

高生产率计算系统

孟 丹 张志宏 陈明宇

(中国科学院计算技术研究所国家智能计算机研究开发中心 北京 100080)
(md@ncic.ac.cn)

High Productivity Computing Systems

Meng Dan, Zhang Zhihong, and Chen Mingyu

(National Research Center for Intelligent Computing Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

Abstract The HPC world has been obsessively focused on simple performance metrics, such as peak floating point performance, or performance on the Linpack benchmark. However, two critical problems, the unbalanced architecture and the deficient programming support, have never obtained a good solution. The recent DARPA High-Productivity Computing Systems (HPCS) initiative has raised the issue of proper productivity measurements, in the context of high-performance computing. The background, objective, challenge, and main efforts toward the research of HPCS are presented here.

Key words HPCS; architecture; programming model; portability; robustness

摘 要 以往的高端计算领域将峰值运算速度作为衡量计算系统的最重要的指标. 然而体系结构的失衡和落后的编程支持成为困扰高性能计算技术发展的关键问题. 未来高端计算的发展方向应该着眼于提高系统的生产率, 强调系统的易用性和好用性. 介绍了 HPCS 的背景、目标、主要技术挑战以及主要的解决方案和关键技术.

关键词 高生产率计算系统; 体系结构; 编程模型; 可移植性; 稳定可靠性

中图法分类号 TP302

1 引 言

目前广泛采用的高性能计算相关技术主要产生于 20 世纪 80 年代末期. 尽管在过去的十几年里高性能计算取得了迅猛的发展, 但困扰其发展的两个关键问题一直没能得到根本解决. 从硬件体系结构来看, 根据摩尔定律, 微处理器的速度每 10 年增长 10 倍以上, 而常用存储器(DRAM)的访存时间却只减小一半. 因此, 相对于处理器来说, 访存延迟以每 10 年 5 倍的速度增长, 处理器与存储器以及通信性能上的差距不断增加, 造成了系统结构的失衡, 形成

了阻碍性能提升的“内存墙”; 在软件方面, 目前的高性能计算系统可编程性差, 为了编写一个有高性能的程序, 程序员在寻找并行性、编程实现和调配资源方面需要花费大量的时间. 因此, 新一代高端计算系统的发展方向不应该是以前那种单纯追求理论上的峰值速度, 而应该关注于如何开发一个设计平衡的计算系统, 提供方便好用、易于维护、稳定可靠的应用平台, 从而缩短应用的开发周期, 减小开发成本, 满足气象、生物和国防等尖端领域科学计算的需要, 尤其是满足未来量子科学计算的要求. 新一代的高端计算系统称为高生产率计算系统, 与以前强调峰值计算能力的高性能计算系统不同, 高生产率

计算系统强调平衡的系统结构、良好的可编程性和可移植性以及稳定可靠性,能够将强大的运算能力迅速转化为生产力。

本文围绕高生产率计算系统展开讨论,首先简单介绍美国国防研究项目局(Defense Advanced Research Projects Agency, DARPA)的 HPCS(High Productivity Computer System)研究计划,然后分析实现高生产率计算系统所面临的挑战和需解决的主要问题,以及目前针对这些挑战所提出的主要技术方案和解决方法,最后对全文归纳结论。

2 DARPA 高生产率计算系统研制计划

高生产率计算系统在性能、生产能力、可编程性和可移植性以及稳定可靠性方面有着远远高于目前技术能力的要求,单纯依靠商业领域内的技术进步很难在短期内达到目标。因此, DARPA 启动了 HPCS 科研计划^[1],希望藉此激发工业界和学术界的创造热情,产生革命性的关键技术和创新性的体系结构,实现高生产率计算系统。与目前的高性能计算系统相比,新一代高生产率计算系统不仅要在性能和计算效率提高 10~40 倍,达到 PetaFlop 的计算能力,而且在生产率、可移植性和稳定可靠性方面都要求巨大的提高。

DARPA 的 HPCS 计划分 3 个步骤实施:

(1) 从 2002 年起,用 1 年时间进行概念研究,提出一些革命性的思路 and 手段。

(2) 用 3 年时间开展关键性的研发活动和旨在降低风险的工程实践,探索上一阶段提出的技术要点和概念性方案的实现途径,实现完整系统的原型。

(3) 用 4 年的时间进行最后的商业性研制,在用户提出的一系列需求和可能采用的技术间进行平衡性设计,完善每一个细节,在 2010 年将完整的系统交付国防部选定的有关部门试用。

