

虚拟化环境中内存隔离性测试与分析

周 斌

(中南民族大学计算机科学学院, 武汉 430074)

摘 要:设计一个虚拟化内存性能隔离度评测系统,该系统采用细粒度的干扰方法,针对虚拟化系统的内存带宽和内存空间的隔离性进行干扰测试,测试多个虚拟机同时访问内存时性能之间的相互影响。测试结果表明,当前的硬件和软件平台在虚拟化性能隔离度的支持方面依然存在着较大的缺陷,在存在干扰情况下,被测虚拟机的性能指标下降严重。

关键词:虚拟化; 性能隔离度; 干扰测试; 内存

Test and Analysis of Memory Isolation Performance in Virtualization Environment

ZHOU Bin

(School of Computer Science, South-central University for Nationalities, Wuhan 430074, China)

【Abstract】 By means of the fine-grained interference and the disturbed test of the memory bandwidth and the memory space isolation, this paper designs an evaluation system of the memory performance isolation in virtualization from which the interaction of multiple VMs visiting the memory at same time can be obtained. Test result shows that the major flaw still exists in the current hardware and software platforms support with the concern of the performance isolation of the virtualization, and the measured performance of the VMs is down severely in face of interference.

【Key words】 virtualization; performance isolation; disturbed test; memory

1 概述

虚拟技术通过一个虚拟机管理程序,可以在一台主机上虚拟多个软件环境或者使得桌面用户可以在一个物理机上同时运行多个操作系统,提供了虚拟机环境的可扩展性和资源控制^[1]。虚拟机之间是相互隔离的,这样不同的操作系统和应用程序就能在同一个物理主机上互不干扰地运行。但灵活性是有代价的。虚拟化给系统增加了复杂度,并给性能增加一定的开销。

当前对于虚拟化环境下的性能评测研究,主要集中于4个方面:(1)服务器整合的性能,比如 vConsolidate^[2]和 Vmmark^[3];(2)虚拟系统的微观性能指标评测,通过使用微观评测程序对比虚拟化环境下的虚拟机同物理机操作系统作性能差异,从而找出虚拟化技术的开销;(3)虚拟机迁移的性能研究;(4)运用 profile 的方法^[4]诊断系统性能开销。

但上述方面着重于对影响单个虚拟机性能指标的研究,而忽略了在虚拟化环境下性能评测的一个重要方面,即虚拟机的性能隔离度。多台虚拟机整合在一台物理机,虽然提高了物理资源利用率,但多虚拟机相互之间会产生影响,从而降低系统的整体性能。这种影响包括错误影响和性能影响。错误影响是指当系统中某个虚拟机发生错误时,会影响到整个系统,甚至导致整个系统瘫痪;而性能影响是指某个虚拟机的运行导致其他关键服务性能降低,给其他用户造成严重损失。

在虚拟化系统各方面的性能指标中,性能隔离度是非常

关键的一项。传统的测试程序并非专门针对虚拟化环境开发,没有考虑到虚拟化环境下的特殊性,所以,在评测的种类和准确度方面表现不足。而当前的基于虚拟化环境的评测套件 vConsolidate 和 Vmmark 主要关注的是系统整合虚拟机的整体性能,因此,也不能用来评测虚拟化平台的性能隔离度。当前对虚拟化性能隔离度的研究大多停留在使用宏观的基准程序进行测试,这种测试结果只能反映宏观性能。还有一些研究是为了加强性能隔离度。

2 内存带宽的重要性

影响虚拟系统隔离性能的因素比较多,如高速缓存、内存带宽^[5]、内存空间膨胀、磁盘、CPU 和网络等都会对系统的隔离性产生影响。但内存带宽是对其影响较大的一个因素,本文首先选择这个最重要的因素进行测试和分析。

分析应用程序的行为,可以看出应用程序主要有3种行为:(1)读写文件;(2)从网络接收或者发送数据;(3)为映射一个文件到内存中。这3种方法都要对内存进行大量访问。当多个虚拟机整合在同一个物理机上时,物理机的内存处理能力很容易成为系统的瓶颈。

