doi:10.3772/j.issn.1002-0470.2015.06.007

延迟存储:一种降低虚拟机退出开销的方法①

吴瑞阳②* ** *** 台运方****

(*中国科学院计算技术研究所计算机体系结构国家重点实验室 北京 100190)

(**中国科学院大学 北京 100049)

(***龙芯中科技术有限公司 北京 100190)

(****甲骨文软件研究开发中心(北京)有限公司 北京 100193)

摘 要 研究了虚拟机退出及恢复运行时的开销问题,提出了一种用于降低虚拟机切换时进行保存及恢复现场的开销的延迟存储方法。该方法的主要思想是利用修改虚拟机软件源代码的方式,通过判断虚拟机恢复运行时是否依然是上次退出时的同一个虚拟机,来减少需要保存和恢复的寄存器数量。这个方法不需要对硬件设计进行改动,可以支持多核操作系统和多个虚拟机同时运行的情况,因此具有广泛的适用性。在龙芯3A1500处理器平台上的试验结果显示,上述延迟存储方法与现有方法相比,可以降低虚拟机退出开销65%,虚拟机整体性能提升3%到10%。

关键词 系统虚拟化,虚拟态,虚拟机退出,延迟存储

0 引 言

进入21世纪以来,计算资源和网络带宽得到了大幅的提升,使得云计算的概念越来越流行。系统虚拟化[1,2]作为云计算中的关键技术,也得到了越来越快的发展。系统虚拟化是通过在宿主机上运行虚拟机程序,将一台物理机器虚拟为另一台或多台虚拟机器。无内部互锁流水级的微处理器(microprocessor without interlocked piped stages, MIPS)架构[3]中对系统虚拟化加入了特定的硬件支持[4,5],其中最重要的就是虚拟态模式。大多数指令都可以在虚拟态模式下运行,然而,还是有一些特权指令不能在虚拟态下运行,需要退出虚拟态,在宿主机中进行模拟,这种退出过程被称为虚拟机退出。多个虚拟机之间进行切换、虚拟机运行时出现特权例外等情况也会导致虚拟机退出。虚拟机退出时,需要进行现场保存,当宿主机处理完毕并再次切换回到虚拟态时,再进行现场恢复。以MIPS虚拟机架构为例,虚拟机有很多需要保存和恢复的现场,包括通用定点寄存器、浮点寄存器,以及一些虚拟机专用的寄存器,如虚拟态处理器控制寄存器[5](CP0寄存器)等。保存和恢复现场是虚拟机退出与恢复运行过程中的主要工作。本文将这两部分开销都视为虚拟机退出带来的开销。

本文针对虚拟机退出及虚拟机恢复运行时的开销问题,提出了一种延迟存储的方法。用此方法,虚拟机退出时暂不保存虚拟机专用寄存器,而是在虚拟机恢复运行时判断与之前退出时是否是同一个虚拟机:如果是同一个虚拟机,则不需要保存和恢复这些虚拟机专用寄存器;如果不是同一个虚拟机,则进行完整的现场保存与恢复工作。该方法可以支持多

① "核高基"科技重大专项课题(2009ZX01028-002-003, 2009ZX01029-001-003, 2010ZX01036-001-002, 2012ZX01029-001-002-002, 2014ZX01020201),国家自然科学基金(61221062, 61100163, 61133004, 61173001, 61232009, 61222204, 61432016)和863计划(2012AA010901,2012AA011002, 2012AA012202, 2013AA014301)资助项目。

② 男,1991年生,博士生;研究方向:高性能计算机体系结构;联系人,E-mail: wuruiyang@ict.ac.cn

(收稿日期:2014-11-20)

核处理器的情况,不需要进行硬件改动,也不局限于 MIPS 虚拟化架构。在龙芯 3A1500 多核处理器上的试验结果显示,单处理器核上运行单虚拟机时,延迟存储方法可以降低 65% 的虚拟机退出开销,虚拟机的整体性能提升了 5% 到 10%。