目前,该计划处于第 2 阶段的中期,第 1 阶段的 3 个方案,包括 IBM 的 PERCS^[2]、CRAY 的 Cascade^[3,4]和 Sun 的 HERO^[5,6],被 DARPA 选中。

3 HPCS 面临的技术挑战

为了实现高生产率的目标, HPCS 需要实现在以下 4 个方面的关键性进展。

3.1 系统的性能测试

长期以来,并行计算领域一直将峰值运算速度

作为衡量高性能计算系统性能的关键指标。而在国际上最权威的超级计算机 Top500 排行榜上也只是将计算机系统运行 Linpack 测试程序的结果作为评价计算机系统性能的惟一标准。事实上,分析人员为求解问题在这台计算机上所耗费的精力和时间比这台计算机的单纯计算能力重要得多。因此,对于强调高生产率的 HPCS 系统,其评价标准应该综合考虑开发时间和执行时间的因素。而如何实现一个标准的体系结构无关的 Benchmark 作为衡量高生产率计算系统的评价标准,成为一个巨大的挑战。正如 Linpack 测试对目前的高性能计算所起到的重大作用一样,一个公正有效的高生产率计算基准的出现也必将对高生产率计算系统的发展产生巨大的推动和深远的影响。

3.2 体系结构

失衡的体系结构是目前高性能计算系统普遍存在的问题。系统均衡意味着没有瓶颈出现。目前计算机系统的瓶颈是访存带宽。由于带宽与处理器的运算能力不匹配导致处理器在大部分时间等待数据而处于“饥饿”状态,使性能大打折扣。目前某些应用任务在当前的万亿次计算机上运行时其计算效率甚至低于 5%。

高生产率计算系统需要平衡的体系结构设计,需要高效率的带宽和解决数据传输的延迟问题,需要在处理器、存储和网络的设计和系统的体系结构上进行综合的考虑并提出整体的解决方案。

3.3 可编程性和可移植性

可编程性是指从提出问题到有效求解过程所花费的时间和精力,同样是衡量计算机系统生产率的重要指标。现有的系统缺乏可编程性,这迫使程序员不得不聪明到善于管理硬件资源。这种聪明称为 Tuning,是指花费精力编写适合特定系统的程序。针对特定系统的“Tuning”是脆弱的,增加了系统的维护费用。这种落后的编程模式往往要求那些需要高性能计算能力的人具有专家的水平,严重影响了高生产率的实现。高生产率计算系统要求具有良好的可编程性,系统应该提供开发者友好的开发环境和高级的开发语言,开发者可以比较轻松地完成编程工作,系统负责 Tuning 的工作。

可移植性不仅包括将程序从一种机器移植到另一种机器,也包括从万亿次量级(terascal)的系统移植到千万亿次量级(petascal)的系统,实现应用软件与硬件系统结构的无关性。可移植性确保程序员只需要关注求解问题本身,而不必分心于系统体系

结构的细节,如处理器类型和数量、内存组织和通信模型等。这种可移植性的提高要确保不会降低使用效率。

3.4 稳定和可靠性

稳定和可靠性包括硬件故障容忍、软件故障容忍、故障间的平均间隔、维修的难易程度、对恶意攻击的抵抗力和对操作失误的抵抗力。HPCS 要求综合运用传统的和革新的手段,提高计算系统的可靠性和稳定性,例如能修复逻辑差错、忍受软件故障和自动进行入侵监测等。

4 解决方案

4.1 性能测试

高端计算系统的发展离不开 Benchmark。与 HPCS 同样受 DARPA 资助的项目称为 HPCCh (HPC challenge benchmark)^[7],其目的是研究适用于 2010 年的高生产率计算系统的性能测试方法。HPCCh 收集了目前应用广泛的 7 个 Benchmark 测试方法:

HPL,一个 Linpack TPP Benchmark,通过求解线性方程组的形式测试计算系统的浮点运算能力;

DGEMM,通过双精度矩阵乘法测量计算系统的浮点运算能力;

STREAM,测试系统内存带宽;

PTRANS,通过处理器之间的同步通信测试系统网络的总体通信能力;

RandomAccess,测试内存的随机更新速度;

FFTF,通过一维离散傅里叶变换的形式测试系统的浮点运算能力;