作者简介:周 斌(1971—),男,讲师、博士研究生,主研方向:虚拟计算,网络工程

收稿日期:2010-07-08

E-mail: benz20000@tom.com

3 虚拟化内存性能隔离度评测系统

本文设计了一个用来评测虚拟化系统内存性能隔离度的系统。它考虑到虚拟化场景下性能评测的特殊性,采用细粒度的干扰测试程序,用户可以根据自己需求方便地定制测试环境,同时也可以用于诊断虚拟化平台的细粒度的性能隔离问题。它包括干扰程序、基准程序和一个测试管理程序。

虚拟化内存性能隔离度评测系统测量的是虚拟化平台内存隔离性能影响的能力,因此,需要在多个虚拟机之间协同测试。这就需要有一个协同多个虚拟机的管理程序来负责测试程序的部署、启动、结果收集、处理等。为此,系统包含三层结构,系统的详细结构如图 1 所示。

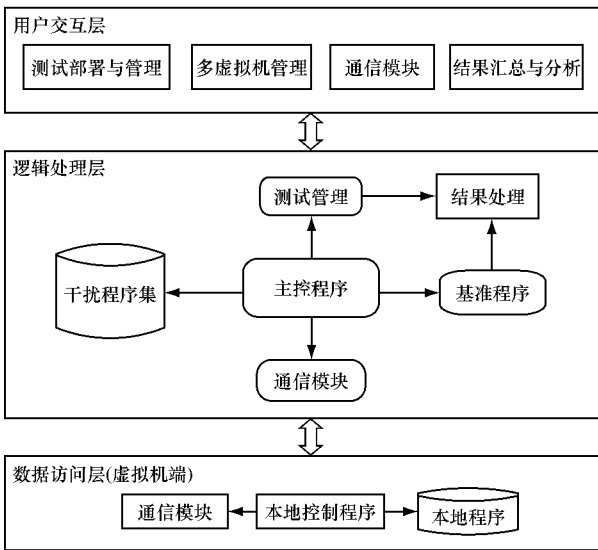


图 1 内存性能隔离度评测系统结构

三层结构把应用程序分为 3 个层次:用户交互层,逻辑处理层,数据访问层。用户交互层负责用户输入输出的处理,通常并不对输入输出进行处理。逻辑处理层连接了用户交互层和数据访问层,主要的责任是解释用户输入、控制处理过程、输出处理结果。数据处理层则主要是完成实际的操作。

4 内存测试干扰程序

虚拟机的内存有大有小,如果内存过小从而不能容纳全部的测试程序,那么测试程序在运行中将会频繁地访问交换分区,这时被测项目的结果将会严重受到磁盘性能的影响。如果此时用户的测量项目不是磁盘,而是其他项目,如内存带宽、内存空间等,那么测试结果将会受到磁盘的影响而表现出不准确,甚至得到测试结果可能是磁盘性能隔离度的结果,从而影响用户的判断。为了能给出准确可信的测试结果,本文采用了预分配内存检测方法。

预分配内存检测方法首先检查当前虚拟机的可用内存是否足够容纳所有的干扰程序。程序访问内存数据时,如果此时内存不足,那么程序将等待内核把一些内存中的内存页刷新到磁盘上,从而腾出一些空闲的内存页给用户程序使用。刷新内存页到磁盘是一个耗费时间的过程,在这个时间段,程序将被阻塞直至有可用内存位置。分析可知,当内存中有空闲页时,程序将很快就访问完毕;而内存资源紧张时,程序将花费大量时间。

预分配内存检测算法就是基于这种情况下,程序访问内存数据所花费的时间的差异设计的。下面给出预分配内存检测算法的具体处理流程:

输入 期望的内存大小

输出 可用的内存大小

步骤 1 分配预需求大小的内存空间;当前访问位置指向已分配内存的开始。

步骤 2 访问当前访问位置开始的一页内存;记录访问时间 time。

步骤 3 如果访问时间 time 大于访问内存时间,转步骤 4。否则,当前访问位置前进一页大小。如果当前位置超出了分配预需求大小的内存空间转向步骤 4;否则转步骤 2。

步骤 4 根据当前指针和已分配的内存空间首地址计算可用内存大小,返回。

5 内存带宽隔离性测试程序

程序对内存的操作都是通过 bcopy() 函数调用来完成,而 bcopy() 函数最终由直接的内存赋值操作来完成,本文的内存干扰模块也采用直接内存字节拷贝的方式进行设计,如图 2 所示。