1 相关工作

虚拟机退出是每一个虚拟机软件和每一种虚拟机架构都会遇到的问题。VMWare[6] 使用扫描替换的技术,在不退出虚拟机的情况下用软件模拟特权指令,减少了虚拟机退出的次数。Xen[7] 也使用了类似的方法,通过修改软件的方式来减少特权指令的数量。通过硬件设计也可以减少虚拟机退出,其中 Intel 通过加入非根(non root)模式[8],使得大多数特权指令都可以直接在该模式下执行。MIPS 架构的硬件虚拟化支持[5]则加入了客户(Guest)模式,并加入了虚拟机专用的 CP0 寄存器。在 Guest 模式下可以执行大多数的特权指令,指令的执行受到虚拟机专用的 CP0 寄存器控制,而不受宿主机的 CP0 寄存器控制。在 Guest 模式下执行的读取或修改 CP0 寄存器的指令则直接作用于虚拟机专用的 CP0 寄存器。MIPS 架构的虚拟化支持减少了导致虚拟机退出的特权指令的数量,从而减少虚拟机退出的次数。

除了上述减少虚拟机退出次数的方法,还有一些技术可以降低每次虚拟机退出的开销。Intel 在处理器中加入了 VMCS 结构[8],于是每一个虚拟机有其对应的 VMCS 域,当虚拟机退出时,处理器会自动将现场保存到 VMCS 中对应的域,在重新回到虚拟态时,处理器还会自动进行恢复。然而,MIPS 架构的虚拟化规范[5]并没有相关的功能,虚拟机退出时,需要由软件显式地进行现场保存和恢复。本文针对该问题,提出了一种延迟存储的方法,来降低虚拟机退出的开销。该方法不需要硬件改动,因此不需要进行硬件验证与重新流片,也不违背 MIPS 的虚拟化规范。

2 方法的可行性分析

在 MIPS 虚拟化架构中,进入虚拟态模式与退出虚拟态模式时,需要由软件显式地进行现场的保存与恢复工作,其过程如下:

(1)在进入虚拟态模式时,保存当前宿主机的通用寄存器及浮点寄存器,并且恢复该虚拟机对应的通用寄存器、浮点寄存器及虚拟机专用寄存器。

(2)在退出虚拟态模式时,保存当前虚拟机的通用寄存器及浮点寄存器,并保存虚拟机专用寄存器,并且恢复宿主机的通用寄存器及浮点寄存器。

以 Linux 系统中常用的虚拟化平台 KVM[1] 为例,需要保存和恢复的寄存器如表 1 所示。

表 1 需要保存和恢复的寄存器个数

寄存器类型	个数
通用寄存器	34
浮点寄存器	16
虚拟机专用寄存器	21
其他	3
总计	66

表 1 中,通用寄存器与浮点寄存器的保存只需要一条存储指令,如 SD 或 SDC1 指令。但是,虚拟机专用寄存器保存时,需要先使用 MFGC0 指令将该寄存器的值取到通用寄存器之后,才能使用 SD 指令进行保存。同理,恢复现场时,恢复通用寄存器与浮点寄存器也只需要一条读取指令,如 LD 或 LDC1 指令,而恢复虚拟机专用寄存器时,需要先使用 LD 指令将被保存的值取到通用寄存器中,再使用 MTGC0 指令将其存储到虚拟机专用寄存器中。

按照 MIPS 虚拟化的设计规范,MTGC0 指令所操作的虚拟机专用寄存器没有寄存器重命名机制,必须等到该指令成为流水线中最老的指令之后才可以发射并执行。此外,与 MFGC0 或 MTGC0 指令配对使用的存储或读取指令还会导致寄存器数据相关。相比之下,用于保存和恢复通用寄存器或浮点寄存器的指令则是普通的存储指令或读取指令,这些指令可以全流水地在访存功能部件执行。从上述分析中可以发现,虚拟机专用寄存器的保存与恢复是虚拟机退出时所需要进行的现场保存与恢复工作中的重头戏,会对虚拟机的整体运行性能产生影响。