Communication Bandwidth and Latency,通过大量的同步通信模式测试通信的带宽和延时。

HPCCh 考虑在进一步的工作中对 Benchmark 集进行扩展,并融合以上的各种测试形成一个更全面的 Benchmark 测试框架。

除了 HPCCh 的工作以外,另一些研究针对 HPCS 的特点提出了新的 Benchmark 方案。

文献[8]认为,Benchmark 应该能够真正反映用户的最终使用目标,由此出发提出了一个 PPB (purpose-based benchmark),PPB 的目标直接与用户的利益相关。例如,“准确预测美国 3 天内的天气状况”就可以作为一个 PPB 测试。当然,实际的测试应该有精确的定义,说明要做什么事,给出对任务的文字上和数学上的定义说明和具体的数字精度的要求。但是,PPB 并不指出怎样完成任务,Benchmark

的测试者可以自由地采用任何方式来完成。然而,不论采用何种技术,必须将该测试方法公开,以便其他的 Benchmark 测试者同样可以使用这种技术,否则该测试结果将被视为无效。

文献[9]认为,time-to-solution 是考察计算系统生产率的必要标准,应该平衡考虑开发时间和执行时间的重要性。以此为出发点,基于经济学上对生产率的定义以及 Utility Theory 理论,提出了一个测量高端计算系统生产率的框架模型。

文献[10]认为,应该用一台计算机的整个生命周期里的生产能力考察它的生产率,并且认为生产率与峰值运算速度、运算效率、可编程性、可移植性、可维护性、可靠性和可访问性等因素相关,并用一个概念性的方程式定量地表达了生产率与这些因素的关系。

4.2 体系结构和部件

高生产率计算系统需要平衡的体系结构。目前的系统体系结构失衡的原因主要是处理器的运行速度与带宽不匹配。因此,解决这一问题的关键是提高带宽,更加有效地利用带宽,以及隐藏数据访问延迟。

Sun 和 CRAY 针对 HPCS 分别提出的方案反映了工业界为解决这一问题在体系结构上的创新。

Sun 的 HPCS 方案称为 HERO。支持 HERO 体系结构最具创新性的技术是 Proximity Communication。该技术利用电容代替导线进行芯片间通信。它所提供的通信带宽影响了 HERO 设计的每一个领域。目前,通过导线和焊点互联的芯片间数据传输的极限是 50Gb/s,而 Sun 的实验室的研究显示到 2007 年 Proximity Communication 可以达到这个速度的 1600 倍。Proximity Interconnect 符合摩尔定律的发展趋势。在 2007 年之内,估计这项创新将使紧耦合的芯片组中的每片芯片达到 80Tb/s(10TB/s)的带宽。处理器、内存和其他类型的电路用 Proximity Communication 技术组合起来,构成 HERO 体系结构的基本构造模块——Sea of Memory。

图 1 显示了利用 Proximity Communication 技术进行芯片间通信的机制。

图 2 显示了由边长为 1cm 的芯片利用 Proximity Communication 技术构成的 Sea of Memory。图中展示了处理器、内存和 I/O 芯片的布局。10×10 个芯片与 9×9 个芯片交叉排放,共 181 片。处理器、内存和 I/O 芯片用重叠的传输芯片粘在一起。181 个芯片构成的模块边长为 20cm,按如

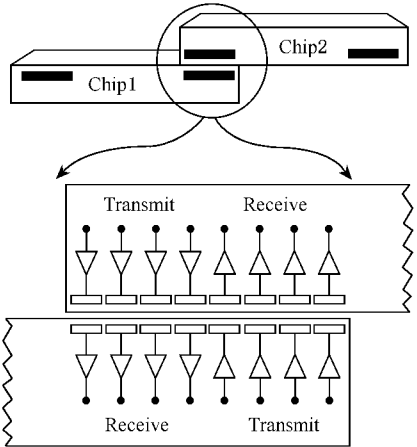
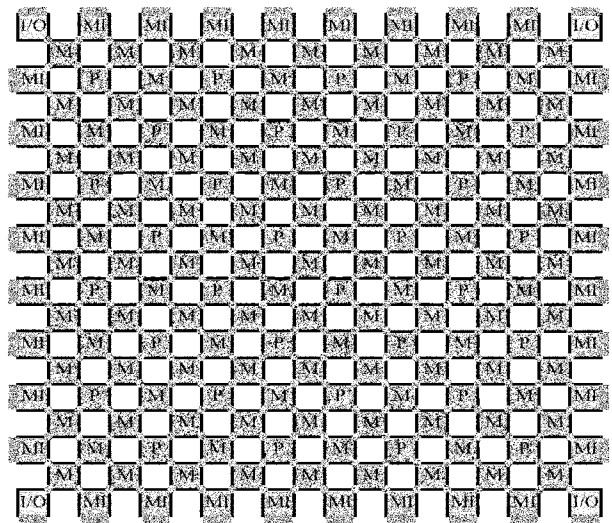


Fig. 1 Proximity Communication.

