软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(Suppl.(2)):127−139

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

龙芯 3 号处理器多核虚拟化技术*

阮 利[1,2], 徐 鹏[1,2], 王慧祥[1,2], 祝明发[1,2], 肖利民[1,2], 唐浩夫[1,2]

[1](软件开发环境国家重点实验室(北京航空航天大学),北京 100191)

[2](北京航空航天大学 计算机学院,北京 100191)

通讯作者: 阮利, E-mail: ruanli@buaa.edu.cn, http://scse.buaa.edu.cn/

摘 要: MIPS 处理器是精简指令集(RISC)处理器中的一个重要代表,通常应用于嵌入式系统中.近年来,随着MIPS处理器性能的大幅度提升,其应用渐渐扩展到了高性能服务器领域.龙芯3号处理器是MIPS架构的典型代表.在目前的服务器研究领域中,多核技术是一项重要的技术指标,而虚拟化技术是另一项重要的技术指标.当前,虽然虚拟化技术得到了快速发展,但是龙芯 3 号处理器上的虚拟化技术却鲜有成果.基于龙芯 3 号处理器的多核虚拟化技术面临许多问题,虚拟多核架构结构复杂、核间通信方式难以模拟等都会为龙芯3号处理器上的多核虚拟化带来困难.分析了多核龙芯 3 号处理器的硬件结构以及物理多核的核间中断通信方式,在此基础上介绍了龙芯 3 号处理器上多核虚拟化关键技术.主要在多核处理器虚拟化总体架构设计、虚拟多核结构设计以及虚拟多核的核间通信方式等方面进行了讨论.实验的结果表明,在龙芯3号处理器上,该多核虚拟化方法具有良好的效果.

关键词: MIPS;龙芯 3 号;虚拟化;多核;中断通信

中文引用格式: 阮利,徐鹏,王慧祥,祝明发,肖利民,唐浩夫.龙芯 3 号处理器多核虚拟化技术.软件学报,2013,24(Suppl.(2)):127−139. http://www.jos.org.cn/1000-9825/13031.htm

英文引用格式: Ruan L, Xu P, Wang HX, Zhu MF, Xiao LM, Tang HF. Multi-Core virtualization on Loongson-3 processor. Ruan Jian Xue Bao/Journal of Software, 2013,24(Suppl.(2)):127−139 (in Chinese). http://www.jos.org.cn/1000-9825/13031.htm

Multi-Core Virtualization on Loongson-3 Processor

RUAN Li[1,2], XU Peng[1,2], WANG Hui-Xiang[1,2], ZHU Ming-Fa[1,2], XIAO Li-Min[1,2], TANG Hao-Fu[1,2]

[1](State Key Laboratory of Software Development Environment (BeiHang University), Beijing 100191, China)

[2](School of Computer Science and Engineering, BeiHang University, Beijing 100191, China)

Corresponding author: RUAN Li, E-mail: ruanli@buaa.edu.cn, http://scse.buaa.edu.cn/

Abstract: MIPS architecture is an important member of RISC processor family, and it is mainly applied in embedded systems. With its performance improvement, the MIPS processor is gradually used in the field of high-performance servers. The Loongson-3 processor is a typical representative of the MIPS architecture. As a main feature of high-performance server, the multi-core architecture is indispensible. Meanwhile, virtualization technology is another important application for the server, however in recent years the technology rarely has success on the Loongson-3 processor. Combining virtualization technology and multi-core technology on the Loongson-3 processor has much more difficulties as the method to simulate the multi-core architecture is hard and collaborative mechanism among different cores is complex. This paper analyzes the multi-core architecture and communication mechanisms of the multi-core architecture Loongson-3 processor and introduces a virtualization method for the multi-core architecture Loongson-3 processor. It mainly discusses the overall design of virtualization method of the multi-core Loongson-3 processor, the simulation of Loongson-3 multi-core architecture and the virtual inter-processor interrupt communication in the virtual machine. Experimental results show that the presented method provides useful and efficient multi-core virtualization support for the Loongson-3 processor.

* 基金项目: 国家自然科学基金(61003015, 61232009); 国家高技术研究发展计划(863)(2011AA01A205); 教育部博士点专项基金(2010110 2110018); 北京市自然科学基金(4122042); 软件开发环境国家重点实验室自主研究课题(SKLSDE-2012ZX-23)

收稿时间: 2012-08-05; 定稿时间: 2013-07-22

Key words: MIPS; Loongson-3; virtualization; multi-core; interrupt communication

作为精简指令集处理器的代表,MIPS 指令集架构以其高性能、低功耗的特点被广泛应用于嵌入式处理器领域.龙芯 3 处理器是 MIPS 指令集架构的一个典型代表.近年来,随着其性能的不断提升,龙芯 3 处理器逐渐被应用到服务器领域.然而,在服务器研究领域如何提高服务性能和硬件资源利用率成为当今的一个热点问题.目前最重要的两个研究方向分别是多核技术和虚拟化技术.

多核技术是指在一枚处理器(chip)中集成两个或多个完整的计算引擎(内核)[1].随着处理器的发展,硬件工程师发现仅仅提高单核芯片的速度会产生过多的热量并且无法带来相应的性能改善.此外,即便是没有热量问题,其性价比也难以令人接受,速度稍快的处理器价格会高出很多.英特尔的工程师们开发了多核芯片,使其满足横向扩展方法,从而提高处理器性能.多核技术实现了分治法策略[2].得益于线程技术的应用在多核处理器上运行时将显示出卓越的性能可扩充性.多核技术能够使服务器并行处理任务,此前则可能需要使用多个处理器.多核系统更易于扩充,并且能够在更纤巧的外形中融入更强大的处理性能.这种外形所用的功耗更低、计算功耗产生的热量更少.