通过分析可以发现,虚拟机专用寄存器的值只

有在虚拟态模式下才能发挥作用,并不会对宿主机执行的指令产生任何影响。而宿主机除了使用特定的 MTGC0 指令之外,也无法修改虚拟机专用寄存器的值。因此,虚拟机退出时暂时不保存虚拟机专用寄存器的值不会导致错误,等虚拟机恢复运行,处理器切换回到虚拟态模式时,还可以继续运行。不过,在多个虚拟机同时在一台宿主机上运行的情况下,每一个虚拟机都有其对应的运行环境,各自的虚拟机专用寄存器的内容也不相同。但是硬件上只实现了一份虚拟机专用寄存器,因此在多个虚拟机之间进行切换时,还是要对虚拟机专用寄存器进行保存与恢复工作。

根据上面分析的结果,可以用下述方法来减少虚拟机专用寄存器的保存与恢复:

在虚拟机切换到宿主机的过程中,不进行虚拟机专用寄存器的保存,在宿主机切换到虚拟机的过程中则进行判断:如果与之前虚拟机退出时是同一个虚拟机,则不需要恢复虚拟机专用寄存器;如果是不同的虚拟机,则将虚拟机专用寄存器的值保存到之前退出的那个虚拟机的存储空间中,并从新虚拟机的存储空间中导入属于它的虚拟机专用寄存器的值。这种方法使得简单的虚拟机退出时不需要保存和恢复虚拟机专用寄存器,只有多个虚拟机进行切换时才需要保存和恢复。

在宿主机是多核处理器的情况下,该方法也可以适用,但是需要一些改动:虚拟机软件有可能在处于非虚拟态模式下时被操作系统调度到另一个处理器核上执行,此时需要在操作系统进行调度之前保存虚拟机专用寄存器,并在调度完成后进行恢复。

该方法的安全性需要两方面来保证,一方面是宿主机不会在虚拟机退出后使用 MTGC0 指令来修改另一个虚拟机的专用寄存器,另一方面是程序可以正确区分是否进行了虚拟机切换,以及多核处理器情况下准确地识别出是否发生了进程调度。前者可以通过检查虚拟机软件的源代码来保证,因为从虚拟机的功能来说,没有修改非当前激活的虚拟机的需求,宿主机并不会主动去修改虚拟机专用寄存器的值。而后者的要求则可以通过判断虚拟机的虚拟机号,并给每一个虚拟机分配一个表示当前宿主机处理器核序号的变量来实现。因此,延迟存储方法的安全性可以保证。

3 方法的实现方案

通过上面的分析,可以设计一套延迟存储的方法,该方法需要对虚拟机软件进行一定的修改:

(1)新增两类变量,其中第一类变量是每个处理器核各有一个的全局变量,表示该处理器核上次运行的虚拟机的序号;第二类变量则是每一个虚拟机的局部变量,表示该虚拟机上次运行的宿主机处理器核的序号。这两类变量的初始值都可以设置为 -1,表示虚拟机还未被运行过的情况。

(2)虚拟机退出时,不再需要进行虚拟机专用寄存器的保存工作。但是,通用寄存器和浮点寄存器等还是需要进行保存,并恢复宿主机的寄存器备份。

(3)从宿主机回到虚拟态的虚拟机恢复运行过程需要进行修改。回到虚拟态模式时,应当检查即将被运行的虚拟机的序号是否与当前处理器核上次运行的虚拟机序号相同,然后再检查当前处理器核的序号是否与这个即将被运行的虚拟机所记录的上次运行的处理器序号相同。如果两次检查都没有发现变化,则不需要进行虚拟机专用寄存器的恢复,仅需要处理通用寄存器等其他寄存器;若两次检查中任意一次发现了序号变化,则需要将当前的虚拟机专用变量保存在之前运行的虚拟机的存储空间中,并从即将运行的虚拟机的存储空间中恢复虚拟机专用变量,并进行通用寄存器等其他寄存器的保存与恢复工作。

(4)操作系统进行进程调度时,需要额外的虚拟机专用的处理过程。该处理过程会根据指示虚拟机序号的全局变量,将虚拟机专用变量保存在之前运行的虚拟机的存储空间中,并将上述全局变量改为 -1,表示当前处理器核中的虚拟机专用变量已经过时了。