图1 Proximity 通信

下分布：

- (1) 32 个处理器均匀分布在中部；
- (2) 32 个模块接口芯片分布在周边,用于模块互连；
- (3) 113 个内存站(4 个内存芯片组合在一起,形成标记为 M 的一个站),共 452GB 内存；
- (4) 4 个外部 I/O 芯片.



Hero Module "Sea of Memory" —Actual Size

- Processor
- Module Interface
- Proximity Interconnect
- Memory
- Input/Output
- Optical Interconnect

Fig. 2 Sea of memory.

图2 Sea of memory

Sun 将这样的芯片集合称为一个 HERO 模块. 这个模块是 Sun 的体系结构中最小的建筑材料. 以 2007 年的处理器和内存技术推算,单个模块的处理速度可达到 2T/s 的峰值速度、1T/s 的保持速度、内存容量为 452GB. 尽管内存容量更大,但由于小的

物理尺寸和 Proximity Communication,HERO 模块的内存延迟要比目前的系统强得多. 为了解决 HERO 模块间的带宽问题,Sun 采用了光互联技术提供带宽保证.

Cray 的 HPCS 方案称为 Cascade. 为了使 Cascade 成为一个设计平衡的系统,Cray 认为带宽是系统中最重要也最昂贵的技术环节. 其设计方针是提供非常高的全局带宽,但这种带宽是可配置的,即可以随着系统的不同需求而改变. 在处理器设计上强调更加智能化的使用带宽,同时通过系统结构的总体设计减小系统对带宽的依赖性.

Cascade 中最基本的组成模块称为 Locale. 一个 Locale 由一个重量级处理器、一个路由器、许多 PIM (processor in memory,在一个芯片内包含多个处理器单元和多个内存单元的技术)轻量级处理器和 DRAM 存储池组成,其结构如图 3 所示：

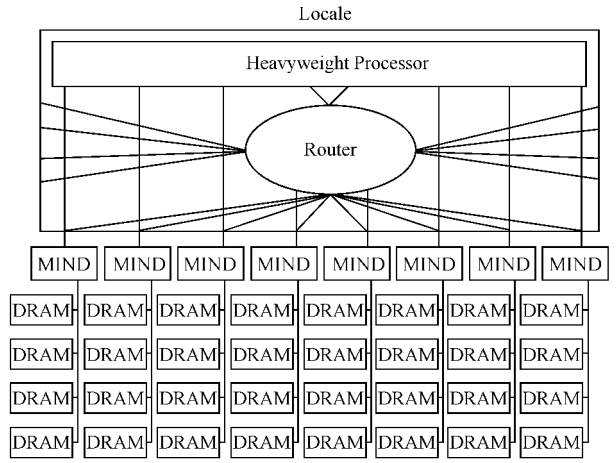


Fig. 3 The structure of a locale.

图3 Locale 结构

其中,重量级处理器结合了向量计算、流计算和多线程等领域内最先进的设计技术,执行高吞吐率的计算任务. 在重量级处理器上执行的线程称为重量级线程,具有较好的时间和空间局部性. 轻量级线程在 PIM 轻量级处理器上执行,这些线程的时间和空间局部性较差,在执行过程中,不是将数据向线程移动,而是把线程向靠近数据的方向迁移 (migrating). Cray 有可能采用 Caltech 和 JPL 研制的 MIND (memory intelligence and network device) 芯片作为 Locale 中使用的轻量级 PIM,这种芯片集成了计算单元、存储单元和网络功能,为轻量级线程的执行提供了很好的支持.