虚拟化技术是当今工业界和学术界的另一个研究热点.许多重要的应用程序都是基于虚拟化技术,其中最具代表性的是云计算技术.虚拟化技术可以显著地提高硬件资源利用率[3],此外,该技术可以有效地将系统隔离开来,从而提高系统安全性[4].如今,X86 架构下的虚拟化技术发展迅速,而基于 MIPS 架构处理器的虚拟化相关研究较少.北京航空航天大学龙芯处理器虚拟化课题组前期已展开龙芯 3 服务器上多核虚拟化的研究[5]和虚拟机镜像存储优化等方面的研究[6].基于现有的 MIPS 虚拟化技术以及 MIPS 处理器多核架构的特点,我们提出了一种龙芯 3 处理器多核虚拟化的方法,并且该方法在龙芯 3 号多核处理器上表现出了良好的效果.

本文第 1 节介绍实现龙芯 3 处理器多核虚拟化所面临的主要挑战.第 2 节介绍国内外相关工作现状.第 3 节介绍龙芯 3 处理器多核虚拟化的总体设计.第 4 节介绍基于全虚拟化的多核架构模拟方法.第 5 节介绍基于陷入模拟的核间通信.第 6 节对我们实现的多核虚拟化系统进行测试.

1 面临的挑战

由于龙芯 3 处理器多核架构的复杂性及其存在的虚拟化漏洞,实现龙芯 3 处理器的多核虚拟化是非常困难的.具体原因如下:

1.1 MIPS处理器存在虚拟化漏洞

龙芯 3 处理器的虚拟地址空间不满足 Popek 和 Goldberg 在 1974 年提出的可虚拟化理论:虚拟地址空间必须支持客户机操作系统(GOS)和宿主机操作系统(HOS)运行在不同特权级的地址空间[7].此外,MIPS 架构的地址翻译机制也造成了 GOS 和 HOS 的多核寄存器将会被映射到相同的物理地址.更要重要的一点是,龙芯 3 处理器目前还不支持硬件辅助虚拟化[4,7,8].因此,龙芯 3 处理器在虚拟地址空间方面存在的虚拟化漏洞使得在龙芯 3 处理器上实现多核虚拟化非常困难.

1.2 复杂的龙芯3多核架构

为了实现多核虚拟化,龙芯 3 处理器为每个核提供 8 个寄存器,用于核间中断和通信[9].这 8 个寄存器的位数并不相同,有些是 32 位,有些是 64 位,这给多核虚拟化带来了困难.此外,每个核间中断(IPI)寄存器的物理地址是固定的,这就使得客户机和物理机核间中断寄存器的物理地址相互冲突.为了避免这种地址冲突,我们需要设计一种新的策略.

1.3 难以保证多个核的协同工作

因为多核系统在启动和运行期间都要考虑多个核的调度和协作问题,所以,如何模拟物理多核之间的协同工作机制就成为一个难题.除此之外,多核处理器中的核间通信和中断机制非常复杂,但这都需要我们对其进行模拟.

2 国内外相关工作

龙芯 3 处理器的多核虚拟化主要涉及 MIPS 架构的虚拟化和龙芯 3 的多核架构.所以,本节我们将从 MIPS 的虚拟机、龙芯 3 的多核架构以及多核架构的虚拟化这 3 个方面来展开介绍.

2.1 基于MIPS的虚拟化技术

龙芯 3 处理器是基于 MIPS 架构的多核处理器,能够完全兼容 MIPS 64 R2 版本的所有指令.在 MIPS 虚拟化方面,比较有代表性的包括 1997 年斯坦福大学所研制的基于 MIPS R10000 处理器的虚拟化系统 DISCO[10],以及 Open Kernel Labs 所研制的开源系统软件平台 OKL4[11].为了解决当时流行的大型共享内存多核处理器系统的高效运行问题,斯坦福大学利用在 20 世纪 70 年代十分流行的虚拟机监控器的概念,在一个可扩展的 MIPS R10000 多处理器系统上实现了一个能够运行商用操作系统的系统虚拟机,并且该虚拟机的物理资源利用率较高.然而,DISCO 需要进行大量的设备驱动开发,工作量极大.所以,系统只停留在了实验室阶段,并未成为一个成功的商用系统.OK Labs 公司(Open Kernel Labs)基于 L4 微内核系统开发了一款名为“OKL4”的微内核虚拟机监控器系统.虽然基于半虚拟化思想的 OKL4 微内核虚拟机监控器可以很好地避免 MIPS 指令集不可虚拟化的“漏洞”以及 MIPS 地址空间不支持虚拟化的设计缺陷,但是半虚拟化所带来的巨大工作量给虚拟化开发带来了很大的困难.除了以上两种技术以外,还有一种全虚拟化方案 QEMU[10]也支持 MIPS 架构.但是,QEMU 效率很低,并不符合服务器的要求.

2.2 龙芯3处理器的多核架构

龙芯 3 服务器一个物理 CPU 芯片中存在 4~8 个处理器核,每个处理器核按照固定的硬件地址空间进行寻址访问.以四核的龙芯 3 处理器为例,每个处理器核的地址见表 1.

Table 1 Register address

表 1 IPI 寄存器地址

处理器核	寄存器地址
Core 0	0x_3FF0_1000
Core 1	0x_3FF0_1100
Core 2	0x_3FF0_1200
Core 3	0x_3FF0_1300

龙芯 3 处理器为每个处理器核提供 8 个寄存器用于系统启动和运行时的核间中断和通信.这些寄存器被称为 IPI 寄存器.8 个寄存器具有不同的位数以及不同的功能,见表 2.其中,4 个缓存寄存器 IPI_MailBox0-3 用于供启动时传递参数使用,按照 64 或者 32 位的非缓存方式进行;另外 4 个寄存器用于控制核间中断的状态.

