降低系统额外开销。

(2)减小 VM 映像管理开销。

虚拟化集群系统通过三种方法优化 VM 映像开销:

①最小化 VM 映像:在虚拟化集群系统中,VM 映像中包含了定制的 OS 环境,该 OS 环境几乎只包含了 MPI 函数库、必要的系统工具集以及一些运行时环境。这种定制 VM 映像非常小,可以在网络上高效传输,并在其它节点上快速实例化。

②快速可扩展的 VM 映像分布:创建一个 VM 映像分布模块负责将 VM 映像广播到各个节点,并在各节点上实例化,比如通过基于二叉树结构的广播算法将映像广播到各个目标物理节点上。这种做法可加速 VM 映像的分布。

③缓存 VM 映像:在有足够的存储空间的前提下,可以在计算节点的本地存储器上缓存 VM 映像。如果 VM 在最后一次实例化后没有发生更新,就不需要传输新的 VM 到该计算节点,当发生更新时,可以仅传输更新部分。

## 2.3 系统容错——基于 Xen 的主动容错系统

随着高性能计算机系统规模的扩大,系统故障率大大增加,传统的容错技术难以满足高性能计算机可靠性的要求。针对上述问题,Nagarajan 和 Mueller 等人在 Xen 的基础上实现了一种主动容错机制,允许进程自动地从失效节点迁移到健康节点<sup>[9]</sup>。

该主动容错系统主要包括三种机制:容错机制、节点失效预测和负载均衡机制。

(1)基于 Xen 的容错机制。高效的容错机制必须能够在系统出错时花费最小的开销对计算任务进行重定位,使其能在不同的物理节点上运行。更重要的是,在计算任务迁移时不应该中断任务本身的执行。基于 Xen 半虚拟化技术,VM 可以在节点出错时进行动态迁移。通过预迁移方法,可以避免中断任务运行,使 VM 内的所有状态预先迁移到目标 VM。迁移过后,VM 可以从迁移前的状态继续向下执行。

(2)基于 IPMI 的节点失效预测。作为主动容错系统,必须具备预测某些未来事件的能力。通过采用智能平台管理接口(IPMI)可以实现系统运行状态的实时监控,预测未来的节点失效事件。当预测到某节点即将失效时,可以通知系统启动上述容错机制,主动将该节点上的 VM 迁移到健康节点上。

(3)基于 Ganglia 的负载均衡。VM 迁移之前需要进行目标节点的选择。该系统中采用 Ganglia 监控各节点的本地资源使用状态,并将监控数据发送给其它节点,使所有节点都可以获得整个集群系统的全局状态。VM 迁移时根据最新的全局状态选择目标节点。

## 2.4 虚拟化系统环境——VSE

在构建虚拟化系统环境方面,美国橡树岭国家实验室的 Engelmann C 等人提出一种虚拟系统环境(Virtual System Environment,简称 VSE)<sup>[10]</sup>。VSE 具有良好的虚拟环境隔离性,可用于为高性能计算应用程序的开发和部署提

供沙盒(Sandbox)环境,也可以支持系统软件的开发和部署。

(1)VSE 的系统架构。VSE 采用了系统级的虚拟机管理器(VMM)和 VM 配置机制,为高性能计算机系统软件和应用程序的开发和移植提供基础环境的保障。VSE 的系统架构如图 3 所示。

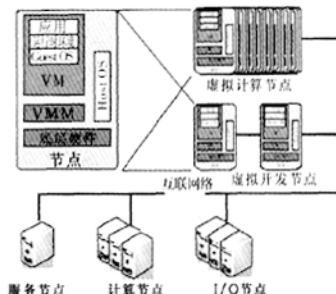


图 3 VSE 的系统架构

VSE 的核心部分是 VMM。VMM 实现对物理硬件资源的抽象,支持虚拟计算节点和虚拟开发节点的部署,以及虚拟节点上操作系统的运行。VMM 通过 VM 保护机制保证 VM 之间以及 VM 和宿主机操作系统(Host OS)之间相互隔离。

(2)VSE 的管理机制。

①系统管理。VSE 采用一组系统管理工具和配置文件进行系统管理。这些系统管理工具和配置文件被用于在 VSE 系统上创建、部署、清除和销毁 VM。

②配置管理。VSE 采用层次化的 VM 配置描述管理机制,对操作系统配置、软件环境配置和应用程序配置进行分层描述。配置管理机制支持用户权限的设置,以保证用户环境的隔离性和安全性。