经过上述修改后,虚拟机进入与退出的处理流程图如图 1 所示。

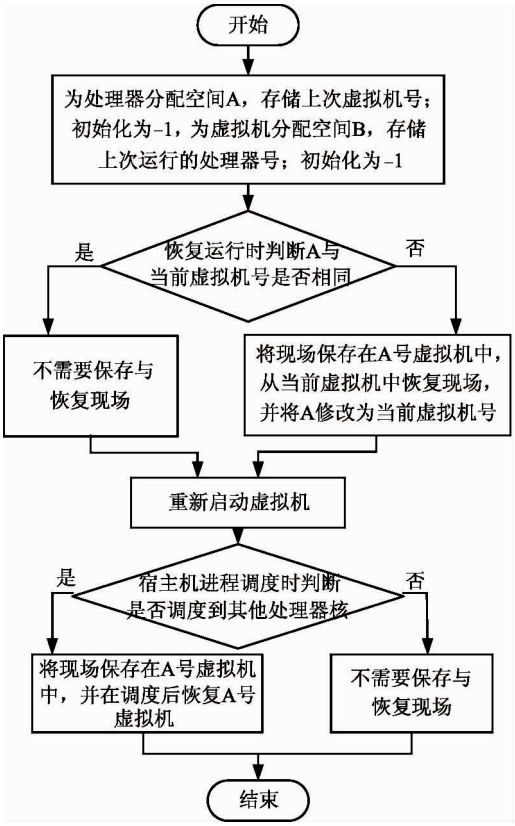

图 1　虚拟机退出与恢复的流程图

通过上述修改，就可以实现本文所提出的降低虚拟机退出开销的延迟存储方法。可以预计，其效果在单处理器核上运行单个虚拟机的情况尤为显著。在多核操作系统的情况下，则可以通过绑定程序运行处理器号的方式，如 taskset 命令，将多个虚拟机分别绑定在特定的处理器核上，使得每一个处理器核只运行一个特定虚拟机，以此达到与单个处理器核上运行单虚拟机情况的等同效果。

4　试验平台和试验结果

本文在虚拟化平台 KVM 中实现了所提出的延迟储存方法，并在多个平台上进行了试验，包括龙芯 3A1500 处理器的 RTL 平台、EVE 仿真加速器[9]平台，以及 QEMU 模拟器平台[10]。不过受限于试验平台对多核处理器模式的支持不足，因此只测试了单个处理器核运行单个虚拟机的情况。但是在理论上，当同时运行的虚拟机数量不超过处理器核数目时，可以通过程序绑定处理器核运行的方式得到近似的性能结果。下面分别介绍各个试验平台及其试验结果。

龙芯 3A1500 处理器是龙芯 3 号处理器系列的最新产品，在 2014 年三季度进行了流片，最多支持 4 个处理器核，是一款四发射的超标量、动态流水线、支持乱序执行的高性能处理器，拥有两个定点运算部件、两个浮点运算部件以及两个访存功能部件，设计有 Guest 模式等多种虚拟化辅助功能。在龙芯 3A1500 的 RTL 平台上，对虚拟机退出及恢复运行时所需要的保存与恢复现场的指令序列进行了模拟，结果如表 2 所示。可以看出，在单处理器核单虚拟机情况下，使用延迟存储方法可以降低虚拟机退出开销 24%，降低恢复运行时的开销 77%，总开销降低 65%。

表 2　虚拟机退出指令所需周期数

操作	优化前	优化后
虚拟机退出	83	62
虚拟机恢复	272	62

QEMU[10]是一款跨平台模拟器，可以运行完整的操作系统和应用程序，并可以在一个架构上高效地模拟另一种架构的处理器。QEMU 作为 KVM 解决方案的一个重要部分，在系统虚拟化领域中得到了广泛的使用。试验中使用的 QEMU 模拟器的版本为 0.14.0，运行的宿主机操作系统和虚拟机操作系统都是 Linux，内核版本都为 2.6.36。虚拟机中运行的应用程序为 SPEC CINT2000[11]测试集。试验结果如图 2 所示，应用程序的运行时间减少了9.6%。