Table 2 Inter-Cores interrupt registers

表 2 核间中断寄存器

核间中断寄存器	访问权限	位宽(bit)	偏移地址	功能
IPI_Status	读	32	0x_00	状态寄存器
IPI_Enable	读/写	32	0x_04	使能寄存器
IPI_Set	写	32	0x_08	置位寄存器
IPI_Clear	写	32	0x_0c	清除寄存器
IPI_MailBox0	读/写	64	0x_20	缓存寄存器 0
IPI_MailBox1	读/写	64	0x_28	缓存寄存器 1
IPI_MailBox2	读/写	64	0x_30	缓存寄存器 2
IPI_MailBox3	读/写	64	0x_38	缓存寄存器 3

与处理器核固定的偏移地址类似,IPI 寄存器相对于处理器基址的偏移量也是固定的,计算公式如下:

$$IPI\_Register\_Addr = Processor\_Addr + Core\_Offset + IPI\_Register\_Offset \quad (1)$$

其中, $IPI\_Register\_Addr$ 代表一个 IPI 寄存器, $Processor\_Addr$ 代表一个处理器的基址, $Core\_Offset$ 代表处理器核相对于处理器基址的固定偏移量, $IPI\_Register\_Offset$ 是指 IPI 寄存器相对于它所在的核的固定偏移量.

其他节点、处理器核以及 IPI 寄存器也是以相同的方法计算得出的[9].

2.3 多核架构的虚拟化

服务器上的虚拟化技术需要对服务器的多核架构进行支持.不同的虚拟化方案针对各个体系结构的多核架构有着不同的虚拟化处理方式.在多核硬件结构方面,各种硬件架构的硬件多核原理相似,各个处理器核有相应的寄存器维护处理器核的状态,而各个处理器核的寄存器具有固定的物理地址以便于寻址访问.针对各种不同处理器的硬件架构,各种虚拟化技术对其多核具有不同的处理方式.表 3 中列出的主流虚拟化产品及技术对各种主流的硬件体系结构的多核架构进行了虚拟化的支持.

Table 3 Multi-Core processor virtualization comparison[12−16]

表 3 多核处理器虚拟化对比[12−16]

多核架构	VMware	KVM	Xen	Microsoft
X86	支持	支持	支持	支持
IA64	支持	支持	支持	支持
MIPS	不支持	不支持	不支持	不支持
PowerPC	支持	支持	不支持	不支持
ARM	支持	支持	支持	不支持
SPARC	不支持	支持	支持	不支持

当前的各种主流虚拟化方案对各种多核硬件架构有不同程度的支持.其中,MIPS 是唯一在多核虚拟化方面存在欠缺的硬件处理器,而以 MIPS 架构为基础的龙芯 3 服务器同样面临此问题.此外,在服务器领域过于低的效率是不切合实际的.针对多核架构的发展状况以及龙芯 3 处理器多核的特点,我们设计了龙芯 3 处理器多核结构的模拟方法和虚拟的核间通信机制,并提出了龙芯 3 处理器多核虚拟化的方法.

3 龙芯 3 服务器多核虚拟化系统总体设计

3.1 设计思路

基于中国科学院计算技术研究所与北京航空航天大学龙芯处理器虚拟化课题的前期工作展开龙芯 3 服务器上多核虚拟化的研究,在 KVM/Loongson-3A 虚拟化系统的基础上增加虚拟多核特性的支持.主要工作体现在以下几个方面:QEMU 多核结构模拟、QEMU 多核核间中断注册、虚拟机监控器(VMM)内核模块多核结构模拟、VMM 核间中断处理.本文将从这几个方面针对龙芯 3 服务器的多核架构特点对 KVM/Loongson-3A 进行改进,以实现龙芯 3 服务器上的多核虚拟化系统.

3.2 模块架构

本文提出的龙芯 3 服务器多核虚拟化架构基于 KVM 实现,采用宿主模型架构[4],其中主要包括内核模块与宿主应用 QEMU 进程两个部分.系统总体架构如图 1 所示.

如图所示,多核虚拟化主要包括 QEMU 中虚拟多核设备的模拟和 VMM 中的虚拟核间通信机制.

3.3 工作流程

龙芯 3 服务器上多核虚拟机的工作流程由客户机操作系统、宿主机 VMM 内核模块以及宿主机 QEMU 进程 3 个部分协调完成.其中,客户机操作系统部分主要完成客户机的正常运行功能;宿主机 VMM 内核模块主要完成客户机环境的准备、多核异常处理等工作;宿主机 QEMU 进程的主要工作是初始化虚拟多核结构并完成核间中断访问的模拟.多核虚拟化系统工作流程如图 2 所示.