(3)VSE 的应用。

①支持应用程序的开发和部署。VSE 可支持高性能计算应用程序早期原型的开发。通过在台式电脑上部署 VSE,可以构建与目标系统一致的小规模开发环境,开发的应用程序可直接移植目标系统上运行而不需要作任何修改。

②支持系统软件的开发和部署。VSE 提供一个虚拟的测试平台,通过构建虚拟硬件设备,以 VM 的形式提供沙盒测试环境,使开发人员不需要卸载原有操作系统软件就可以直接进行新系统的开发和测试。这种开发方式可以保证原有系统软件和其它 VM 不会受到影响。

## 2.5 并行编程环境——Charm++ 和 AMPI

在并行编程环境方面,Illinois 大学并行程序设计实验室(Parallel Programming Laboratory,简称 PPL)实现了一种基于虚拟化技术的并行编程模型,它可将程序分解成大量可并行处理的部分,这些可并行处理的部分构成大量虚拟实体,每个虚拟实体对应一个虚拟处理器。该模型使程序员不需要显式地考虑物理处理器的使用,他们在编程时只需要考虑各个虚拟实体之间的交互<sup>[11]</sup>。基于上述编程模型,Illinois 大学实现了 Charm++ 和 AMPI 编程语言。

(1)Charm++ 和 AMPI。Charm++ 基于 C++ 实现,其最基本的并行单元为一个 C++ 对象,称为 chare。

chare 中包含了可被其它该 chare 异步调用的方法。chare 对象之间通过传递消息进行通信。Charm++ 支持共享变量、优先级调度、chare 对象组等机制。通过这些机制实现对高性能计算机体系结构的抽象,使应用程序的编写更为方便。

除了 Charm++, Illinois PPL 还将虚拟化技术与 MPI 相结合,实现了一种自适应 MPI 模型——AMPI。在 AMPI 中,应用程序由大量虚拟的 MPI 进程组成,这些虚拟的 MPI 进程独立于处理器。AMPI 使用户可在更高的系统抽象层次上使用熟悉的消息编程接口来编写科学计算程序。

(2) 虚拟化技术对并行编程的影响。虚拟化技术应用于高性能计算机并行编程中可以带来以下几个方面的好处:

① 通过虚拟化技术可以提高编程模型的抽象层次,使用户可以更多地关注应用程序本身的逻辑而不是硬件结构的细节;

② 基于消息驱动的虚拟处理器在运行时可以自动获得消息/通信重叠的效果,如果再加上高效的预测机制,可以实现代码的运行时优化;

③ 虚拟处理器拥有动态迁移的能力,可以实现自适应动态负载均衡,更好地利用空闲计算资源。

### 3 虚拟化技术应用于高性能计算中存在的问题

从目前的研究可以看出,虚拟化技术尚未被广泛应用于高性能计算领域,其主要原因包括以下几个方面:

(1) 虚拟化带来的性能开销问题。传统的服务器虚拟化技术带来额外的系统性能开销。在一个虚拟化系统中,VMM 运行在最高特权级上,VM 和 Guest OS 作为 VMM 上的用户级程序运行。这就导致 Guest OS 运行过程中遇到特权级操作时,必须内陷到 VMM 中,由 VMM 执行。这种执行方式需进行上下文切换,且会导致更长的设备访问延迟,对于对系统性能敏感的高性能计算应用来说是难以接受的。因此,需要研究面向高性能计算机系统的虚拟化技术,针对应用对高性能的需求,进行 VMM 优化设计。

(2) 多 VMM 高效协同问题。面向高性能计算机系统的虚拟化技术不同于传统的服务器虚拟化技术。传统的服务器虚拟化只需要单个 VMM 对服务器底层硬件设备进行抽象。而在多维异构的高性能计算机系统中,每个节点都需要运行一个独立的 VMM,该 VMM 只对单个节点进行虚拟化。整个系统中运行了大量相互关联的 VMM。VMM 之间协同工作,构成统一的大规模系统级虚拟化环境。因此,需要研究高效的 VMM 协同机制,其中包括大量 VMM 之间的协同管理、协同部署、VM 跨物理节点的高效通信、VM 迁移以及系统级容错等技术问题。

(3) 大量 VM 的管理问题。为了支持高性能计算应用程序的运行,可能需要一次部署运行上千个乃至上万个的 VM。传统的服务器虚拟化技术只涉及少量 VM 的部署问题和运行时管理问题。如何支持大量 VM 的动态部署、根据应用需求快速分配所需的硬件资源、以较低的系统开销快速启动 VM、对大量 VM 进行运行时管理等,都是实现高

