

高性能计算机的系统分区技术研究^{*}

卢 凯 胡湘华

(国防科大计算机学院 长沙410073)

摘 要 随着 SMP 和 ccNUMA 等高性能计算机系统的规模越来越大,为了满足用户应用动态负载平衡的需求,并同时提高系统利用率和高可用性,越来越多的高性能计算机系统采用了分区技术。本文在分析对比了各种分区技术的基础上,深入介绍了系统分区的关键技术:支持分区技术的硬件设计技术、Firmware 监控层技术和操作系统技术等关键技术,并展望了其面临的问题和技术发展趋势。

关键词 SMP, CcNUMA, 系统分区, 操作系统

Technology of System Partition on High Performance Computer System

LU Kai HU Xiang-Hua

(School of Computer, National University of Defence Technology, Changsha 410073)

Abstract High performance computer, such as SMP and ccNUMA, are becoming larger and larger. To fill with users' requirement of high system available and dynamic workload balance, Partition technology is becoming more and more important. In this paper, all kinds of partition tech. are compared first. And then, we also introduce the key technologies including hardware design, firmware hypervisor and operating system to support the partition technology. The future and problems are also showed in the last of this paper.

Keywords SMP, CcNUMA partition, Operating system

1 引言

随着计算机技术的飞速发展,高性能计算机系统的规模正不断地扩大。目前商用 SMP 系统规模已多达64个处理器,而采用交叉开关的 CMP 系统已推出支持128处理器的 SMP 系统。ccNUMA 系统基于分布式存储器体系结构,通过高速缓存一致性(Cache Coherent)技术来解决存储一致性问题,较 SMP 系统具有更好的系统可扩展性和可配置性。目前 SGI O3000系列高性能计算机系统已可支持512个处理器,并可进一步支持更大规模的系统。

不同类型的大规模并行科学计算应用,如计算密集型、通信密集型、I/O 密集型等,对计算机资源有不同的需求。即使在一个应用的不同阶段,其特性也各不相同。传统固定规模的计算机体系结构由于 CPU 数目、内存大小和通信能力固定,无法满足各种用户应用需求的动态变化。所以,根据用户需求动态配置计算机系统,支持应用动态负载平衡的系统可重构技术是未来高性能计算机系统的重要发展方向之一。此外,从应用和程序开发以及操作系统实际性能的角度考虑,用户尚无法真正充分利用规模如此庞大的系统资源,目前的操作系统也无法有效管理64个处理器的 SMP 和 ccNUMA 系统。规模庞大的计算机系统在可靠性等方面也存在巨大的问题。为了有效提高系统资源划分、多个环境、更高的可用性和可扩展性,“分区技术”是一种重要的技术途径。

分区技术最初出现在大型机(MainFrame)上,它支持将大规模计算机系统的 CPU、内存、I/O 等资源合理地进行分区和调配,不同分区内可以执行不同的任务,最大限度地挖掘

了服务器的性能。一个大规模 ccNUMA 或 SMP 系统可以当作几个独立系统来使用,提高了硬件利用率。并且,在任何某个分区发生故障时,都不会影响其他分区的运行,不同分区的应用之间保持独立性。

本文系统地对现有“分区技术”进行分类和对比,在此基础上重点介绍了分区技术的关键点。本文首先介绍了现有的主要分区技术,其次重点介绍了分区技术的关键技术,最后介绍了分区技术目前面临的技术难点和未来的发展趋势。

2 分区技术综述

目前计算机系统分区所采用的技术方式有多种类型,根据不同分类方式可以分为:根据实现技术手段可分为硬件分区和软件分区;根据分区边界可分为物理分区和逻辑分区;根据调整时机可分为固定分区和动态分区等。

硬件分区技术一般在系统加电前通过修改系统的硬件配置,如修改系统跳线等,来改变系统规模的分区方式。而软件分区技术则是系统加电后,通过 Firmware 或专用的资源配置软件来实现的分区方式。软件分区方式较硬件分区方式有更好的灵活性和便容性,是目前所广泛采用的分区方式。

物理分区技术是以系统模块边界为单位来进行资源整合的。而逻辑分区技术则可根据实际需求,在所有的资源范围内任意地选择资源来构成不同区域。显然,逻辑分区的灵活性较物理分区方式大。

