

GOSBMB: 基于客户操作系统行为的虚拟机内存均衡方法

钮 艳, 郑衍松, 杨 春, 程 旭

(北京大学信息科学技术学院, 北京 100871)

摘 要: 在虚拟机环境中为客户操作系统分配内存资源时, 需在性能和资源利用效率之间进行权衡. 本文提出一种基于客户操作系统行为的虚拟机内存均衡方法 GOSBMB (Guest Operating System Behaviors based Memory Balancer). 该方法在尽量降低性能损失的前提下, 根据客户操作系统中进程工作集和磁盘页面缓存对虚拟机监控器呈现的行为特征, 以对客户操作系统透明的方式估计它们各自的内存需求, 并按需动态调整内存资源. 在 Xen 上实现了 GOSBMB 原型系统, 实验表明, 使用 GOSBMB 动态调整内存资源时, 在节约内存资源达 69.6% 的情况下, 客户操作系统的性能损失低于 7.6%.

关键词: 虚拟机; 内存均衡; 客户操作系统行为; 性能

中图分类号: TP312 **文献标识码:** A **文章编号:** 0372-2112 (2011) 09-2178-06

GOSBMB: Dynamic Memory Balancing Based on Guest Operating System Behaviors for Virtual Machines

NIU Yan, ZHENG Yan-song, YANG Chun, CHENG Xu

(School of EECS, Peking University, Beijing 100871, China)

Abstract: The application performance is as important as the resource utilization when we allocate memory for guest operating systems in virtualized environments. This paper presents a memory-balancing method for virtual machines, named GOSBMB (Guest Operating System Behaviors based Memory Balancer). It aims to minimize the performance loss brought by balancing memory. The memory need of the guest operating system consists of two parts which respectively, cover the amount of the memory consumed by the process working sets, and the memory used as the page cache. GOSBMB estimates the memory need by observing VMM architecturally visible events transparently. Then it allocates the memory resource for the guest operating systems on demand. We have implemented a working prototype of GOSBMB on Xen platform. The memory can be saved up to 69.6% with only 7.6% performance loss when we use GOSBMB to balance memory dynamically.

Key words: virtual machine; memory balancing; guest operating system behaviors; performance

1 引言

随着虚拟化技术的广泛应用^[1], 虚拟机内存资源利用的研究也不断深入^[2~7]. 尽管内存容量不断扩充, 但内存却仍始终是紧缺资源, 在一台物理主机上运行多个客户虚拟机这种大数据集的计算环境中尤为凸显^[8]. 内存资源一旦分配不足, 将会导致显著的性能下降. 一种简单的策略是按峰值需求为客户操作系统提供内存, 在保证性能的同时却以牺牲资源利用效率为代价. 另一种策略是根据客户操作系统的内存需求动态调整内存资源. 在虚拟机监控器 (Virtual Machine Monitor, 简称 VMM) 中动态均衡内存资源时, 应尽量准确地估计客户操作系统的内存需求并按需分配内存资源, 避免因内存不足产生额外的磁盘访问, 导致性能下降. 一些研究^[2~4, 6, 7]对估计客户操作系统的内存需求和调整策略进行了探索,

他们多针对内存资源的利用效率进行优化, 对动态调整内存资源引起的性能下降的考虑尚有不足.

VMware ESX server^[2]基于工作集模型^[9], 利用取样技术, 通过监视部分内存页面的利用率估计全部内存页面的利用率, 进而获得客户操作系统的进程工作集^[2]. 该方法简单而易于实现, 且开销很低. 但其预测结果的准确性较低^[6], 尤其当内存分配不足时, 取样结果恒为 100%, 会对性能产生负面影响. 一些研究^[3, 4, 6]根据 MRC (Miss Ratio Curve, 简称 MRC)^[8]估计客户操作系统的内存需求, 他们的主要差异在于如何在 VMM 中获得客户操作系统的内存访问踪迹和 MRC. Jones 等人提出的 Geiger^[3]不直接跟踪内存页面的访问序列, 而是跟踪 VMM 可见的磁盘访问序列, 使用类似缓存镜像 (ghost buffer)^[10]的机制获得 MRC, 进而估计客户操作系统的内存需求. Geiger 只在内存分配不足而发生磁盘 I/O 时有