性能计算机系统虚拟化时需要解决的重要技术问题。

(4) 编程模型和软件环境的支持问题。传统的编程模型和支持应用开发和运行的软件环境都是直接针对非虚拟化系统设计的,而虚拟化技术对硬件系统进行了抽象,改变了用户所见的资源组织形态,使得传统的编程模型和软件环境无法适应用户对虚拟化系统的使用要求。因此,需要研究面向虚拟化高性能计算机系统的新型编程模型和针对虚拟化系统进行优化设计的配套软件环境,如并行编译器、链接器、调试优化工具、并行函数库等。

### 4 应用前景分析

目前,虚拟化技术已被广泛应用于台式电脑和服务器的应用,取得了很好的应用效果。相比之下,在高性能计算领域的应用还非常有限。随着高性能计算技术的发展和虚拟化技术研究的深入,两者之间的结合将会变得越来越紧密,虚拟化技术也将为高性能计算领域的诸多难题提供新的解决方法。

(1) 构建虚拟应用开发环境。高性能计算应用程序与运算环境关系密切,不同规模的运行环境下表现的行为也不尽相同。利用虚拟化技术,可以在小规模系统上构建大规模的虚拟应用开发环境,使应用程序可以在更为接近最终运行环境的系统环境下进行开发和优化,开发的应用程序可以被更快、更方便地移植到大规模系统上运行。

采用虚拟化技术,还可以对高性能计算应用程序的开发和调试提供特殊的支持。VMM 可以通过授权允许一个 VM 对其它 VM 的内存、中断和通信状态进行监控,有助于开发人员深入分析应用程序行为,便于高性能计算应用的调试。同样,这一机制也可以用于系统软件的开发与调试。

(2) 整合异构资源。随着应用需求和高性能计算技术的发展,高性能计算机体系结构正在逐步向着异构化的方向发展。如何高效管理和使用异构系统是高性能计算系统软件 and 应用程序开发人员面临的主要技术难题。虚拟化技术在整合异构资源方面有其固有的优势。利用虚拟化技术对底层异构硬件资源进行抽象和管理,可以有效地隐藏硬件平台的异构特性,为用户提供一个统一的系统环境,方便用户对异构系统的使用。

(3) 提供定制的 Appliance。不同的高性能计算应用程序需要不同的系统环境,如面向应用优化的操作系统环境、特定版本的编译器以及通信库等,这使得应用程序的部署费时费力,系统环境也难以做到最优化,系统效能得不到充分发挥。采用虚拟化技术有助于解决上述问题。虚拟化技术支持预先将定制的操作系统的、经过优化配置的应用运行环境 and 应用二进制代码一起打包成 VM 映像,这种映像被称为 Appliance。通过直接部署 Appliance,可以实现高性能计算应用程序的快速部署,并获得较好的运行时性能。

(4) 提高系统可靠性和容错性能。随着高性能计算机规模的不断扩大,部件间复杂性的不断提高,系统硬件故障率也不断增大。在虚拟化的系统中,由于节点上各 VM 之间相互隔离,软件错误如操作系统或应用程序故障只能直接影响到一个 VM,甚至硬件故障如 CPU、内存、设备的故障也只影响到它们所被指定到的 VM。当出现故障时,通

过 VM 迁移、快速重启 VM 等方法可以重新恢复甚至于不中断应用程序的运行。

(5)提高系统安全性。系统安全对于像数据中心这样的高性能计算应用来说非常重要,而 VM 之间的相互隔离和自测试能力为建立安全系统提供了一个平台。由于 VMM 只对底层硬件抽象和管理,提供一些简单的功能,相对于全功能的操作系统内核来说,更为可靠和安全。VMM 本身不会受到恶意代码的干扰,而且它所提供的 VM 之间的相互隔离和授权自测试功能也是完全可信的。VMM 可以对所加载的 VM 和加载到 VM 上的应用程序进行信用检查,可以授权一个 VM 对其他的 VM 状态进行审查,比如进行病毒扫描等。VMM 还可以持续监视 VM 之间的通信和 VM 的状态,以确保 VM 运行状态的正确性。