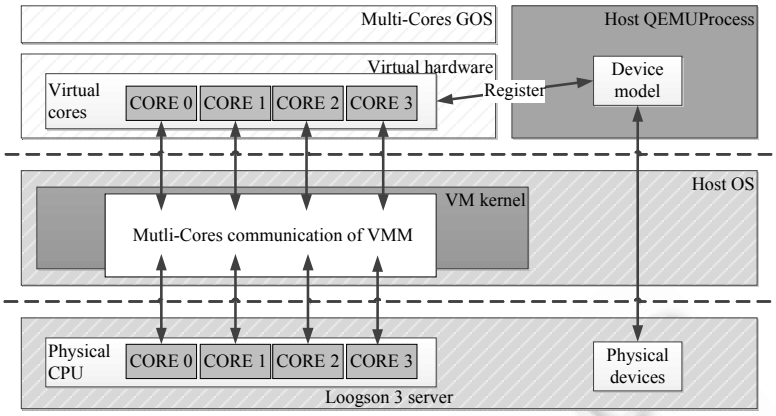

Fig.1 Multi-Core virtualization architecture of Loongson-3 server

图 1 龙芯 3 服务器多核虚拟化模块架构

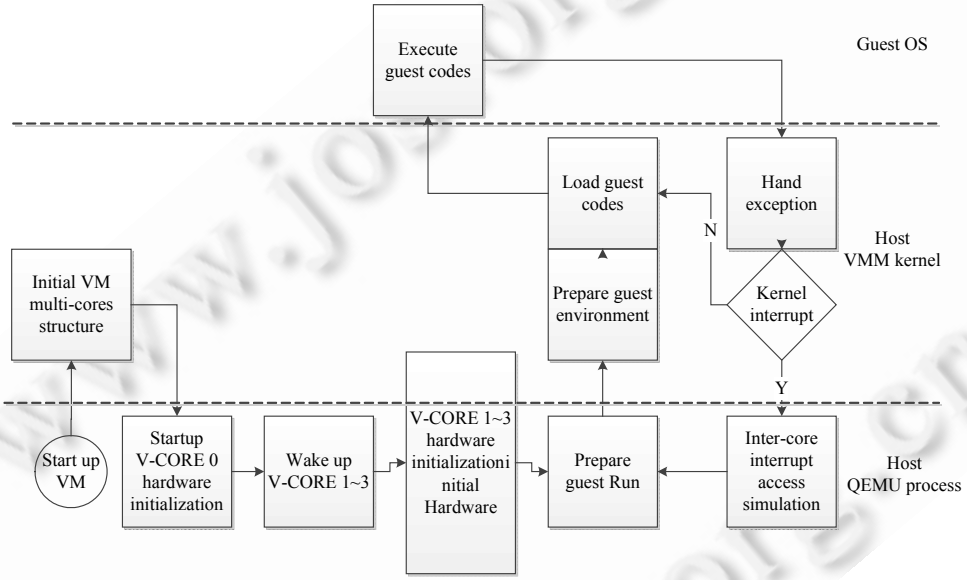

Fig.2 Workflow of Loongson-3 multi-core virtual machine

图 2 龙芯 3 多核虚拟机工作流程

多核虚拟化系统总体工作流程如下:

1) 宿主机 QEMU 进程运行,启动虚拟机,QEMU 通过 IOCTL 系统调用.

2) 宿主机 VMM 内核模块创建虚拟机相关结构并进行相应的初始化,然后退出到宿主机的 QEMU 进程.

3) 在宿主 QEMU 进程中的虚拟机启动第 1 个虚拟处理器核,并进行一些初始化工作,然后唤醒其他虚拟处理器核.

4) 所有的虚拟处理器核启动并初始化完毕后,再次调用 IOCTL 系统调用进入宿主机 VMM 内核模块,为客户机代码准备客户机运行环境.

5) 载入客户机代码,基于 KVM 的模型的客户机指令运行于宿主机上.

6) 在客户机代码运行过程中,当产生异常时,客户机陷入到宿主机的 VMM 内核模块以处理此异常.在多核虚拟化中,最重要的是 VMM 内核模块需要判断此异常是否属于核间中断.

7) 异常处理完成后,继续返回到客户机环境执行客户机代码.

VMM 中的虚拟核间中断处理是龙芯 3 处理器多核虚拟化的核心.

4 基于全虚拟化的多核架构模拟方法研究

基于全虚拟化的多核架构模拟方法主要从 3 个方面进行研究:虚拟多核结构模拟、虚拟核间中断寄存器地址空间注册、龙芯 3 服务器虚拟多核启动方式模拟.

4.1 虚拟多核结构模拟

龙芯 3 服务器处理器的硬件多核结构主要由核间中断寄存器组成,在虚拟化过程中,需要向客户机提供与龙芯 3 服务器硬件上完全一致的虚拟核间中断寄存器,以满足核间中断以及核间通信的基本要求.按照全虚拟化的设计思想,需要完全保证虚拟核间中断寄存器与硬件核间中断寄存器的一致性,以避免修改客户机操作系统.另一方面,宿主机内核模块 VMM 在进行虚拟机创建的过程中,需要将一个虚拟 CPU 结构中的处理器核结构扩展为多份,以对应多个物理处理器核,相应的虚拟 TLB 结构也需要进行扩展.

4.2 虚拟核间中断寄存器地址空间注册

MIPS 处理器在设计时,并未考虑到虚拟化的应用场景,其内核所在的虚拟地址空间被设计为不经过页表映射的地址区域.因此,在此区间内的内核地址空间的虚拟地址是与物理地址直接映射的,而多核架构的核间中断寄存器存在于这段不经映射的虚拟地址空间中.在虚拟化的应用场景下,宿主机的多核核间中断寄存器被映射到一块固定的物理地址上,而客户机的虚拟多核核间中断寄存器也会被映射到同一块物理地址上,从而导致客户机与宿主机核间中断寄存器地址冲突,基于 MIPS 架构的龙芯 3 处理器同样存在着这样的问题.为了解决 MIPS 架构的内存地址空间不可虚拟化问题,需要对客户机操作系统进行修改,使得客户机的虚拟核间中断寄存器被映射到与宿主机核间中断寄存器不同的物理地址空间中.图 3 是本文所采取的修改策略.其中,Xkphys[17]为宿主机内核所在的非映射区间,Xsseg[17]为客户机内核所在的可映射区间.因为客户机内核需要一定的运行权限,故选择 MIPS 架构中具有管理态权限的 Xsseg 区间.

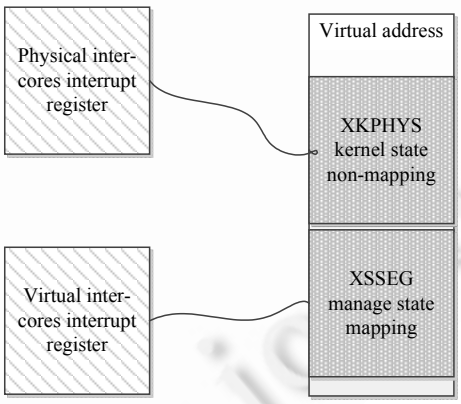

Fig.3 Address space design of inter-core interrupt register

图 3 核间中断寄存器地址空间设计

客户机的启动与运行均由宿主 QEMU 进程来控制,在启动虚拟机的初始化阶段,宿主 QEMU 进程需根据虚拟机分配的内存在宿主机上进行申请,并维护一段 QEMU 地址注册空间.QEMU 地址注册空间分为 BIOS 段、RAM 段、I/O 段以及其他用途的段,其中,BIOS 段用于虚拟机启动时内核的引导与加载,RAM 段为虚拟机内存映射区间,I/O 段在龙芯 3 服务器上以 MMIO 的形式组织起来.本文将虚拟核间中断寄存器的地址注册到 QEMU 注册地址的 I/O 段,并以 I/O 访问的模拟方式对虚拟核间中断寄存器的读写进行模拟,注册方式如图 4 所示.