效,对性能影响较大.Lu 和 Shen 提出的 HECache(Hypervisor Exclusive Cache)^[4]方法,让 VMM 接管客户虚拟机的内存,能以较小的开销获得 MRC,但 HECache 需要修改客户操作系统.Zhao 等人于 2009 年提出的 MEB^[6]则通过降低内存页面访问权限以引发缺页故障获得 MRC;在内存分配不足时,MEB 还监视磁盘 I/O;MEB 对客户操作系统透明,但引入了额外的缺页故障,性能开销较大;MEB 不估计处于内核态的磁盘页面缓存的内存需求,对性能影响较大.GOSBMB 则根据对 VMM 可见的操作系统行为特征估计内存需求,不引入额外的缺页故障,性能开销极低;对磁盘页面缓存的内存需求,GOSBMB 也进行了额外考虑.

本文提出一种基于客户操作系统行为的虚拟机内存均衡方法,称为 GOSBMB(Guest Operating System Behaviors based Memory Balancer).GOSBMB 将尽量降低性能损失作为前提,根据客户操作系统中进程工作集和磁盘页面缓存对虚拟机监控器呈现的行为特征,以对客户操作系统透明的方式估计它们各自的内存需求,并按需动态调整内存资源.对进程工作集,GOSBMB 通过扫描页表和监视页表更新可准确跟踪和估计,平均估计误差仅为 3.1%;引入的性能损失也很低,平均性能开销仅为 0.5%.对磁盘页面缓存,GOSBMB 通过监视磁盘的重复装入操作进行估计和调整,平均性能开销仅为 0.3%.在 Xen 上实现了 GOSBMB 原型系统,实验表明,使用 GOSBMB 动态调整内存资源时,在节约内存资源达 69.6% 的情况下,客户操作系统的性能损失低于 7.6%.

2 GOSBMB 原理

通用操作系统大多基于虚拟内存技术,根据是否映射至进程的虚拟地址空间,可将物理内存分为已映射和未映射两部分.在物理内存充足时,已映射至进程虚拟地址空间的物理内存构成进程工作集(working set);为避免未映射的物理内存闲置浪费,通用操作系统会将其尽量利用以提升性能,磁盘页面缓存是未映射物理内存的主要使用方式.下面分别讨论如何估计和调整它们的内存需求.

2.1 跟踪进程工作集

工作集的概念最先由 Denning 于 1968 年提出,定义为一个进程当前正在使用的内存页面的集合^[11].在通用操作系统的实现中,常将驻留在内存中的虚拟地址空间子集当作工作集^[12].工作集需要通过页表转译才能被进程访问,都对应于有效的页表项.将填充了有效转译信息的页表项定义为有效页表项,则根据有效页表项可以准确地跟踪进程的当前内存需求.

GOSBMB 在 VMM 中通过扫描页表和监视页表更新

来跟踪客户操作系统中所有进程的有效页表项信息.记客户虚拟机的所有进程集合为 P ,进程 $p_i \in P$ 的工作集大小(working set size)为 WSS_i ,集合 P 中所有进程的总工作集大小为 $D = \sum WSS_i$.通过确定进程集合 P 和其中每个进程的 WSS_i ,累加 WSS_i 可获得总工作集大小 D ,如图 1 所示.在确定进程集合 P 时,采用类似 Antfarm^[13]的方法,为每个客户虚拟机维护一张进程记录表,使用页表基址作为进程标识;在跟踪进程 p_i 的内存需求时,根据页表基址指向的页表中有效的页表项确定 WSS_i .在客户操作系统新建进程或切换进程时,需要通过 VMM 将新的页表基址装入页表基址寄存器.如果新装入的页表基址不属于进程记录表,则将其加入进程记录表,并扫描页表中有效的页表项获得 WSS_i 的基值.在 p_i 加入进程记录表后的运行过程中,客户操作系统^[12,14]通常会按需分配和延迟释放物理内存页面.进程申请使用内存时,客户操作系统一般并不会立即将物理页面分配出去,而只分配虚拟地址;只有当进程真正访问到该虚拟地址时,发生缺页异常,系统才为之分配物理页面,并有效化页表项.反之,进程释放内存时,基于效率的考虑,操作系统也不一定立刻回收物理页面;只有当系统内存资源紧张或最终不再需要时,才进行回收并无效化页表项^[13].在此过程中,GOSBMB 通过监视页表的变化来动态调整 WSS_i .在客户操作系统销毁进程 p_i 时,需要通过 VMM 清除页表,这时,有效页表项不断减少, WSS_i 也相应降低,最后将该进程对应的页表基址从进程记录表中删除即可.