Cascade 采用 UMA + NUMA 的存储方式,通过低通信开销、细粒度数据访问、缓存远程数据和并发

性支持等多种手段实现延迟容忍和延迟隐藏。

Cascade 在一个电路板上集成了多个 Locale, 这些 Locale 通过板上的信号线连接。多个电路板通过光缆或电缆连接构成了一个 Chassis, 而 Chassis 间通过光纤网络连接构成一个可扩展的并行计算系统。图 4 显示了 Cascade 的整体结构。

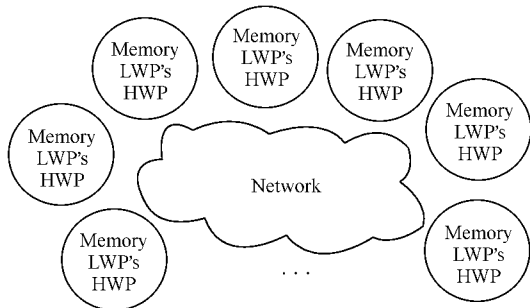


Fig. 4 The structure of cascade.

图 4 Cascade 整体结构

尽管没有公布整体的体系结构方案, IBM 针对 HPCS 目标也提出了一些关键部件的创新思想^[10]。IBM 的 CMP 技术通过在一个芯片内部紧密耦合了多个内核, 来提高系统的集成度和并行处理能力; “Smart Memory”项目研究如何对内存加入智能机制, 以便实现内存的预取、聚集/分发、访问记录、缓存和计算能力, 以缓解体系结构的失衡; “全局名字空间支持”通过硬件支持为高级语言提供全局名字空间的高效访问能力; “异步 SMP”项目研究通过更加松散的硬件一致性模型为软件模型提供支持; 基于目录的缓存一致性”研究基于目录的缓存机制, 与基于 snooping 的缓存一致性相比, 基于目录的缓存机制设计简单, 可扩展性更强。“网络聚集支持”项目通过下面的 3 种方法提高网络聚集支持: 处理器聚集支持、适配器聚集支持和交换机聚集支持。聚集支持有利于提高大规模系统上的应用性能。

4.3 可编程性和可移植性

现有的并行编程模式存在局限性, 不易使用, 许多优化的工作由开发者负担。例如, MPI 利用本地存储和消息传递机制实现并行化编程, 其并行化、局部性和全局视图完全由编程者来管理, 通信、同步、一致性操作等机制都是在很低的抽象级别来完成, 限制了编译器优化的机会。

可编程性不仅意味着 HPCS 要支持目前人们习惯使用的编程语言(如 C/C++、Java、Fortran 等)和并行编程语言 OpenMP、MPI, 还要提供更高抽象级别的编程语言, 使机器自动完成性能优化的工作, 人

们只需要关注所要完成的任务本身。

Sun 的 HERO 方案从两方面入手解决这一问题: ①使内存访问速度有巨大提高, 程序员根本不必考虑内存局部性和内存重用问题; ②实现可移植中间语言 PIL(portable intermediate language)和动态运行环境 RTE(run-time environment), 使剩余的“Tuning”负担从程序员身上转移给系统软件。PIL 类似 Java 的 class 文件, 在编程模型中处于编译器和 RTE 之间, 保证了程序的可移植性, 并代替用户将 Tuning 的任务交给 RTE 完成。PIL 也将提供对目前应用广泛的成熟的语言支持。RTE 在理论上类似于 JVM, 它不仅提供了程序的执行环境, 而且实现动态编译和优化, 通过底层微内核的支持管理进程/线程的调度和资源分配, 对程序性能进行自动调节, 并提供安全和故障恢复机制实现高可靠保障。HERO 还提出了 Literate Programming 的编程模型, 利用该编程模型编写的程序是一个集文本、代码和数学公式于一体的文档, 为科学计算提供了有力的支持。

Cray 的 Cascade 方案也提供更高级别的编程方法、强大的编译器、调试器以及性能调节工具和运行时支持来提高系统的可编程性。Cascade 的编译器在系统中处于十分重要的位置。它不仅具有传统编译器的功能, 而且直接控制系统的数据缓存和线程迁移。编译器决定什么时候数据可以被安全地缓存、哪些数据可以暂时私有化和暂时量化, 编译器控制着硬件, 决定何时缓存、何时回写、何时使缓存失效。强大的编译器支持可以在很大程度上减轻程序员的负担。

IBM 在 PERCS 方案中提供新的高级编程语言 CHAPEL 和 X10^[11], 这些编程语言向用户提供通用的并行编程模型, 以及简单易用的并发模式和存储访问模式, 使用户可以更轻松地编写并程序, 编译器和运行环境的支持确保了语言的平台无关性以及性能的自动调节。此外, PERCS 还将针对特定应用领域提供专用的编程语言。集成的开发环境、可视化工具和基于模块化的开发工具也将为用户的开发提供有力支持。