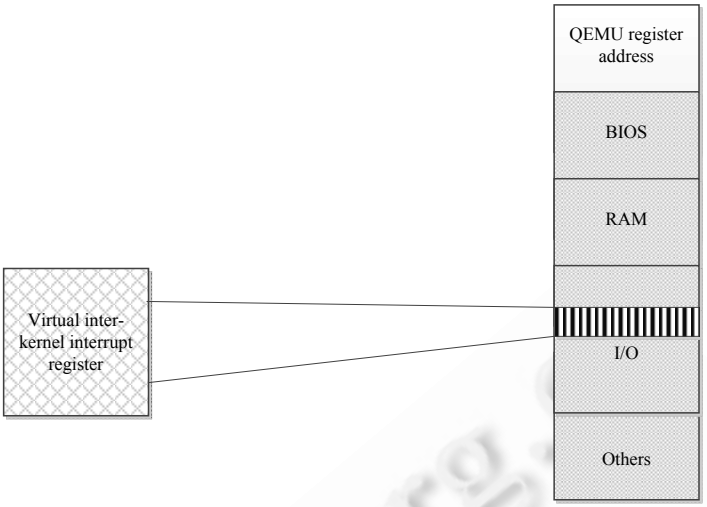

Fig.4 Address space register of QEMU inter-core interrupt register

图 4 QEMU 核间中断寄存器地址空间注册

将虚拟核间中断寄存器注册到 QEMU 的 I/O 地址空间,可复用 VMM 模块中已有的 MMIO 异常处理程序,简化核间中断的设计.

4.3 多核启动方式模拟

在龙芯 3 服务器物理硬件中,处理器多核启动方式如下[18]:

(1) 硬件加电,使各个处理器核出于准备状态.

(2) 指定一个核作为系统的主核(一般为编号为 0 和处理器核),读取 BIOS 的中的第 1 条指令,完成对系统内核的加载与引导.

(3) 主核完成系统内核的引导以及部分硬件初始化工作后,向其他几个处理器核(称为副核)发送启动信号.

(4) 副核接收到主核发送的启动信号后,以核间通信的方式从核间中断寄存器中读取第 1 条需要运行的指令的地址,并开始副核相应的初始化工作.

(5) 主核与副核协同工作,完成系统的启动.

经过上述的启动过程,多核处理器可以正常地启动并运行操作系统.在龙芯 3 服务器的多核虚拟化工作中,也面临着虚拟多核启动的问题.其中最主要的工作是指定虚拟主核并由虚拟主核发送信号触发虚拟副核的启动.对此,本文对龙芯 3 虚拟化多核启动设计思路如下:

1) 将虚拟副核的注册地址通知虚拟主核,以便虚拟主核为虚拟副核发送启动信号.

2) 各个虚拟核的核间中断寄存器 MailBox 被初始化为 0.

3) 当客户机系统启动时,虚拟主核进行虚拟硬件的初始化工作,同时虚拟副核循环读取各自的虚拟核间中断寄存器,若 MailBox 为 0,则继续循环读取;若 MailBox 不为 0,则表示此 MailBox 中的值为虚拟副核需执行的第 1 条指令地址,此时虚拟副核跳转至相应的地址运行.

4) 主核完成系统内核的引导以及部分虚拟硬件的初始化工作后,向各个虚拟副核的 MailBox 中写入非零的指令地址.

5) 各个虚拟副核读取 MailBox 中的指令地址,并跳转至相应的地址进行处理.

5 基于陷入模拟的核间通信方法研究

在龙芯 3 服务器上多核的工作是以多核核间通信的方式协调进行的.核间通信基于核间中断的触发,通过

设置核间中断 Status 等寄存器来对核间通信进行控制,并通过读写核间中断缓存寄存器 MailBox 以达到多核间通信的目的.所有的核间通信寄存器以“基地址+偏移”的方式进行寻址.

5.1 多核虚拟化核间中断模拟

由于虚拟机的核间通信需要出发核间中断来保持一致性,所以宿主机 VMM 内核模块需要对核间中断进行处理,以保证虚拟核间通信的顺利进行.在龙芯 3 服务器上多核虚拟化核间中断设计工作中,几个关键点如下:

1) 虚拟多核核间中断的捕获

通过第 4.2 节中将虚拟核间中断寄存器地址注册到 QEMU 的 I/O 地址空间中,可以将虚拟核间中断寄存器的读写视为 MMIO[17]的一种,以 I/O 的方式进行处理.MMIO 异常作为 TLB MISS 异常的一种,在宿主内核模块 VMM 中统一由 TLB 缺失异常处理程序进行处理.由此,本文对虚拟核间中断异常的处理可简化为对 TLB 缺失异常处理程序的修改.对虚拟多核核间中断的捕获有如下几个关键点:

a) 复用 TLB 缺失异常处理:龙芯 3 上多核虚拟化的核间中断,并未在虚拟机的异常向量表中专门增加核间中断处理函数,而是复用了 TLB 缺失异常处理.

b) 将核间通信视为 MMIO:利用虚拟多核核间中断寄存器在虚拟机中注册的地址,将虚拟核间中断寄存器的读写视为 MMIO 的一种.

c) 核间通信需陷入 VMM:虚拟机读写虚拟核间中断寄存器,可以 TLB 缺失的方式陷入 VMM 中进行处理.

d) 修改 TLB 缺失异常:在 VMM 的 TLB 缺失异常处理中,增加对虚拟核间中断寄存器注册地址的特殊判断.

2) 虚拟多核核间中断的处理

由于虚拟核间中断异常被视为 MMIO 进行处理,所以在产生虚拟核间中断时,被判断为核间中断的 TLB 缺失异常将以 MMIO 的方式模拟相应的读写指令,并通过 QEMU 的外设模拟方式读写虚拟核间中断寄存器.