为进一步降低估计开销,在扫描页表和监视页表更新时,使用粗粒度的页目录项代替细粒度的页表项,该优化会轻度地过量估计进程工作集大小,平均估计误差为 7.3%.但使用该优化后,GOSBMB 需要扫描和监视的项数显著减少,引入的性能开销可得到进一步降低.同时,在分配额度方面,轻度的过量估计内存需求可缓解内存需求激增时可能引起的性能下降,有益于性能.在分配时机方面,客户操作系统为进程分配内存时,其相应的页目录项一定

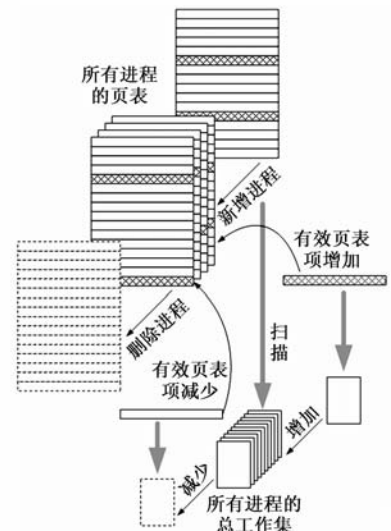


图1 基于页表跟踪进程工作集大小

已经被填充为有效内容;在客户操作系统回收完某页目录项所对应的全部内存后,其相应的页目录项才会被标记为无效;使用该优化可以提前为客户操作系统分配内存或延迟从客户操作系统回收内存,也有益于性能.该优化还可避免对超页(super page)的额外处理.

工作集不一定能够全部放入内存,部分虚拟地址可能被映射至磁盘而非内存.当进程访问到这部分虚拟地址时,客户操作系统需要从磁盘将其内容装入内存,性能会受到影响.为尽量避免此类情况的发生,GOSBMB将映射至磁盘的页表项也作为有效页表项并计入WSS_i和D.此外,在一些客户操作系统^[12,14]中,内核态访问的物理内存会映射至所有进程的虚拟地址空间,所有进程共享这部分物理内存.跟踪总工作集大小D时,只计数一次这部分虚拟地址空间对应的有效页目录项即可.

2.2 调整磁盘页面缓存

磁盘页面缓存具有“多分配多用,少分配少用”的特点,应为其分配内存以提高性能;但也应对其进行适当缩减,以获得更好的内存资源利用效率.通常来讲,当首次访问某数据时,需要从磁盘装入并将其加入至客户操作系统的磁盘页面缓存;当再次访问该数据时,

若其仍在磁盘页面缓存中,则可通过磁盘页面缓存直接访问^[12,14].为磁盘页面缓存适当地分配内存可提高程序性能,尤其是磁盘I/O密集型应用的性能.图2所示为三种不同磁盘页面缓存配置下编译Linux过程中内存的使用情况.打开磁盘页面缓存并分配足量内存(图2(a))时可获得最佳性能;关闭磁盘页面缓存(图2(b))时的性能则只有最佳性能的6.9%.但很难确定应为其分配的内存量.当无内存压力时,磁盘页面缓存会不断增大,一些以后可能不再访问的磁盘页面会一直保留在磁盘页面缓存中;只有当客户操作系统有内存压力时,才会缩减磁盘页面缓存.若主动地适当缩减磁盘页面缓存,释放不再访问的磁盘页面占用的内存,供其它更需要内存的客户虚拟机使用,物理主机的内存资源从整体上可获得更好的利用效率.如图2(c)所示,若打开磁盘页面缓存但不分配足量内存,客户操作系统会缩减磁盘页面缓存,与最佳性能的情况相比,在性能损失11.7%的情况下,内存可节约75.0%.理想情况下,缩减磁盘页面缓存时,客户操作系统的页面替换算法会按照未来访问概率递减的次序依次换出页面,若缩减过度而使得将访问的数据被换出,则会从磁盘再次装入该数据.将从磁盘再次装入数据的动作定义为磁盘重复装入操作.