动态分区技术可在系统运行过程中修改分区资源,而无需重新启动计算机。固定分区技术则必须在系统重启后方可生效。动态分区技术对计算机系统的高可用和动态负载平衡

^{*} 本论文受863课题2003AA1Z2060操作系统新技术研究支持。卢 凯 博士,副教授,主要研究方向是系统软件和大规模并行处理技术。胡湘华 博士研究生,主要研究方向是系统软件和大规模并行处理技术。

有至关重要的意义。

根据上述分区方式,我们通常将现有分区技术归为物理分区技术、逻辑分区技术和动态逻辑分区技术三类。

2.1 物理分区技术

物理分区(PPAR)技术基于计算机系统的物理结构模块来划分区域。目前采用该分区技术的系统主要有 Sun 的 Enterprise 10000 系统、惠普的 Superdome 系统和康柏的 AlphaServer GS 系列系统等。

Sun 的 Enterprise 10000 系统以一个系统板分区单位,全系统最多可分为 16 个物理分区,每个物理分区包含一个或者多个系统板以及系统板上的全部资源。给现有物理分区增加系统板的时候,不论是否需要这些资源,系统板上所有可用资源都会集成到该物理分区中。Enterprise 10000 服务器可在不同的物理分区中运行不同的操作系统,运行在各个物理分区中的应用程序完全隔离,相互之间不会干扰。惠普的 Superdome 系统与 Enterprise 10000 系统相似,它以 cell 为单位(4 路 SMP 的系统板),最多可支持 16 个物理分区。这些分区称为 nPartition。

2.2 逻辑分区技术

逻辑分区(LPAR)技术不必按照系统的硬件结构分组,可从可用系统资源所组成的全部资源池中选择组件,具有更多的灵活性和自由度。LPAR 在单一的内存空间中配置和划分各逻辑分区。所以,所有操作系统映像都在同一内存映射中运行,但通过硬件或软件的地址访问等资源访问控制机制相互保护和隔离。LPAR 技术较 PPAR 技术可以实现更好的资源间隔:一方面系统资源得到了最大化的利用;另一方面则把不必要的资源重新分配最小化了。逻辑分区技术是目前服务器系统所普遍采用的分区技术。LPAR 的实现一般都基于虚拟机技术构造。

惠普的 vPartitioning 技术是典型的逻辑分区技术。它在计算机系统的 Firmware 层以上,通过软件实现的 vPartition 监控层来控制用户应用和操作系统对系统资源的访问。每个逻辑分区资源独享,均可独立启动和 reboot,并运行独立的操作系统映像。用户可借助 Process Resource Manager (PRM) 或 HP-UX Workload Manage (WLM) 等工具将一个物理分区划分为多个逻辑分区,进行资源的管理和分配。

IBM 的 LPARs 是较 HP 的 vPartition 更具灵活度的分区方式:其物理资源可以不受物理系统板的限制,处理器、内存和 I/O 插槽无论其物理位置都可任意分配给任何分区。LARP 主要由硬件管理控制台(HMC)、系统管理程序(Hypervisor)和 AIX 5L 操作系统三部分协同实现,其结构如图 2 所示。

•HMC 负责管理 LPARs,提供配置和创建、启动和停止、以及相应的控制和显示功能。

•AIX 5L 5.1 同传统 UNIX 操作系统有所不同。为了支持 LPARs, AIX 操作系统在以下方面进行了修改:1)通过调用 Hypervisor 层的功能来访问存储器和外设,而不是直接访问硬件和地址映射设备;2)不存在物理控制台,而是由固件为 AIX 提供虚拟 tty 作为标准的 tty 使用;3)与硬件平台相关的操作被局限在 LPARs 内。

•Hypervisor 是 LPARs 技术的关键,它以固件形式提供基于微码的分区隔离手段,在 LPARs 系统中提供分区控制和仲裁的功能,从处理器、存储器和 IO 设备级别上实现分区间的隔离。一些对硬件的操作指令只能由 Hypervisor 在处理

器的 Hypervisor 模式下执行,从而防止了执行非 Hypervisor 代码对其他分区的影响。

LPAR 技术较 PPAR 技术具有更好的灵活性,但由于通过虚拟机技术实现,故在系统性能上有所损失,一般在 10% 左右。

2.3 动态逻辑分区技术

动态逻辑分区技术(DLPAR)是在逻辑分区技术的基础上,支持用户在系统运行过程中改变分区的资源配置的分区技术。DLPR 技术可以灵活地根据用户应用的需求增加、删除处理器和 IO 设备,并可将故障设备替换出去,对系统的动态负载平衡和高可用性提供了更好的支持。