5.2 多核虚拟化核间中断寄存器寻址访问模拟

通过第 5.1 节对多核虚拟化核间中断的模拟,宿主机 VMM 内核模块在 TLB MISS 异常中将虚拟核间中断寄存器的访问以 MMIO 的方式进行处理,并将寻址读写的具体操作交给 QEMU 进程进行.为完成虚拟多核核间中断寄存器的以 MMIO 的方式进行读写的方式,需要在宿主机 VMM 内核模块以及 QEMU 进程中分别进行处理.

在 VMM 内核模块方面,主要是为虚拟多核核间中断寄存器的读写异常设置 MMIO 异常环境,以交给 QEMU 以 MMIO 的方式进行处理.具体分为两个方面:在 VMM 内核虚拟处理器结构中设置标记位;VMM 内核态与 QEMU 用户态的切换.

在 QEMU 进程方面,在 VMM 内核模块对虚拟多核核间中断寄存器的读写异常设置 MMIO 异常环境后,QEMU 对其进行一些处理,具体如下:

1) 虚拟多核核间中断寄存器读写方式的设计

在龙芯 3 处理器硬件结构中,各个处理器核的核间中断寄存器的地址是固定的,因此便于寻址.对于虚拟多核核间中断寄存器,QEMU 进程也根据其注册的固定地址进行读写,为此本文设计了虚拟核间中断寄存器读写函数.以虚拟核间中断寄存器的读取函数为例,通过 VMM 内核模块传递的 MMIO 读写字段来判断需要读取的虚拟核间中断寄存器,并根据龙芯 3 处理器物理硬件核间中断寄存器的真实读写权限来判断此虚拟核间中断寄存器的值是否可以读取,若此虚拟核间中断寄存器具有读权限,则将 QEMU 进程中对应的虚拟化核间中断寄存器的内容返回.虚拟核间中断寄存器的写操作与读操作类似,只是过程与其相反.

2) 虚拟核间中断寄存器结构的读写机制

因为虚拟核间中断寄存器寻址而产生的 MMIO 访问在 QEMU 进程中需要通过上述的虚拟核间中断寄存器读写函数填入相应的虚拟寄存器中,在 VMM 内核模块设置好 I/O 处理环境后,需要 QEMU 进程对 MMIO 触发标记位、MMIO 读写标记位、MMIO 地址字段、MMIO 读写长度字段等进行判断,以调用不同的 I/O 处理函

数.对于虚拟核间中断寄存器的 MMIO 操作,会调用已经注册好的虚拟核间中断读写函数.

其中需要注意的是,虚拟核间中断寄存器读写函数以 32 位数据为单位,在虚拟多核环境下,需保证 64 位的虚拟核间中断寄存器的读写为原子操作,本文以系统的线程锁来保证此读写操作的原子性.

5.3 多核虚拟化核间中断通信流程

以 MMIO 为触发方式的虚拟核间中断寄存器的访问工作流程如图 5 所示.

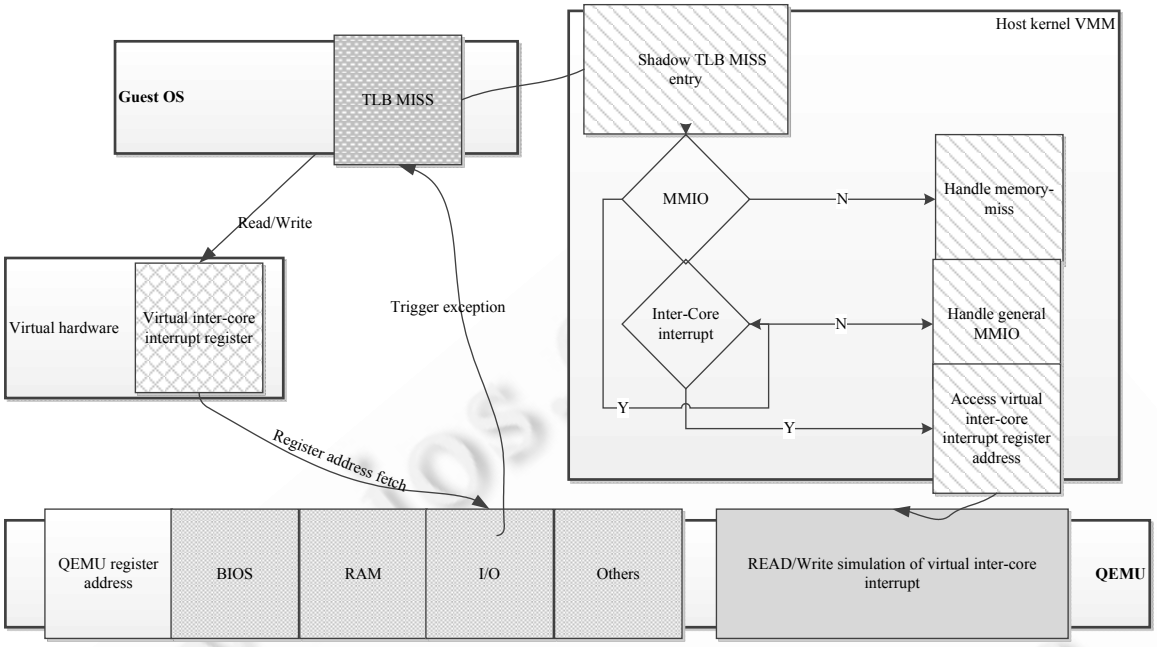

Fig.5 Inter-Core communication process of multi-core virtualization of Loongson-3 server

图 5 龙芯 3 服务器多核虚拟化核间通信流程

详细执行流程如下:

步骤 1. 虚拟机在启动过程中,将虚拟核间中断寄存器注册到 QEMU 的 I/O 地址空间中.在客户机操作系统进行核间通信时,需要对虚拟核间中断寄存器进行读写访问.这一读写操作被 QEMU 视为对 I/O 地址空间中一个 I/O 设备的访问,当设备地址不在客户机的 TLB 中时,客户机产生 TLB MISS 异常.