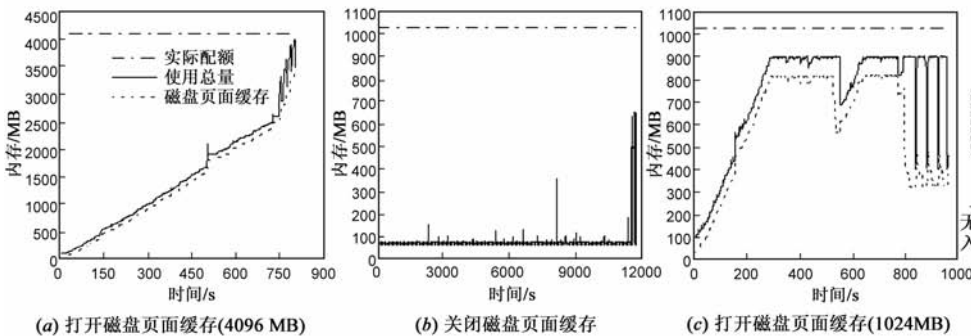


图2 运行BuildLinux时客户操作系统中内存的使用情况

GOSBMB在VMM中监视对其可见的磁盘访问操作并根据磁盘重复装入操作动态调整磁盘页面缓存,如图3所示.初始时,GOSBMB为磁盘页面缓存分配定量内存,当有等量的磁盘操作发生后,磁盘页面缓存中就会填满这些磁盘操作访问的数据.然后,以磁盘重复装入操作为信号,用“试错法”(trial and error)不断调整磁盘页面缓存的大小.GOSBMB维护一个磁盘访问位图,以获得磁盘的重复装入操作信息.当无磁盘重复装入操作发生时,GOSBMB认为这时为磁盘页面缓存提供的内存充足,其中可能存在以后不再访问的磁盘页面,则不断缩减磁盘页面缓存,并释放相应内存;一旦发生磁盘重复装入操作,GOSBMB认为过度缩减了磁盘页面缓存,致使一些还会再访问的磁盘页面也被挤出内存,则立即停止缩减磁盘页面缓存并为之增加内存.

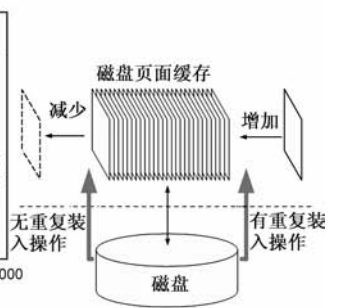


图3 基于磁盘调整页面缓存大小

2.3 按需动态调整内存资源

在跟踪进程工作集大小和调整磁盘页面缓存的基础上,GOSBMB累加进程总工作集和磁盘页面缓存的内存需求估计值,获得客户操作系统的内存总需求估计值,在物理主机内存总量允许的情况下,按需动态调整内存资源.对于总工作集的内存需求估计值,GOSBMB在跟踪其当前内存需求的基础上留出“安全”裕量,以缓解内存需求激增时由于内存不足对性能产生的负面影响.磁盘页面缓存的内存需求估计值则直接等于磁盘页面缓存的当前实际值.在通用操作系统中,进程工作集和磁盘页面缓存一般会有部分重叠,即,部分物理内存既处于进程工作集中,又处于磁盘页面缓存中.为尽量降低开销,GOSBMB不对此进行额外处理.在获得