可编程性是提高计算机系统生产能力的关键环节。目前存在的并行编程模式不能令人满意, 而如何实现串行程序的并行多年来也一直是计算机科学中的“圣杯”。从以上的 3 种方案来看, 设计新的高级语言提高算法的抽象级别、强大的编译器和运行时环境支持是公认的解决办法。

可移植性要求程序与硬件平台的无关性,要求具有“一次编译,处处运行”的编程模式支持。Sun 提出的可移植中间语言 PIL 对可移植性提供了很好的支持。PIL 是一种系统间可移植的中间二进制形式代码。它不仅包含目标文件信息,还包含算法程序的注释和分支预测。PIL 并不包含与特定体系结构相关的信息,所有这些信息由 RTE 处理。PIL 所效仿的两位前辈是 Acrobat's PDF 和 Sun's Java。它们都定义了一个公开的标准中间形式,任何人都可以对其进行翻译。不同的平台提供自己的“阅读器”或者“虚拟机”。这样,Acrobat 和 Java 都可以到处执行。

4.4 稳定可靠性

稳定可靠性的实现需要从底层硬件结构到操作系统和管理工具的全面支持。从底层看,硬件应该提供冗余和故障自动屏蔽和分离机制,操作系统提供自动治愈的机能,管理软件实现全部资源的可视化管理,检查点支持可以实现故障恢复。

在 HERO 方案中,广泛使用了冗余技术确保稳定和可靠性。能源、时钟、散热设备都是冗余的。Sun 可以在每个 HERO 模块使用冗余的服务处理器来替换一个内存芯片。服务处理器用“心跳方法”连续监测模块,动态管理失常的部件。服务处理器记录软故障以便查明薄弱环节,以免这些环节成为系统使用的障碍。RTE 使用多种方法加强系统的可用性。在每次执行前,RTE 检查 PIL 的正确性。通过使用并行冗余代替并行执行,可以以性能减弱的代价换取可靠性增长。HERO 提供这种选择。RTE 在用于高可靠性要求的环境时,还可以提供双重、同步的操作,或者三重冗余、硬件投票的工作模式。

PERCS 延续了 IBM 近年来提出的自主计算 (autonomous) 的思想,即系统应具有自动配置 (self-configuring)、自我修复 (self-healing)、自我优化 (self-optimizing) 和自我保护 (self-protecting) 的功能。通过具有 Hot Swapping 技术的 K42 操作系统支持,实现系统的动态重配置;通过连续的性能监测和自动调整,实现自我优化;通过对用户透明的检查点和具有自我恢复功能的中间件达到系统的稳定、可靠目标。

5 结 论

本文介绍了美国国防部先进研究项目局提出的下一代高生产率计算系统研制计划,分析了实现

HPCS 所面临的主要挑战和需解决的关键技术,列举和分析了针对每一个环节所提出的主要解决思路和方案。

高生产率计算系统代表了未来高端计算的发展方向,其关键技术涉及到先进的系统结构设计、高级编程语言、可靠性研究等高端计算的热点问题,为国内高性能计算机系统研制提供了很好的借鉴。

参 考 文 献

- 1 DARPA. High Productivity Computing Systems (HPCS) Program. <http://www.darpa.mil/ipto/Programs/hpcs/index.htm>, 2002-04
- 2 Mootaz Elnozahy. PERCS: IBM effort in HPCS. <http://www.ncsc.org/casc/meetings/vision-public.pdf>, 2003
- 3 Cray Inc. The cascade project. <http://www.cray.com/cascade/>, 2002
- 4 Per Nyberg. The Cray rainier system: Integrated scalar/vector computing. <http://www.ecmwf.int/newsevents/meetings/workshops/2004/high-performance-computing-11th/pdf/Per-Nyberg.pdf>, 2004
- 5 John L. Gustafson. Sun's HPCS approach: Hero. <http://www.ncsc.org/casc/meetings/CASC2.pdf>, 2003-08
- 6 Jim Mitchell. Sun high productivity computing systems research program. <http://research.sun.com/sunlabsday/docs/talks/1.01-Mitchell.pdf>, 2004
- 7 HPCC. HPC challenge benchmark. <http://icl.cs.utk.edu/hpcc/>, 2004
- 8 J. Gustafson. Purpose-based benchmarks. <http://www.highproductivity.org/IJHPCA/8f-Gustafson-Productivity.pdf>, 2004
- 9 Thomas Sterling. Productivity metrics and models for high performance computing. <http://www.highproductivity.org/IJHPCA/4f-Sterling-ProductivityPaperv3.pdf>, 2004
- 10 John D. McCalpin. Issues in the architecture of large scale systems. <http://www.spscicomp.org/ScicomP10/Presentations/McCalpin.pdf>, 2004-09
- 11 Kemal Ebcioglu, Vijay Saraswat, Vivek Sarkar. X10: New opportunities for compiler-driven performance via a new programming model. <http://www.cs.ualberta.ca/~amaral/cascon/CDP04/slides/sarkar.pdf>, 2004-10



