(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2015 年 7 月 23 日 (23.07.2015)

WIPO | PCT

(10) 国际公布号
WO 2015/106497 A1

(51) 国际专利分类号:
G06F 9/45 (2006.01)

(21) 国际申请号: PCT/CN2014/075253

(22) 国际申请日: 2014 年 4 月 14 日 (14.04.2014)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201410018108.3 2014 年 1 月 15 日 (15.01.2014) CN

(71) 申请人: **上海交通大学 (SHANGHAI JIAO TONG UNIVERSITY)** [CN/CN]; 中国上海市闵行区东川路 800 号, Shanghai 200240 (CN)。

(72) 发明人: **管海兵 (GUAN, Haibing)**; 中国上海市闵行区东川路 800 号, Shanghai 200240 (CN)。 **李健 (LI, Jian)**; 中国上海市闵行区东川路 800 号, Shanghai 200240 (CN)。 **马汝辉 (MA, Ruhui)**; 中国上海市闵行区东川路 800 号, Shanghai 200240 (CN)。 **朱敏君 (ZHU, Minjun)**; 中国上海市闵行区东川路 800 号, Shanghai 200240 (CN)。 **周凡夫 (ZHOU, Fanfu)**; 中国上海市闵行区东川路 800 号, Shanghai 200240 (CN)。

(74) 代理人: **上海旭诚知识产权代理有限公司 (SUN-RAY INTELLECTUAL PROPERTY ATTORNEYS)**; 中国上海市浦东新区东方路 710 号汤臣金融大厦 1212 室郑立, Shanghai 200122 (CN)。

(81) 指定国 (除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国 (除另有指明，要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

[见续页]

(54) **Title**: DYNAMIC INTERRUPT BALANCED MAPPING METHOD BASED ON CURRENT VCPU SCHEDULING STATE

(54) **发明名称** : 一种基于当前 VCPU 调度状态的动态中断均衡映射方法

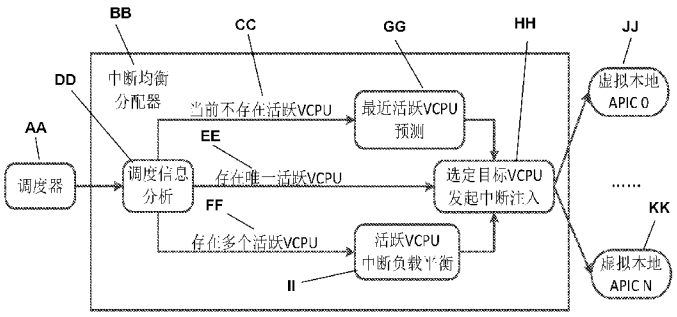


图 4 /FIG.4

AA	Scheduler	GG	Next active VCPU prediction
BB	Interrupt balance distributor	HH	Select target VCPU and initiate interrupt injection
CC	Currently no active VCPU	II	Active VCPU interrupt load balance
DD	Scheduling information analysis	JJ	Virtual local APIC 0
EE	Single active VCPU	KK	Virtual local APIC N
FF	Plurality of active VCPUs		

(57) **Abstract**: Disclosed in the present invention is a dynamic interrupt balanced mapping method based on a current VCPU scheduling state: when a virtual I/O APIC of an SMP virtual machine has received a virtual interrupt, and this virtual interrupt needs to be mapped to a VCPU of the virtual machine, on the basis of the scheduling states in a VMM scheduler of every VCPU of the current VM, parse out those VCPUs in an active state, and map the virtual interrupt to the active VCPU to acquire a lower interrupt processing latency. If a plurality of VCPUs are in an active state at the same time, then the interrupt load of each VCPU is further considered, and the interrupt is mapped to the active VCPU having the current lowest load, further ensuring balance of the interrupt processing load between the VCPUs, such that the VCPU load in the SMP structure is more symmetrical and balanced overall performance of each VCPU in the SMP structure is thereby enhanced.