内存总需求估计值后,通过气球驱动(balloon)^[2,15]按需动态调整内存。

GOSBMB的系统结构如图4所示.主要包括位于VMM中的进程与页表监视器、磁盘监视器和内存均衡器三部分.进程与页表监视器负责跟踪进程工作集的大小,当客户操作系统请求切换或修改页表时,它根据请求信息扫描页表和监控页表更新情况,并将信息传递给内存均衡器.磁盘监视器负责调整磁盘页面缓存的大小,当发生磁盘读写操作时,磁盘监视器截获请求并分析是否属于磁盘重复装入操作.内存均衡器则根据前两者提供的信息按需动态调整客户操作系统的内存资源。

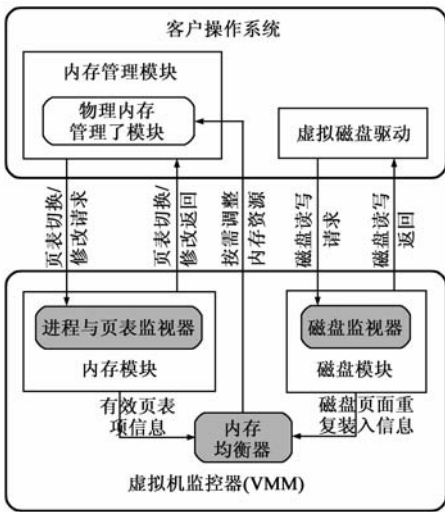
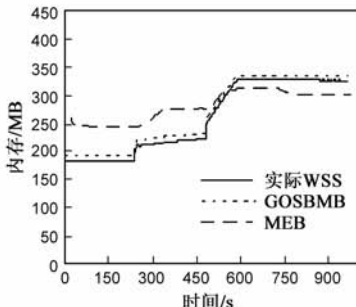


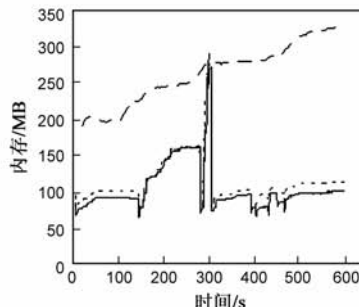
图4 GOSBMB的系统结构

3 实验结果与分析

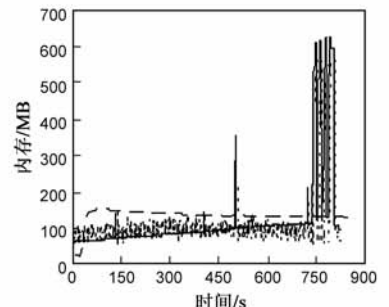
为评测 GOSBMB 的性能,在 Xen-3.4.2 和 Linux-2.6.18.8 上实现了 GOSBMB 原型系统.物理主机的软硬件配置如表 1 所示,采用的测试程序如表 2 所示.GOSBMB 的参数可自由配置,评测 GOSBMB 和 MEB^[6]使用的参数如表 3 所示,其中,GOSBMB 采用了比 MEB 更短的调整周期.调整周期越短,估计内存需求的准确度越高,但系统开销也会相应提高.尽管如此,GOSBMB 的系统开销仅为 0.8%,仍远远低于 MEB 的 24.0%。



(a) SPECjbb2005



(b) Dacapo



(c) BuildLinux

图5 估计进程工作集的内存需求(基准测试程序和实际应用)

表 1 实验环境的软硬件配置

	硬件	软件
物理主机	AMD Opteron 2214 2.2GHz (dual core) x 2 处理器; 8 GB 内存;320 GB SATA 硬盘	Xen-3.4.2; Linux-2.6.18.8; RedHat EL-5.3
客户虚拟机	VCPU x 1 处理器; 128 MB ~ 4 GB 内存; 4 GB 虚拟硬盘	Linux-2.6.18.8; Fedora 12

表 2 使用的测试程序

类型	测试程序	描述
访存密集型应用	SPECjbb2005	使用缺省配置;java 运行时环境使用 OpenJDK 1.6.0; warehouse 数目从 1 递增至 4,步长为 1.
	Dacapo	使用 large 尺寸,java 运行时环境使用 OpenJDK 1.6.0.
磁盘 I/O 密集型应用	BuildLinux	使用 allyesconfig 编译 Linux 内核.
计算密集型应用	SPEC CPU2006	运行 SPEC INT2006.