Men Dan, born in 1965. Supervisor in ICT, CAS. His research interests include the architecture of HPC, Operating System, etc.

孟丹, 1965 年生, 研究员, 博士生导师, 主要研究方向为高性能计算机体系结构、操作系统等。



Zhang Zhihong , born in 1977. Ph. D. candidate in ICT , CAS. His research interests include cluster management and middleware , etc.
张志宏 ,1977 年生 ,博士研究生 ,主要研究方向为机群管理和中间件技术.



Chen Mingyu , born in 1972. Ph. D. , associate professor. His main research interests include architecture of high performance computer , system software , information security.
陈明宇 ,1972 年生 ,博士 ,副研究员 ,主要研究方向为高性能计算机体系结构、系统软件、信息安全等.

Research Background

The work in this paper is supported by the National High-Tech R & D Program(863 Program) of China. High-performance computers are still in the limitation list of strategic technologies from western countries. In this way , it concerns the national security and plays a key role in the economic and social development. The National Research Center for Intelligent Computing Systems (NCIC) , who aims at the research and development of computer systems with super performance , attaches importance to the rising of high productivity systems. This paper gives an introduction and analysis on the development of high productivity systems.

第五届全国虚拟现实与可视化学术会议(CCVRV '05)
征文通知

由中国计算机学会虚拟现实与可视化技术专业委员会和中国图像图形学会虚拟现实与可视化技术专业委员会主办、北京航空航天大学承办的第五届全国虚拟现实与可视化技术及应用学术会议将于 2005 年 9 月 24~25 日在北京举行. 本次会议将集聚国内从事虚拟现实与可视化技术的研究人员和工程技术人员 , 广泛开展学术交流、研究发展战略、推动成果转化、共同促进虚拟现实与可视化技术的发展与应用.

本次大会录用的学术论文将在我国计算机领域的权威期刊《计算机研究与发展》(增刊)发表. 会议将邀请国内外著名专家作专题报告 , 同时将举办科研成果和最新产品展示会 , 为各研究开发单位及有关厂商展示自己的成果、产品提供场所. 欢迎大家积极投稿.

征文范围(包括但不限于)

建模技术	动画技术	可视化技术	多媒体技术	人机交互技术
虚拟制造	仿真技术	分布式系统	空间化声音	模式识别应用
图形平台	网络技术	遥控操作技术	VRML 技术	逼真图形图像技术
增强现实	协同操作	数字博物馆	网络游戏	图像绘制技术
可视化地理信息系统		基于图像的视景生成技术		虚拟现实与可视化应用系统

征文要求

- (1) 论文未被其它会议、期刊录用或发表 ;
- (2) 要求接受电子投稿(同时提交 Word 与 Pdf 格式文件);
- (3) 论文包含 : 题目、中英文摘要、正文、参考文献等 ;
- (4) 正式论文格式见论文录用通知 ;
- (5) 投稿者请在论文最后务必写清姓名、单位、通信地址、电话及 E-mail 地址.

重要日期

征文截止日期 : 2005 年 5 月 1 日(收到日期)
录用通知日期 : 2005 年 5 月 25 日(发出日期)

来稿联系方式

联系单位 : 北京航空航天大学 6863 信箱
联系地址 : 北京市海淀区学院路 37 号
电 话 : (010) 82339332 82317644 82317610
邮政编码 : 100083
联 系 人 : 周忠、吴威、陈小武
电子邮件 : ccvrv05@vrlab. buaa. edu. cn

会议网站

http ://vrlab. buaa. edu. cn
欢迎上网查询大会各项文件和最新通知.