IBM 在 2002 年基于 LPARs 技术推出了动态逻辑分区技术(Dynamic LPARs),同操作系统配合,支持对分区资源的动态调整:能够在无须重启操作系统的情况下对分区的 CPU、内存和 IO 设备进行随意的添加和删除。但该技术要求与硬件系统相配合,对硬件设计有特殊要求,如热插拔的处理器、内存和 IO 设备等。Cellular Disco 是一个可同时运行多个操作系统映像的虚拟机监视器(monitor),由于它向操作系统提供了同底层计算机硬件系统的操作接口,故传统多处理器操作系统并不感知其的存在,故也不必作任何的修改。Cellular Disco 采用 VCPU 迁移的办法实现分区的动态变化:VCPU 系统支持 VCPU 在结点内、分区内和分区间的迁移。

动态分区技术由于其众多的技术优势,是目前计算机分区技术发展的主流。

3 关键技术

支持动态负载均衡和高可用的动态分区技术需要计算机系统在设计、系统固件设计和实现、操作系统,以及应用软件等方面多方配合才能实现。

3.1 硬件设计

目前的动态分区技术主要针对 SMP 体系结构和 ccNUMA 体系结构设计。SMP 系统中所有处理器对存储器和外设的访问具有相同的访问特性,故通过分区技术划分逻辑区域后,各分区均为一个小的 SMP 系统。ccNUMA 体系结构虽然是基于 NUMA 结构实现,但由于采用 Cache 一致性技术,如果不考虑通信和访存的延迟差异性,用户在操作系统层所看到的与 SMP 系统类似。

为了在 ccNUMA 体系结构中支持动态逻辑分区技术,就必须支持任何处理器对任何资源的全局访问,即意味着采用同样的方式,如物理地址、端口地址或内存映射地址可以访问到位于不同结点的存储资源、I/O 资源,外设中断也可动态路由到不同的处理器。目前的 ccNUMA 系统一般都支持全局的存储器访问,但对全局 IO 设备的访问支持目前还只有 SGI 的 O3000 系统。

同时,为了提供高可用服务,计算机硬件设计还必须支持热插拔技术。热插拔包括处理器热插拔、存储器热插拔和外设热插拔。目前较成熟的是外设热插拔技术,CompactPCI 总线技术标准提供了全套热插拔支持。处理器和存储器的热插拔支持一般由各个计算机厂家独立提供,目前全面支持处理器、存储器和外设热插拔的计算机系统还很少。

3.2 Firmware 监控层

Firmware 是提供动态逻辑分区技术的关键。动态逻辑分区技术一般基于虚拟机技术构造。从 IBM 的 LPARs 和 Disco 的结构图(图 1)可以看出,支持动态分区的虚拟机一般以软件

方式实现在操作系统之下,计算机硬件以上,即 Firmware 层中。在逻辑分区系统中,Firmware 层提供了分区控制和仲裁的功能,在处理器、存储器和 I/O 设备级别上实现逻辑分区之间的隔离。操作系统对硬件资源的访问是通过 Firmware 监控层进行的,而且一些低层的处理器指令只能由 Firmware 在处理器的高特权模式下执行,从而防止了执行操作系统代码和用户应用对其他分区的影响。

Firmware 监控层代码存储在系统初始化时被载入到内存中(一般在地址的最低端),负责分区环境的创建,是唯一可以访问特殊的处理器寄存器和地址转换表项的代码。各逻辑分区内部的程序不能访问 Firmware 监控层代码和数据,而只能调用 Firmware 监控层所提供的服务接口。

Firmware 提供的监控和管理功能主要通过通过对存储器地址访问的控制、IO 设备访问的控制和 DMA 操作的控制来实现。Firmware 监控层主要提供了以下四种地址转换和控制功能:

实模式寻址 操作系统的内核启动代码、固定的内核数据结构以及中断处理等都要求使用物理地址方式访问内存空间。在逻辑分区环境中,所有的逻辑分区均使用相同的物理地址加载操作系统和进行中断处理,但在实际使用中,每个逻辑分区均使用独立的地址空间(从0开始),互不共享。这样就给不同逻辑分区使用相同的物理地址带来了问题:要访问的物理地址不在本分区内。所以,Firmware 监控层就必须支持实模式寻址的重新映射功能。