(57) **摘要**:

[见续页]

WO 2015/106497 A1

本国际公布:

— 包括国际检索报告(条约第 21 条(3))。

本发明公开了基于当前 VCPU 调度状态的动态中断均衡映射方法，当 SMP 虚拟机的虚拟 I/O APIC 接收到一个虚拟中断后，需要将该虚拟中断映射给虚拟机的某一个 VCPU 时，根据当前 VM 的所有 VCPU 在 VMM 调度器中的调度状态，分析出处于活跃状态的部分 VCPU，并且将该虚拟中断映射到活跃 VCPU 中，以获得较低的中断处理延时。如果同时有多个 VCPU 处于活跃状态，则进一步考虑各个活跃 VCPU 的中断负载，并且将该中断映射给当前负载最低的活跃 VCPU，进一步保证各个 VCPU 之间的中断处理负载均衡，使得 SMP 结构下 VCPU 的负载更加对称，从而促进 SMP 结构下各个 VCPU 整体性能的均衡发挥。

说明书

发明名称：一种基于当前VCPU调度状态的动态中断均衡映射方法

[1] 技术领域

[2] 本发明涉及计算机系统虚拟化、虚拟机中断处理、虚拟机调度器领域，具体地，涉及一种基于当前VCPU调度状态的动态中断均衡映射方法。

[3] 背景技术

[4] 虚拟化技术通常将原本需要多台物理设备实现的计算或存储功能整合到一台功能相对强大的物理服务器中去，从而实现了硬件资源的整合与再分配，提高了硬件设备的利用率，在云计算以及数据中心构建中发挥着非常重要的作用。虚拟机相对于真实的物理设备，体现出很多明显的优势。首先，云计算与虚拟化技术允许企业用户使用虚拟机进行作业，从而用户不需要购置一套真实的IT设备，有效减少了企业用户在维护IT基础架构上的经济成本和管理成本。第二，基于镜像的虚拟机创建方式灵活实用，通过镜像备份技术能够很好地处理服务器灾备、重建、批量复制等问题。此外，基于虚拟机监视器（Virtual Machine Monitor，VMM）的虚拟机管理方式能够从软硬件的层面灵活处理物理资源和虚拟资源的映射关系，提供性能隔离、安全隔离、状态监控等一系列必要功能。

[5] 虚拟机监视器是指存在于硬件和传统操作系统之间的一个软件管理层，其主要作用是管理真实的物理设备，例如物理CPU、内存等，并将底层的硬件抽象为对应的虚拟设备接口，使多个操作系统得到各自所需的虚拟硬件，从而使他们能够同时在同一个物理设备上运行。

[6] 物理设备和虚拟操作系统之间引入的虚拟机监视器作为中间层，不可避免地使得虚拟操作系统的性能受到一定的影响，其中一个重要的方面是增加了中断的响应延时。多个VCPU共享一个物理CPU时，需要对物理CPU进行分时复用，VCPU在VMM的调度队列中的等待时间会反映到该VCPU对事件的响应时间中，进而使得中断等事件的响应延时严重增加。

[7] 学术界目前对于虚拟中断延时问题的研究，主要集中在单核虚拟机上。在单核虚拟机中，由于一个虚拟机只持有一个VCPU，对中断延时的解决方法多采用减

少物理CPU分时复用的时间片长度、提高VCPU之间的切换频率以及增加抢占机会等手段。这些方法在一定程度上能够降低调度延时，但同时也不可避免地增加了VCPU上下文切换的频率，引入了额外的系统开销，从而浪费了CPU的计算能力。对于对称式多处理器(Symmetric Multi-Processor，SMP)架构的虚拟机，目前学术界的研究则多集中在VCPU协同调度的问题上，即虚拟机中多个VCPU之间的同步和通信问题，对中断等事件延时问题的研究则相对较少。

[8] 目前，在虚拟机监视器的实例，例如KVM与XEN的实现中，主要采用虚拟中断处理设备的方式来处理虚拟机中的中断。在XEN中，虚拟机根据虚拟化的方式分成两种类型，即全虚拟化和半虚拟化。在全虚拟化模式下，虚拟操作系统不需要经过任何修改，XEN通过虚拟中断处理平台来实现虚拟中断处理；而在半虚拟化中，虚拟操作系统的内核需要被修改以适应宿主机的操作系统，XEN通过事件通道机制实现中断和事件处理。KVM则没有全虚拟化和半虚拟化的区别，虚拟机的中断在KVM中都是通过虚拟中断处理设备来处理的。

[9] 在虚拟机的中断架构中，和物理平台一样，每一个VCPU都对应了一个虚拟本地APIC（Advanced Programmable Interrupt Controller）由于接收中断。虚拟平台也包含了虚拟I/O