## 5 结束语

基于 Xen 等成熟的虚拟化技术,一些研究人员已经在高性能计算领域做了一些开拓性的探索,在集群系统虚拟化等方面开辟了一些成功的应用。到目前为止,虚拟化技术在高性能计算领域的应用还并不多见,其原因主要是高性能计算应用通常对系统性能有苛刻的要求,而虚拟化必然带来性能的损失,这对于许多高性能计算应用来说是不被容许的。因此,需要专门研究面向高性能计算机系统的高效率虚拟化技术,如针对高性能计算机系统的虚拟化模型、大规模虚拟化系统上的应用开发和部署机制、支持虚拟化系统的编程模型和配套软件环境等。随着虚拟化技术的不断发展,底层硬件越来越多地加入对虚拟化的支持,有效地降低了虚拟化的性能开销。而高性能计算机体系结构向多层次、多粒度的异构化方向发展的趋势也必将进一步推动高性能计算机虚拟化技术的发展。

### 参考文献:

- [1] Nanda S, Chiueh T. A Survey of Virtualization Technologies [R]. Technical Report TR-179, Stony Brook University, 2005.
- [2] Barham P, Dragovic B, et al. Xen and the Art of Virtualization [C]//Proc of the 19th ACM Symp on Operating Systems Principles, 2003; 64-177.
- [3] Uhlig R, Neiger G, et al. Intel Virtualization Technology[J]. IEEE Computer Society, 2005, 5: 48-56.
- [4] Abramson D, Jackson J, et al. Intel Virtualization Technology for Directed I/O[J]. Intel Technology Journal, 2006, 10 (3): 179-191.
- [5] Advanced Micro Devices. Amd64 Virtualization Codenamed "Pacifica" Technology Secure Virtual Machine Architecture Reference manual[M]. USA: AMD, 2005.
- [6] Snell A, Willard C G. Bridging the Capability Gap: Cray Pursues "Adaptive Supercomputing" Vision[R]. Cray White Paper, 2006.
- [7] Lazou C. Cray's Adaptive Supercomputing: A Paradigm Shift [EB/OL]. [2009-08-12]. <http://www.hpcwire.com/hpc/601369.html>.
- [8] Huang W, Liu J, Abali B, et al. A Case for High Performance Computing with Virtual Machines[C]//Proc of the 20th

ACM Int'l Conf on Supercomputing, 2006.

- [9] Nagarajan A B, Mueller F, Engelmann C, et al. Proactive Fault Tolerance for HPC with Xen Virtualization[C]//Proc of the 21st Annual Int'l Conf on Supercomputing, 2007.
- [10] Engelmann C, Scott S L, Ong H, et al. Configurable Virtualized System Environments for High Performance Computing [C]//Proc of the 1st Workshop on System-Level Virtualization for High Performance Computing, in Conjunction with the 2nd ACM SIGOPS European Conf on Computer Systems, 2007.
- [11] Kal'e I. V. The Virtualization Approach to Parallel Programming: Runtime Optimizations and the State of Art[C]//Proc of the Los Alamos Computer Science Institute Annual Symp, 2002.

(上接第 281 页)

## 6 结束语

与 Linpack 相比, HPCC 更全面一些。但是,作为 Linpack 的补充, HPCC 仍有一些局限:(1)HPCC 仍然是一个底层的测试标准,目前还缺乏有效的手段从底层指标准确地预测实际应用程序在系统上的性能;(2)HPCC 缺乏对 I/O 的测量;(3)Linpack 流行的原因之一是可以总结出一个分数来进行排序,HPCC 现在变成了多个分数,从技术的角度来看是更加精确了,但对公众和厂商的影响力反而可能下降。衡量高性能计算机时,除计算能力、内存访问、互连网络等能力外,一些指标同样值得注意,如:性能密度(单位空间里达到的运算性能)、性能节点比(单位节点所能达到的运算性能,也即以节点数为单位的性能密度)、性能能耗比(系统消耗单位电力所能达到的运算性能)、性能价格比、总节点数量、单位节点价格、峰值运算速率、内联总带宽、总存储空间、单位节点数据吞吐量等。

高性能计算机的性能评测在高性能计算领域中显得越来越重要,怎样更准确地评测其性能是一个需不断探索的问题。

### 参考文献:

- [1] Dongarra J, Luszczek P, Petitet A. The LINPACK Benchmark: Past, Present and Future[J]. Concurrency and Computation: Practice and Experience, 2003, 15(9): 803-823.
- [2] Luszczek P, Dongarra J, Kepner J. Design and Implementation of the HPC Challenge Benchmark Suite[J]. CTWatch Quarterly, 2006.
- [3] 肖文名, 李永生, 陈晓宇, 等. 高性能计算系统性能评测关键问题探讨[J]. 计算机系统应用, 2008(3): 115-118.
- [4] 郑邦民. 高性能计算机性能评测[P]. 计算机世界报, 2006-09.