实模式寻址的重新映射功能需要计算机系统硬件设计的配合。在启动逻辑分区时,Firmware 监控层为每个逻辑分区分配一个单独的实模式地址偏移量和区间长度,并把这两个值填写到该逻辑分区内的每个处理器的特定寄存器。地址偏移量的作用是把实地址映射到分配给该逻辑分区的物理内存区间。当逻辑分区的程序工作在实模式下时,硬件逻辑自动地为每个实地址加上地址偏移量值构成新的物理访问地址。这样,在逻辑分区看来,其所使用的物理地址都是从0开始的。但实际上,物理地址被硬件逻辑透明地转换到了该逻辑分区所属的地址空间中。硬件逻辑将同时保证软件的地址越界访问将导致寻址异常中断。

虚地址模式访问 虚地址模式访问通过页表来实现虚拟地址和物理地址的转换。页表保存在系统内存中。在非逻辑分区系统,操作系统通过页表的物理地址来访问、创建和维护页表。在使用逻辑分区的系统中,页表被放置在特定的预留内存区域内,并且只能由运行在高优先级的 Firmware 监控层访问,操作系统无法直接访问页表。操作系统对页表的操作必须通过 Firmware 监控层服务调用来执行,由 Firmware 监控层代表操作系统对页表项执行创建、修改和删除操作,实现页到逻辑分区内物理内存区域(又称为逻辑地址区域)的映射。

I/O 地址空间访问 与逻辑分区的虚地址访问机制类似,操作系统通过 Firmware 监控层调用将 PCI 设备适配器上的物理寄存器映射到系统的虚拟空间中。PCI 设备必须预先被分配到指定的逻辑分区上,对逻辑分区外的 I/O 空间的地址映射是被禁止的。

DMA 寻址模式 PCI 设备的 DMA 操作完成数据在 I/O 设备和系统内存之间的数据传送。DMA 基于类似于页表的地址重定位数据结构,由 PCI 北桥设备负责把 I/O 设备产生的地址转换成物理内存地址。用于地址转换的数据结构表也是保存在物理内存中,并且只能由 Firmware 监控层访问。

逻辑分区通过 Firmware 监控层为分配给本分区的 PCI 设备创建、修改和删除转换项,桥设备转换后的 DMA 地址落在本逻辑分区的物理内存空间内。

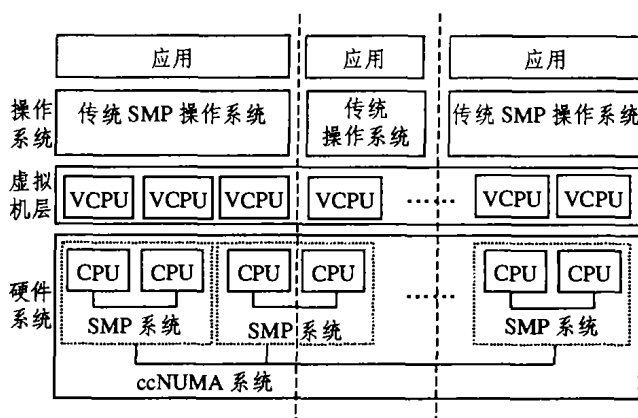


图1 DISCO 结构

3.3 操作系统

逻辑分区对操作系统几乎是透明的。在逻辑分区环境中,操作系统与 Firmware 监控层的交互如图2所示。首先,在操作系统引导过程中,Firmware 监控层负责报告分配给该逻辑分区的设备列表;其次,在运行过程中,Firmware 监控层代替操作系统完成虚存管理中的地址转换操作。

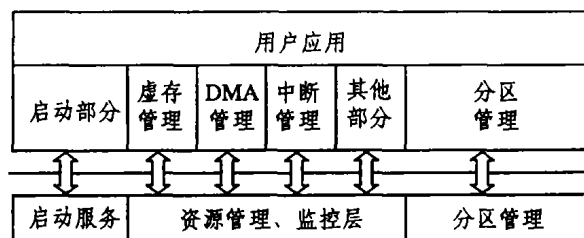


图2 操作系统同 Firmware 监控层的关系

为了配合 Firmware 监控层实现逻辑分区,操作系统在存储管理、DMA 操作、中断处理、系统调用、设备管理等方面同运行在一般服务器上的传统操作系统有所不同:

1) 运行在非最高特权层,通过调用最高特权层的 Firmware 监控层来访问内存和设备,而不是直接访问硬件和地址映射设备。

2) 由于一些设备是由 Firmware 监控层来模拟实现的,如 NVRAM、时钟等,故对这些设备的直接访问仅限于调试或其他特殊用途。

3) 与硬件平台相关的操作被局限在逻辑分区内部。操作系统的中断处理、虚存管理、DMA 空间管理等均通过 Firmware 监控层提供的功能函数实现,而不是传统操作系统中的直接汇编编程。

4) I/O 操作是通过调用 Firmware 监控层的操作功能实现的。原来 inb()、outb() 等函数对 I/O 端口的直接赋值操作都不允许。IO 操作都必须通过运行在高特权中的 Firmware 函数实现。同样,对 PCI 设备通过 mmap 映射虚存空间的地址也不可以直接通过访问内存的方式来访问,而必须通过 memcpy 或其他的访存函数通过调用 Firmware 监控层的功能才能实现。

5) 为了支持动态逻辑分区的计算资源切换功能,操作系统必须支持对处理器、存储器和 IO 设备等的热插拔功能。操

(下转第204页)

$$\forall i |dC_i(t)/dt - 1| < \alpha \quad (\alpha < 1)$$

$$\forall i \forall j |C_i(t) - C_j(t)| < \beta \quad (\beta < 1)$$

其时钟同步方法为:

(1) 对每个 i , 如果 P_i 在物理时间 t 时没有收到消息。则 C_i 在 t 可微并且 $dC_i(t)/dt > 0$;

(2) 如果 P_i 在物理时间 t 发送一个消息 m , 则 m 包含了 $C_i(t)$ 。

(3) 当在时间 t 收到消息 (m, C_j) 时, 进程 P_i 设置 C_i 为 $\max(C_i(t-0), C_j + \mu)$, 其中 μ 是预先定义的从一个进程发送消息 m 到另一个进程的最小延迟。

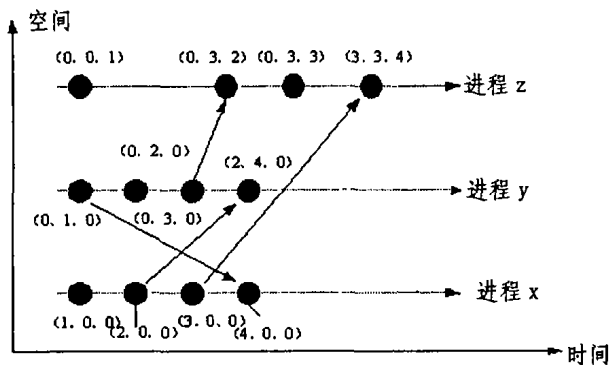


图3 向量时钟原理图

在以上算法中, $t-0$ 表示紧接在 t 之前的时刻, $dC_i(t)/dt > 0$ 意味着时钟 C 总是往前走。在许多应用中, 例如数据库管理系统, 消息按照它们发送的顺序被接收是很重要的。如每个消息是一个数据更新操作, 每个负责更新数据的进程按相同的顺序接收更新消息以维护数据库的一致性, 要求对消息进行排序。

设 $s(m)$ (或者 $r(m)$) 表示从进程 P_i 到进程 P_j 的消息 m 的发送 (或者接收) 事件, $C(e)$ 表示事件 e 被逻辑时钟 C 记录

下来的时间, 如果下列条件为真, 那么消息是有序的。

$$C(s(m_1)) \rightarrow C(s(m_2)) \Rightarrow C(r(m_1)) \rightarrow C(r(m_2))$$

两个从同一个进程 P_i 发出的消息在接收方会发生乱序, 这种情况可以用逻辑时钟解决, 通过赋值和初始化, 让逻辑时钟的值对应于进程的事件数, 当接收方收到一个乱序消息时, 它将一直等待直到该消息之前的所有消息到达。

当两个来自不同进程的消息到达接收方时乱序时, 来自 P_1 的消息 m_1 比来自 P_2 的消息 m_2 先发出, 但是迟收到。向量时钟可以解决这个问题。因为消息 m 中携带着发送进程的时钟信息, 接收者可以方便地区分哪个消息发生在先, 接收进程在先收到消息 m_2 的时候, 等待消息 m_1 的到来。所以为了保证分布式系统中消息的顺序, 基本方法是只有前一个消息递交后才能接收下一个消息。方法如下:

(1) 在发送消息之前, P_i 先更新它的 C_i ;

(2) 当 P_j 收到来自 P_i 的带 C_i 的消息 m 时, 不立即递交消息 m , 一直等到: ① $C_j[i] = C_i[i] - 1$; ② $C_j[k] \leq C_i[k] \quad \forall k \neq i$ 。

条件①保证进程 P_j 已经收到所有来自 P_i 的消息, 是 P_i 发出的。条件②保证 P_j 已经收到了所有那些在 P_i 发送消息 m 之前的消息。

用电的监测和管理是保证用电安全和节约用电的重要手段, 采用分布式的计算机技术来加强和完善用电管理的工作, 已引起人们的高度重视。研制和开发相关的产品也成了人们关注的问题。虹桥国际机场用电监测和管理系统经过一段时间的运行, 工作可靠, 性能良好。

参考文献

- 1 Buyya R. High Performance Cluster Computing, Volumel and 2. Prentice-hall, Inc. 1999
- 2 Wu J. Distributed System Design. CRC Press LLC, 1999. 43~108
- 3 杨学良, 等. 分布式多媒体计算机系统教程. 电子工业出版社, 2002. 76~115

(上接第181页)

作系统的热插拔功能是系统高可用技术中重点考虑的问题。

操作系统通过专用函数接口调用 Firmware 监控层函数。其调用一般采用直接物理调用方式实现。

3.4 资源管理系统

用户通过资源管理系统来管理和配置逻辑分区。简单的资源管理系统只是在系统启动时通过配置不同的启动配置文件实现逻辑分区的设定。系统一旦运行后分区结构不可再变化。但在动态逻辑分区中, 资源管理系统还要支持分区资源的动态在线配置能力, 能够在无须重启操作系统的情况下对分区的 CPU、内存和 IO 设备进行随意的添加和删除, 大大增强了逻辑分区的灵活性。动态逻辑分区需要资源管理系统和操作系统的紧密配合。

资源管理系统通过远程监视/控制实体、动态重构管理器 and 资源管理器实现动态的资源分配和去配。用户应用可通过操作系统提供的分区配置命令或函数来发起一次分区资源的在线重配置, 也可由用户在远程监视/控制实体来发起。动态重构管理器收到请求后将要求操作系统释放硬件资源。在成功释放资源后, 资源管理器将重新分配资源, 并调用相应的操作系统命令将新资源加到正在运行的操作系统中。

发展趋势和存在的问题 同高可用技术相结合的动态逻辑分区技术是分区技术发展的主流趋势。同时, 为了满足不同用户应用或同一应用中不同阶段应用模式的不同, 于应用相密切结合的目前负载动态平衡的动态逻辑分区技术更是未来

分区技术发展的热点。

由于目前的逻辑分区技术普遍采用了虚拟机技术, 势必导致性能的损失。同时, 大多数的动态逻辑分区技术主要是面向 SMP 系统的, 针对 ccNUMA 系统的通信和访存不一致特性的动态逻辑分区优化技术尚不成熟。动态逻辑分区技术对操作系统有或多或少的的影响, 这对用户的安装使用十分不便。如何尽量少地影响操作系统是目前动态逻辑分区技术要研究的关键。由于高可用技术本身的不成熟, 基于此的动态逻辑分区技术也并不成熟。实现一次动态的在线资源重分配性能低下, 无法满足灵活变化的负载平衡需求, 故针对高效、快捷的动态逻辑分区技术尚有待研究。

随着 SMP 技术和 ccNUMA 技术的不断成熟和完善, 计算机系统的规模将变得越来越大, 为了满足用户高可用和动态负载平衡, 动态逻辑分区技术将越来越重要。

参考文献

- 1 Dynamic Logical PARTitioning on the AIX-PPC Platform. <http://www.research.ibm.com/DLPPAR>
- 2 The OpenVMS Galaxy Advantage. <http://www.compaq.com/openvms/>
- 3 hp-ux virtual partitions-product brief. <http://www.hp.com/go/servicecontrol>
- 4 Disco: Running Commodity Operating Systems on Scalable Multiprocessors. <http://www-fash.stanford.edu/Disco>
- 5 Sun Enterprise(tm) 10000 Dynamic Reconfiguration User Guide. <http://www.sunmicro.com>