步骤 2. 客户机产生 TLB MISS 异常后,陷入到宿主内核模块 VMM 中,宿主内核模块 VMM 捕捉到这一异常后经过宿主机与客户机运行环境的切换,有 VMM 维护的影子 TLB MISS 处理程序来处理异常.

步骤 3. 在影子 TLB MISS 处理程序中,宿主 VMM 模块首先判断产生异常的虚拟地址是否属于 MMIO,若不属于 MMIO,则判断为正常的内存地址异常,并由相应的内存缺失程序进行处理;若属于 MMIO 异常,则进一步判断此异常虚拟地址是普通的 I/O 地址还是虚拟核间中断寄存器注册的 I/O 地址,普通的 I/O 地址由通用 MMIO 处理程序进行处理.虚拟核间中断寄存器地址访问异常则需要由 QEMU 进程中的虚拟核间中断寄存器读写模拟来完成此次访问.

6 系统测试与分析

我们进行了功能和性能测试以证明多核虚拟化方案的正确性.

6.1 测试环境概述

1) 硬件测试环境概述

本文的硬件测试平台目前采用基于龙芯 3A 四核处理器、RS780E 芯片组的硬件开发平台,具体信息见表 4.图 6 展示了我们测试使用的龙芯 3 多核处理器测试板.

Table 4 Hardware configurations

表 4 硬件测试环境

硬件类型	硬件配置
CPU 类型	Loongson-3A
CPU 数目	1
处理器核数目	4
内存大小	2G
芯片组	AMD-RS780E
网卡	RTL8193
硬盘	SSD 32G SATA

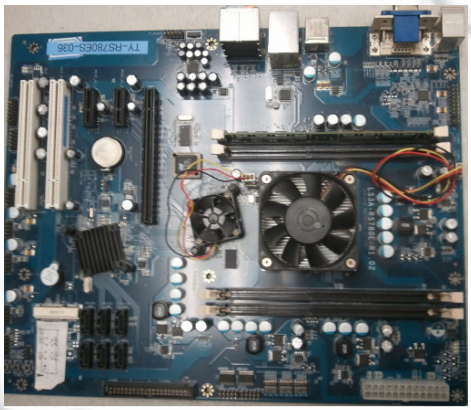

Fig.6 Test board of Loongson-3 multi-core processor

图 6 龙芯 3 多核处理器测试板

2) 软件测试环境概述

软件测试环境需要底层宿主系统、中间宿主 QEMU 进程以及上层客户系统 3 个部分,具体信息见表 5.

Table 5 Software configurations

表 5 软件测试环境

软件类型	软件配置
宿主机系统发行版本	Redhat 6.0
宿主机系统内核版本	修改过的 Linux-2.3.32
客户机系统发行版本	Debian 6
客户机系统内核版本	修改过的 Linux-2.3.32
QEMU 版本	修改过的 QEMU-0.14.0

6.2 功能测试

目前,多核虚拟化工作在功能方面可以正常运行.我们以 2 核虚拟化进行功能测试.图 6 和图 7 显示了龙芯 3 服务器上 2 核虚拟化的运行情况.其中,图 6 所示为进入客户机操作系统后通过查看/proc 中的 cpuinfo 信息,可以看到,目前客户机中运行着两个核,处理器核的类型为 loongson-3a-qemu.图 7 显示了客户机运行过程中的 top 命令信息,并显示了两个虚拟处理器核的利用率情况.

```
sh-3.2# cat /proc/cpuinfo
system type             : loongson-3a-qemu
processor               : 0
cpu model               : ICT Loongson-3-qemu V0.5  FPU V0.1
BogoMIPS                : 530.43
wait instruction        : no
microsecond timers      : yes
tlb_entries             : 64
extra interrupt vector  : no
hardware watchpoint     : yes, count: 0, address/irw mask: []
ASEs implemented        :
shadow register sets    : 1
core                    : 0
VCED exceptions         : not available
VCEI exceptions         : not available

processor               : 1
cpu model               : ICT Loongson-3-qemu V0.5  FPU V0.1
BogoMIPS                : 716.80
wait instruction        : no
microsecond timers      : yes
tlb_entries             : 64
extra interrupt vector  : no
hardware watchpoint     : yes, count: 0, address/irw mask: []
ASEs implemented        :
shadow register sets    : 1
core                    : 0
VCED exceptions         : not available
VCEI exceptions         : not available
```

Fig.7 cpuinfo of dual-core virtual machine

图 7 双核虚拟机的 cpuinfo 信息

```
top - 10:04:55 up 10:15,  0 users,  load average: 2.20, 2.24, 2.00
Tasks:  29 total,   1 running,  28 sleeping,   0 stopped,   0 zombie
Cpu0  :  4.5%us,  2.1%sy,  0.0%ni, 92.5%id,  0.0%wa,  0.0%hi,  0.9%si,  0.0%st
Cpu1  :  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    249984k total,    21696k used,   228288k free,     3456k buffers
Swap:        0k total,        0k used,        0k free,     4352k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1029 root      20   0  3904 3264 2240 R  739  1.3  69:09.11 top
    1 root      20   0  4544 4160 3008 S    0  1.7 151:01.75 sh
    2 root      20   0     0    0    0 S    0  0.0   2:55.81 kthreadd
    3 root      20   0     0    0    0 S    0  0.0 183:47.32 ksoftirqd/0
  173 root      20   0     0    0    0 S    0  0.0   0:00.00 kworker/0:0
  ??? root      20   0     0    0    0 S    0  0.0  43:32.65 kworker/u:0
    6 root      RT   0     0    0    0 S    0  0.0   2:01.75 migration/0
    7 root      RT   0     0    0    0 S    0  0.0   2:55.81 kthreadd
    4 root      20   0     0    0    0 S    0  0.0   0:00.00 kworker/0:0
    5 root      20   0     0    0    0 S    0  0.0  50:16.25 ksoftirqd/1
    6 root      RT   0     0    0    0 S    0  0.0   0:00.00 khelper
    7 root      RT   0     0    0    0 S    0  0.0   0:00.00 kworker/u:1
    9 root      20   0     0    0    0 S    0  0.0  50:16.25 ksoftirqd/1
   10 root       0 -20     0    0    0 S    0  0.0   0:00.00 khelper
   11 root      20   0     0    0    0 S    0  0.0   0:00.00 kworker/u:1
  174 root       0 -20     0    0    0 S    0  0.0   0:00.00 khlockd
```

Fig.8 Top command of dual-core virtual machine

图 8 双核虚拟机的 top 命令信息

6.3 性能测试

在性能测试部分,为了体现多核相对于单核的性能优势,我们分别在单核虚拟化和多核虚拟化环境下虚拟机进行多线程的整数加法、整数乘法、浮点加法以及浮点乘法的对比测试.其中,每一种计算均为 10 亿次级别的运算.