表 3 评测 GOSBMB 和 MEB 时采用的参数

调整周期	基于经验选择的最佳参数	
GOSBMB	1 秒	对总工作集大小,“安全”裕量 = 10 MB; 对磁盘页面缓存,增长量 = 从磁盘装入的数据量/32,降低量 = 4 MB.
MEB ^[6]	12 秒	LRU 跟踪单元 = 32 页;hot set size = 8 MB;允许的缺页故障率 = 5%.

3.1 估计进程工作集的准确度

图 5 所示为使用 Java 基准测试程序 SPECjbb2005 和 Dacapo 以及实际应用 BuildLinux 评测 GOSBMB 和 MEB 估计进程工作集的准确度的实验结果.为系统配置足量内存,这时“实际 WSS”值等于系统实际消耗的内存总量与磁盘页面缓存消耗的内存量之差.对 SPECjbb2005 和 Dacapo, GOSBMB 可以很好地跟踪内存需求,平均估计误差分别为 0.4% 和 1.5%.对 BuildLinux,由于频繁地创建和销毁进程,进程生存期短,页目录指向的页表页面中有效页表项较少,使用有效页目录项代替有效页表项后会有较大误差,平均估计误差

为 7.3%。MEB 实现的原型系统只能估计进程级的内存需求,对 SPECjbb2005、Dacapo 和 BuildLinux,平均估计误差分别为 12.6%、151.1% 和 45.0%。

3.2 系统开销

图 6 所示为使用 Dacapo、SPECjbb2005、BuildLinux 和 SPEC CPU2006 评测 GOSBMB 与 MEB 的时间开销的实验结果。为区分 GOSBMB 在估计进程工作集和磁盘页面缓存的内存需求时分别产生的时间开销,对以下四种情况进行评测:原始情况、使用 MEB 估计内存(MEB)、只使用 GOSBMB 的进程与页表监视器估计内存(GOSBMB-WS)和使用完整的 GOSBMB 估计内存(GOSBMB)。四种情况中都为客户虚拟机配置了足量内存;后三者估计内存时,只收集数据并估计内存需求,并不根据估计值实际调整内存资源。GOSBMB 根据 VMM 可见的客户操作系统行为特征估计和调整内存,其时间开销主要来自扫描与监测页表和监测磁盘的操作,不引入额外的缺页故障,平均时间开销仅为 0.8%。GOSBMB-WS 的时间开销则更低,只包括扫描和监测页表时引入的操作开销,平均时间开销为 0.5%。MEB 为了跟踪内存页面的访问踪迹引入了额外的缺页故障,时间开销相对较大,平均为 24.0%。

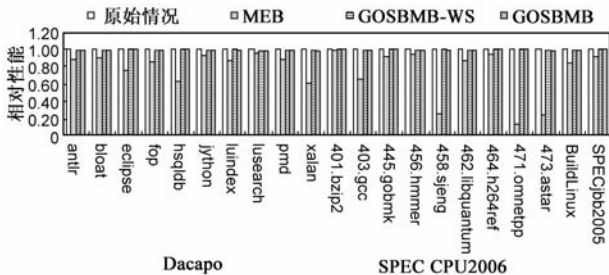


图6 GOSBMB和MEB的时间开销

GOSBMB 的空间开销主要来自位于内存中的进程记录表和磁盘访问位图。每个客户虚拟机对应一个进程记录表和一个磁盘访问位图。进程记录表中,每个进程占用一个表项,消耗 4 字节内存,可忽略不计。磁盘访问位图中,每个磁盘块占用 1 位,在 Linux 中,缺省的磁盘块单元为 512 字节,1GB 磁盘需要的位图大小为 256