EVE 仿真加速器[9]是一款基于 FPGA 的仿真加速器平台，可以周期级精确地实现处理器的功能，并提供有调试接口和验证工具集成环境，是处理器流片前验证的重要工具。该平台运行的处理器为龙芯 3A1500 处理器，操作系统为 Linux，内核版本 2.6.36。在 EVE 平台上，测试了虚拟机内核的完整启动过程。这个启动过程被划分为 6 个阶段，从图 3 中可以看到应用延迟存储方法前后的性能对比。Linux 内核的完整启动过程速度提升了2.8%，其中的一个阶段获得了8.5%的提升。

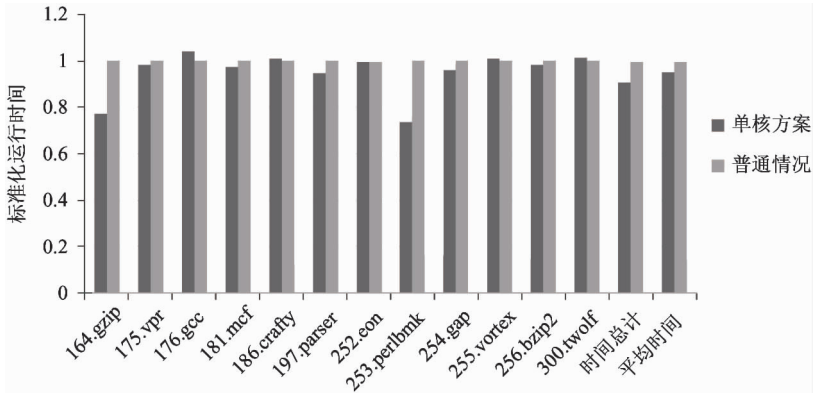

图 2　标准化的 SPEC 程序运行时间

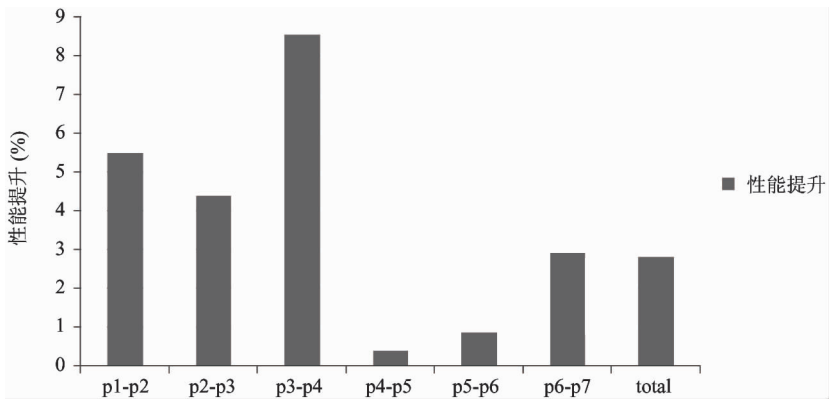

图 3　Linux 内核启动性能提升

在上述三种平台的试验结果表明,虚拟机退出所消耗的时间是整个虚拟机性能中的一个重要组成部分,而本文中提出的延迟存储方法可以降低虚拟机退出及恢复运行时所需的时间,降低效果在单处理器核单虚拟机的情况下尤为明显。虚拟机整体性能获得的提升幅度则与虚拟机退出的数量有关。运行 SPEC 测试程序获得的性能提升比启动 Linux 内核时要大,是因为龙芯 3A1500 处理器实现了 MIPS 虚拟化架构所需要的硬件支持,大部分特权指令都可以在虚拟态下直接执行,反而是内存缺页请求会导致虚拟机退出,因此越大规模的程序也会导致越多的虚拟机退出。虚拟机退出更频繁的情况下,延迟存储方法对性能的提升也越大,总体而言,本文提出的方法可以对虚拟机的运行带来约 6% 的性能提升。