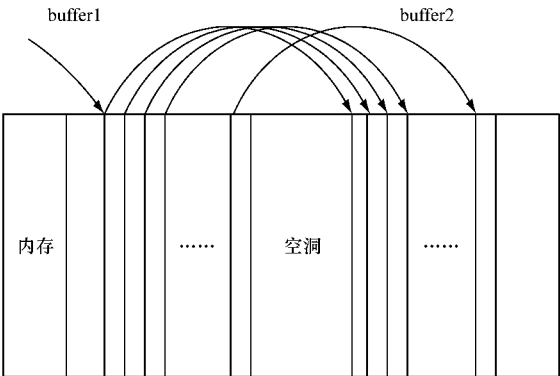


图 2 内存带宽干扰操作

下面给出内存带宽干扰算法的具体处理流程:

步骤 1 分配 2 个缓冲区间 buffer1 和 buffer2。

步骤 2 把 buffer1 和 buffer2 的地址放入寄存器变量 source 和 target。

步骤 3 如果 source 小于缓冲区数组 buffer1 的末地址,则把从 source 开始的 128 个字节拷贝到 target 开始的地址空间;否则转步骤 2。

步骤 4 source 和 target 都前进 128 个字节;转步骤 3。

图 2 为操作示意图,在程序开始时,先分配 2 片内存空间。然后定义了 2 个寄存器变量,分别指向 2 片内存空间的开始。寄存器变量把变量存放在寄存器中,当再次使用变量时,只需从寄存器中读取,不用再访问内存,从而减少无关的重复访存操作,这样就减少了不必要的干扰。之后,从 2 片内存空间的开始进行复制操作,当进行到内存空间的结尾时,重新从开始复制。

6 内存隔离性测试与分析

内存隔离性测试分为 2 个测试:(1) 内存空间隔离性测试;(2) 内存带宽隔离性测试。内存空间隔离性测试是为了衡量虚拟化平台在隔离虚拟机内存空间膨胀时所产生的性能

影响的能力。内存带宽隔离性测试程序测量了多个虚拟机同时访问内存时,性能之间的相互影响。

图3显示了内存空间和内存带宽干扰程序对被测虚拟机平均内存访问延迟的影响。图4显示了内存带宽对各种本地进程间通信带宽的影响。

图3中的正常情况是没有干扰程序时,被测虚拟机的主存顺序访问和随机访问的响应时间^[6];干扰情况是干扰虚拟机中有干扰程序运行时,被测虚拟机的主存顺序访问和随机访问的响应时间。其中,前2组为内存带宽干扰测试的结果,后2组为内存空间膨胀的测试结果。从图中可以看出,在有干扰程序的情况下,不管是内存带宽干扰还是内存空间膨胀干扰,被测虚拟机的性能都有所下降。其中内存带宽的影响更大,被测虚拟机的内存顺序访问时间延迟从140.2 ns增加到168.5 ns,随机访问延迟从155.6 ns增加到183.9 ns。而内存空间膨胀干扰只使得被测虚拟机内存顺序访问的时间延迟从140.2 ns增加到142.8 ns,随机访问延迟从156.4 ns增加到158.9 ns。