表 6 列出了在客户机中同时运行 2 个线程进行计算的测试结果,由对比可见,我们的多核虚拟化工作在运行同一级别的运算时运行时间几乎比单核虚拟化的运行时间快 1 倍,优化百分比达到了 40%~50%.同样地,表 7 所示的 4 个运算线程的性能测试优化百分比也达到了 40%以上.

Table 6 Performance of two computing threads

表 6 两个运算线程计算性能测试

	运行时间(s)		
	单核虚拟化	多核虚拟化	优化百分比
整数加法	76.889	41.519	46%
整数乘法	80.299	45.489	43%
浮点数加法	106.639	55.059	48%
浮点数乘法	105.539	51.969	51%

Table 7 Performance of four computing threads

表 7 4 个运算线程性能测试

	运行时间(s)		
	单核虚拟化	多核虚拟化	优化百分比
整数加法	77.089	44.899	42%
整数乘法	80.588	47.809	41%
浮点数加法	107.419	61.519	43%
浮点数乘法	107.799	61.578	43%

综上,我们进行了功能和性能测试以确认多核虚拟化的正确性和有效性.性能测试的结果表明,我们设计的龙芯 3 多核虚拟机上计算任务的运行时间大约是单核虚拟机的一半.

7 结束语

本文分析了龙芯 3 处理器在实现多核虚拟化时所面临的挑战.通过对龙芯 3 处理器多核架构的进一步分析,设计并实现了龙芯 3 处理器上的多核虚拟化.为了证明本文工作的有效性,在多核龙芯 3 处理器上进行了测试.测试结果表明,本文所提出的方法能够有效地支持龙芯 3 号处理器多核虚拟化.

致谢 在此,我们感谢蔡万伟和台运方在 CPU 虚拟化和异常处理机制方面所做的工作,也感谢徐威给我们提供的帮助.

References:

[1] Geer D. Chip makers turn to multicore processors. IEEE Computer, 2005,38(5):11−13.

[2] Posner EA, Spier KE, Vermeule A. Divide and conquer. Journal of Legal Analysis, 2010,2(2):417−471.

[3] Wolf C. Let's Get Virtual: A look at today's server virtualization architectures. Data Center Strategies, 2007.

[4] Intel® Corporation. System Virtualization. Beijing: Tsinghua University Press, 2008.

[5] Ruan L, Wang HX, Xiao LM, Zhu MF, Li FO. Memory virtualization for MIPS processor based cloud server. In: Advances in Grid and Pervasive Computing. LNCS 7296, 2012. 54−63.

[6] Ruan L, Xiao LM, Zhu MF. Content addressable storage optimization for desktop virtualization based disaster backup storage system. China Communications, 2012,9(7):1−13.

[7] Popek GJ, Goldberg RP. Formal requirements for virtualizable third generation architectures. In: Proc. of the ACM Symp. 1974.

[8] Intel® Virtualization Technology (Intel® VT).

[9] MIPS Technologies, Inc. MIPS64® architecture for programmers, Volume I: Introduction to the MIPS64® architecture. 2003.

[10] Edouard B, Scott D. Disco: Running commodity operating systems on scalable multiprocessors. In: Proc. of the 16th ACM Symp. on Operating Systems Principles. New York: ACM, 1997.

[11] Open Kernel Labs. Virtualization for embedded systems. 2007.

[12] Bellard F. QEMU, a fast and portable dynamic translator. In: Proc. of the Annual Conf. on USENIX Annual Technical Conf. USENIX Association Berkeley, 2005.

[13] Lin JW, Wang CC, Chang CY, Chen CH, Lee KJ. Full system simulation and verification framework. In: Proc. of 2009 the 5th Int'l Conf. on Information Assurance and Security. 2009.

[14] Kivity A, Kamay Y, Laor D, Lublin U, Liguori A. KVM: The Linux virtual machine monitor. In: Proc. of the Linux Symp. 2007.

[15] MIPS Technologies, Inc. MIPS64® architecture for programmers, Volume II: Introduction to the MIPS64® architecture. 2003.

[16] Revucky MA. Optimizing indirect branch prediction accuracy in virtual machine interpreters. In: ACM SIGPLAN Notices. 2003.

[17] Sweetman D. See MIPS Run Linux. Morgan Kaufmann Publishers, 2007.

[18] 中国科学院计算技术研究所.龙芯 GS464 处理核手册 V1.01. 2010.

阮利(1978－),女,四川成都人,博士,讲师,CCF 高级会员,主要研究领域为虚拟化与云计算,分布式与并行存储,大数据.

E-mail: ruanli@buaa.edu.cn

徐鹏(1989－),男,硕士生,主要研究领域为计算机系统结构.

E-mail: xupeng812@gmail.com

王慧祥(1988－),男,硕士生,主要研究领域为计算机系统结构.

E-mail: xiangzi_asang@163.com

祝明发(1945－),男,博士,教授,博士生导师,主要研究领域为计算机体系结构,并行处理,高性能计算机系统和网络,人工智能.

E-mail: zhumf@buaa.edu.cn

肖利民(1970－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为虚拟化与云计算,分布式与并行文件系统,高性能计算,软件定义网络.

E-mail: xiaolm@buaa.edu.cn

唐浩夫(1988－),男,硕士生,主要研究领域为计算机系统结构.

E-mail: thfbjhkhtdx@gmail.com