KB(2^{21} 位 = 2^{18} 字节 = 256 KB)。若按类似多级页表的组织方式,只为访问过的磁盘块维护位图,可大幅降低磁盘访问位图占用的内存空间。

3.3 性能表现与内存节约

图 7 和表 4 所示使用 BuildLinux、SPECjbb2005 和 Dacapo 评测 GOSBMB 动态分配内存时的性能表现和内存节约情况。MEB 只估计进程级内存需求,不估计磁盘页面缓存的内存需求,经实验,使用 MEB 动态调整内存时程序性能普遍较差,不报告 MEB 的评测结果。首先按峰值需求为测试程序分配内存(图 7 的“最佳性能”),尽管可以获得最佳性能,但消耗内存较多。如欲提高内存利用效率,就需降低内存分配,分配的内存为使用 GOSBMB 动态调整内存资源时消耗的内存均量(图 7 的“固定分配”)。这时,Dacapo 性能下降最为显著,与最佳性能相比,性能损失高达 70.0%;BuildLinux 次之,性能损失为 34.2%;SPECjbb2005 的内存需求较为平缓,其受到的影响也最小,性能损失为 7.3%。如果采用 GOSBMB 动态调整内存(图 7 的 GOSBMB),则既可以节约内存资源,又对性能的负面影响很小。与最佳性能情况相比,BuildLinux 在节约内存资源 69.6%的情况下,性能损失仅为 3.7%;Dacapo 和 SPECjbb2005 在节约内存资源 59.7% 和 22.2% 的情况下,性能损失分别为 7.6% 和 2.7%。

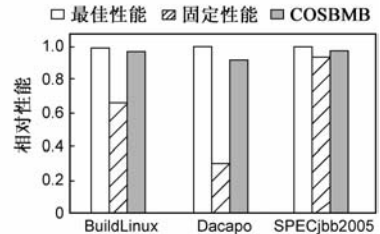


图7 使用不同内存分配策略时程序的性能表现

客户虚拟机发生的缺页故障总数和磁盘重复装入总量如表 4 所示。最佳性能时,不发生缺页故障和磁盘重复装入操作,但消耗的内存最多;使用 GOSBMB 动态调整内存时,消耗的内存显著降低,由于估计误差,存在少量的缺页故障和磁盘重复装入操作;固定分配时,消耗的内存等于 GOSBMB 动态调整内存的内存均量,但缺页故障数和磁盘重复装入操作最多,性能显著低于 GOSBMB 动态调整内存的情况。

表 4 使用不同内存分配策略时消耗的内存均量、缺页故障总数和磁盘重复装入总量

	BuildLinux			Dacapo			SPECjbb2005		
	最佳性能	固定分配	GOSBMB	最佳性能	固定分配	GOSBMB	最佳性能	固定分配	GOSBMB
消耗的内存均量(MB)	3972	1208	1208	479	193	193	356	277	277
缺页故障总数(个)	0	222	25	0	592	6	0	187	38
磁盘重复装入总量(MB)	0	5854	113	0	6792	544	0	1504	489

4 结论与未来工作

随着虚拟化技术的再次兴起,虚拟机内存资源的

分配和使用也成为重要的研究问题。内存资源一旦分配不足,将会产生显著的性能下降。如果按峰值需求供应内存,则以牺牲内存资源利用效率为代价。如果

动态调整内存资源,则应尽量准确地估计内存需求并按需分配内存资源,避免因内存不足,产生额外的磁盘访问,导致性能下降.已有相关研究对动态调整内存资源时引入的性能下降尚欠充分考虑.为此,本文提出一种基于客户操作系统行为的虚拟机内存均衡方法,称为 GOSBMB.它将尽量降低性能损失作为前提,以对客户操作系统透明的方式,根据客户操作系统中进程工作集和磁盘页面缓存对虚拟机监控器呈现的行为特征估计它们各自的内存需求,并按需动态调整内存资源.