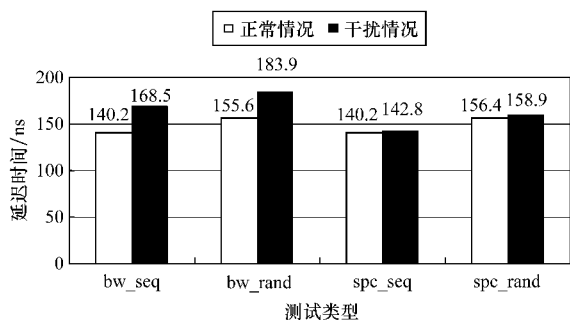


图3 内存隔离性测试结果

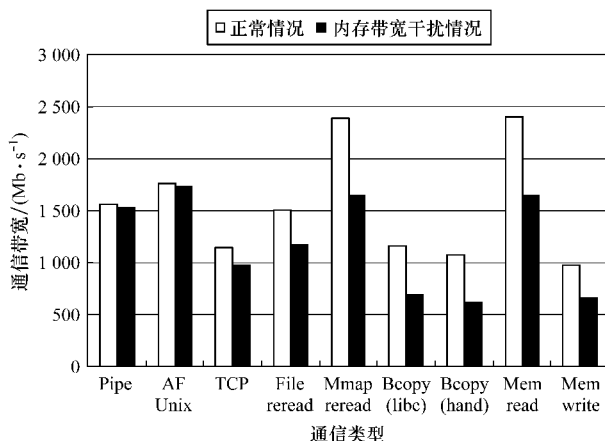


图4 内存带宽干扰情况下的本地进程通信带宽

图4显示了内存干扰程序对被测虚拟机的一些本地数据传递带宽的影响。正常情况是没有干扰程序时被测虚拟机的本地数据传递的带宽,干扰情况是被测虚拟机在有干扰情况下的本地数据传递的带宽。从图中可以看出,被测虚拟机的本地数据传递带宽在有干扰机的情况下都有所下降。

发生上述情况的主要原因在于当前硬件架构并没有提供内存带宽的一些控制策略,不能很好地支持虚拟化技术,尤其

是保证有特殊需求的虚拟机的性能。所有的虚拟机对内存带宽的争用是一种混乱的状态,某个虚拟机数据块拷贝的操作多,它就使用多的内存带宽;当其他虚拟需要使用内存时,就必须等待。这种混乱的控制状态在一定程度上提升了某些虚拟机的内存操作的性能,但是它也损害了其他的具有关键服务的虚拟机的性能,致使其他虚拟机的性能处于一种不稳定的状态。

传统的计算机中关于资源的合理分配是由操作系统完成的,是以CPU为核心的,调度的公平性也主要围绕在CPU资源的公平利用。在某一时刻,一个物理机上只能有一个操作系统在运行,一个操作系统中通常只有一种服务在运行。在这种情况下,以CPU的公平利用为核心的资源分配方案不会产生性能影响的问题。但是在虚拟化场景下,每个运行着服务的虚拟机不再是一个简单的进程^[7],它对于用户来讲就是一个物理机。可是,当前对虚拟环境下,虚拟机监控软件对资源分配也还是停留在以CPU为核心的阶段,其结果正如内存带宽隔离性测试结果所显示的那样,将会导致一些虚拟机的内存的性能受到较大影响。

7 结束语

本文通过使用虚拟化内存性能隔离度评测系统针对Xen虚拟化平台进行了测试,给出了测试结果。分析发现,当前的硬件和软件平台在虚拟化性能隔离度的支持方面依然存在着很大的缺陷,内存带宽的性能隔离度很差,在有干扰情况下,被测虚拟机的性能指标下降严重。可见,当前的硬件体系结构在某些方面还不能很好地支持虚拟化技术实现虚拟机的性能隔离。这些不足之处为进一步地研究和改进性能提供了方向。

参考文献

- [1] Barham P, Dragovic B, Fraser K, et al. Xen and the Art of Virtualization[C]//Proceedings of the 19th Symposium on Operating System Principles. Bolton Landing, USA: [s. n.], 2003: 164-177.
- [2] Casazza J P, Greenfield M, Shi K. Redefining Server Performance Characterization for Virtualization Bench-marking[J]. Intel Technology Journal, 2006, 10(3): 243-252.
- [3] Makhija V, Herndon B, Smith P, et al. VMmark: A Scalable Benchmark for Virtualized Systems[R]. LOS Angeles, USA: VMware Inc., Technical Report; VMware-TR-2006-002, 2006.
- [4] Levon J, Elie P. Oprofile: A System Profiler for Linux[EB/OL]. (2004-10-20). <http://oprofile.sf.net>.
- [5] 徐正超, 喻成. Java虚拟机中内存管理机制研究[J]. 中南民族大学学报:自然科学版, 2007, 26(3): 87-91.
- [6] Endo Y, Wang Z, Chen J, et al. Using Latency to Evaluate Interactive System Performance[J]. ACM SIGOPS Operating Systems Review, 1996, 30(S1): 185-199.
- [7] 杜海, 陈榕. 基于完全虚拟化的进程监控方法[J]. 计算机工程, 2009, 35(8): 88-89.

编辑:索书志