5　结　论

虚拟机退出的开销是系统虚拟化性能开销的重要部分,在硬件虚拟化辅助设计较少的处理器架构中,虚拟态与正常运行状态之间的切换更是占用了大量时间。目前业界有多种方法来减少虚拟机的退出次数,但是并没有用于降低虚拟机退出时间的软件方法。本文针对虚拟机退出代价的问题,提出了一种延时存储的方法,该方法不需要进行硬件改动,通过减少虚拟机退出和恢复运行时所需的现场保存工作,降低了虚拟机退出的开销。通过在国产龙芯 3A1500 处理器平台上的试验可以看到,即使在拥有完备虚拟化硬件辅助设计的处理器中,虚拟机的运行性能依然可以获得可观的提升。本文提出的延迟存储方法在虚拟机研究领域具有广泛的适用性,是一种有效的提升虚拟机性能的手段。

延迟存储方法并没有完全消除虚拟机退出时需要的保存现场工作,定点通用寄存器与浮点寄存器还是需要进行切换。通过同时使用其他方法,如利用影子寄存器[12]来进行通用寄存器切换,可以进一步降低虚拟机退出对虚拟机运行性能的影响。上述

工作以及通过减少虚拟机退出次数来提高虚拟机性能的工作是下一步研究的方向。

参考文献

[1] Kivity A, Kamay Y, Laor D, et al. KVM: the linux virtual machine monitor. In: Proceedings of the Ottawa Linux Symposium, Ottawa, Canada, 2007. 225-230

[2] Popek G J, Goldberg R P. Formal requirements for virtualizable third generation architectures. *Comm ACM*, 1974, 17:412-421

[3] MIPS Technologies Inc. MIPS64 architecture reference manual volume IV-i, 2010

[4] MIPS Technologies Inc. MIPS32 architecture for programmers volume IV-i: virtualization module of the MIPS32 architecture, 2013

[5] MIPS Technologies Inc. MIPS64 architecture for programmers volume IV-i: virtualization module of the MIPS64 architecture, 2013

[6] Waldspurger C A. Memory resource management in VMware ESX server. In: Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Boston, USA, 2002. 181-194

[7] Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 2003, 37: 164-177

[8] Uhlig R, Neger G, Rodger D, et al. Intel Virtualization Technology. *Computer*, 2005, 38: 48-56

[9] Synopsys. EVE. http://www.eve-team.com:Synopsys, 2010

[10] Bellard F. Qemu, a fast and portable dynamic translator. In: Proceedings of the USENIX 2005 Annual Technical Conference, Anaheim, USA, 2005. 41-46

[11] Standard Performance Evaluation Corporation. SPEC CPU2000. http://www.spec.org/cpu2000: Standard Performance Evaluation Corporation, 2000

[12] Nethercote N, Seward J. Valgrind: a framework for heavyweight dynamic binary instrumentation. *ACM Sigplan Notices*, 2007, 42(6):89-100

Delay storing: a method to reduce the cost of virtual machine exit

Wu Ruiyang[*,**,***], Tai Yunfang[****]

([*] Key Laboratory of Computer Architecture, Institute of Cumputing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] University of Chinese Academy of Sciences, Beijing 100049)

([***] Loongson Technology Corporation Limited, Beijing 100190)

([****] Oracle Asia Research and Development Center (Beijing), Beijing 100193)

Abstract

The cost of virtual machines' exit and restoration was studies, and a method based on delay storing was proposed to reduce the cost of saving and restoring registers when virtual machines exit or resume. The main mechanism of the method is to reduce the amount of registers to be saved and restored by changing the source code of the virtual machine software and judging whether the virtual machine in resuming is still the same one that exited last time. The proposed method needs no hardware change, and supports multicore operating systems and concurrent operation of multiple virtual machines, leading to a wide applicability. The results of the test conducted on the Loongson 3A1500 platform demonstrated that the cost of virtual machine exit of the proposed method was reduced by 65% compared with the existing method, and the performance of the whole virtual machine was increase by 3% to 10%.

Key words: system-level virtualization, guest mode, virtual machine exit, delay storing