为充分发挥 GOSBMB 在透明性方面的优势,未来将首先把 GOSBMB 实现在 Xen 的影子页表模式中,使之可以跟踪和调整商业客户操作系统如 Windows 的内存.此外,对共享内存的额外处理也是未来要考虑的内容.

致谢 感谢 Weiming Zhao 和 Xinxin Jin 提供的 MEB 代码、指导与建议.

参考文献

- [1] 张彬彬,罗英伟,汪小林,等.虚拟机全系统在线迁移[J].电子学报,2009,37(4):894-899.
ZHANG Bin-bin, LUO Ying-wei, WANG Xiao-lin, et al. Whole-system live migration mechanism for virtual machines [J]. Acta Electronica Sinica, 2009, 37(4): 894-899. (in Chinese)
- [2] Waldspurger CA. Memory resource management in VMware ESX server[A]. Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI'02)[C]. New York, USA: ACM Press, 2002. 181-194.
- [3] Jones ST, Arpaci-Dusseau AC, Arpaci-Dusseau RH. Geiger: Monitoring the buffer cache in a virtual machine environment [A]. Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'06)[C]. New York, USA: ACM Press, 2006. 14-24.
- [4] Lu P, Shen K. Virtual machine memory access tracing with hypervisor exclusive cache[A]. Proceedings of the 2007 USENIX Annual Technical Conference (USENIX'07)[C]. New York, USA: ACM Press, 2007. 29-43.
- [5] Gupta D, Lee S, Vrable M, et al. Difference engine: harnessing memory redundancy in virtual machines[A]. Proceedings of the 8th Symposium on Operating Systems Design and Implementation (OSDI'08)[C]. New York, USA: ACM Press, 2008. 309-322.
- [6] Zhao W, Wang Z, Luo Y. Dynamic memory balancing for virtual machines [A]. Proceedings of the 5th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE'09)[C]. New York, USA: ACM Press, 2009. 37-47.
- [7] Hwang W, Roh Y, Park Y, et al. Hyperdealer: reference-pattern-aware instant memory balancing for consolidated virtual machines[A]. Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD'10)[C]. Washington, DC, USA: IEEE Computer Society Press, 2010. 426-434.
- [8] Zhou P, Pandey V, Sundaresan J, et al. Dynamic tracking of page miss ratio curve for memory management [A]. Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'04)[C]. New York, USA: ACM Press, 2004. 177-188.
- [9] Denning PJ. The working set model for program behavior[J]. Communications of the ACM, 1968, 11(5): 323-333.
- [10] Patterson RH, Gibson GA, Ginting E, et al. Informed prefetching and caching [A]. Proceedings of the 15th ACM Symposium on Operating System Principles (SOSP'95)[C]. New York, USA: ACM Press, 1995. 79-95.
- [11] Tanenbaum AS. Modern operating system[M]. Saddle River, NJ, USA: Prentice Hall Press, 2007.
- [12] Russinovich ME, Solomon DA. Microsoft Windows Internals [M]. Redmond, WA, USA: Microsoft Press, 2004.
- [13] Jones ST, Arpaci-Dusseau AC, Arpaci-Dusseau RH. Antfarm: tracking processes in a virtual machine environment [A]. Proceedings of the 2006 USENIX Annual Technical Conference (USENIX'06)[C]. New York, USA: ACM Press, 2006. 1-14.
- [14] Bovet DP, Cesati M. Understanding the linux kernel[M]. Sebastopol, CA, USA: O'Reilly Media Press, 2005.
- [15] Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization [A]. Proceedings of the 19th Symposium on Operating Systems Principles (SOSP'03)[C]. New York, USA: ACM Press, 2003. 164-177.

作者简介



钮 艳 女, 1982 年 9 月出生于河南省安阳市. 博士生. 主要研究方向为虚拟机、操作系统与系统软件、交互性能和瘦客户计算.

E-mail: niuyan@mprc.pku.edu.cn



郑衍松(通讯作者) 男, 1980 年 4 月出生于广东省汕头市. 现为北京大学信息科学技术学院博士后, 主要研究方向为系统软件和系统芯片及软硬件协同设计.

E-mail: zhengyansong@mprc.pku.edu.cn