APIC、虚拟PIC等虚拟设备来发送中断。和VCPU一样，虚拟I/O APIC、虚拟本地APIC、虚拟PIC都是VMM维护的软件实体。通常SMP架构虚拟机的虚拟设备，例如虚拟网卡等调用虚拟I/O APCI发出中断请求，虚拟I/O

APIC根据中断和VCPU的映射关系选择一个VCPU作为中断的接收者，将中断请求发送到该目标VCPU的虚拟本地APIC中。虚拟本地APIC进一步利用VT-x的事件注入机制最终完成虚拟中断的注入。

[10] 需要指出的是，在多VCPU架构的虚拟机中，各个VCPU在调度器的调度下对物理CPU进行分时复用，进而出现部分VCPU活跃、部分VCPU处于排队状态的局面。现有的虚拟机中断处理技术中，当虚拟I/O

APIC需要将中断请求映射到VCPU中时，没有考虑VCPU当前的调度状态，盲目地为中断分配VCPU。虚拟中断被分配给处于调度等待状态的VCPU时，VCPU在调度队列中的等待延时成为了该中断的响应延时的一部分，从而大大提高了

中断请求的响应延迟，降低了中断请求的处理速率。

[11] 发明内容

[12] 鉴于现有技术的上述缺陷，本发明所要解决的技术问题是提供一种基于当前VCPU调度状态和中断处理负载分析的动态中断均衡映射方法，在中断映射的同时既考虑了VCPU作为中断映射的目标结构在VMM的调度器中的调度状态，又考虑了各个VCPU之间的中断负载均衡，从而有效地降低了中断处理延时。

[13] 为实现上述目的，本发明提供了一种基于当前VCPU调度状态和中断处理负载分析的动态中断均衡映射方法，其特征在于：当SMP虚拟机的虚拟I/O APIC接收到一个虚拟中断后，需要将该虚拟中断映射给虚拟机的某一个VCPU时，根据当前VM的所有VCPU在VMM调度器中的调度状态，分析出处于活跃状态的部分VCPU，并且将该虚拟中断映射到活跃VCPU中，以获得较低的中断处理延时。如果同时有多个VCPU处于活跃状态，则进一步考虑各个活跃VCPU的中断负载，并且将该中断映射给当前负载最低的活跃VCPU，以实现VCPU中断负载平衡。

[14] 本发明提供一种基于当前VCPU调度状态的动态中断均衡映射方法，其特征在于，方法包括以下步骤：

[15] （1）每当虚拟硬件产生一个虚拟中断，虚拟I/O APIC接收虚拟中断，并将虚拟中断传送到目标VCPU的虚拟本地APIC；

[16] （2）中断均衡分配器在虚拟中断被传送到目标VCPU的虚拟本地APIC之前拦截虚拟中断；

[17] （3）中断均衡分配器查看调度器提供的调度状态信息，获得处于活跃状态的VCPU列表；

[18] （4）中断均衡分配器根据处于活跃状态的VCPU列表，重新选定目标VCPU；

[19] （5）中断均衡分配器将虚拟中断传送到步骤（3）中重新选定的目标VCPU的虚拟本地APIC；

[20] （6）目标VCPU的虚拟本地APIC向目标VCPU进行虚拟中断注入。

[21] 进一步地，步骤（3）中选定目标VCPU的方法包括步骤：

[22] （31）获得处于活跃状态的VCPU列表中VCPU的个数；

[23] （32）如果个数为0，根据调度状态信息，选定目标VCPU；如果个数为1，选

定处于活跃状态的VCPU列表中VCPU为目标VCPU；如果个数大于1，根据处于活跃状态的VCPU列表中VCPU的中断负载，选定目标VCPU。

[24] 当处于活跃状态的VCPU列表中只有一个VCPU，选定该VCPU为目标VCPU，将虚拟中断映射到该活跃VCPU中，以获得较低的中断处理延时。

[25] 进一步地，步骤（32）中根据调度状态信息，选定目标VCPU的方法包括步骤：

[26] （321）读取调度状态信息；

[27] （332）选定所有VCPU中预测最快进入活跃状态的VCPU为目标VCPU。

[28] 当处于活跃状态的VCPU列表中没有VCPU，也就是没有VCPU处于活跃状态，选定预测最快进入活跃状态的VCPU为目标VCPU，将虚拟中断映射到该活跃VCPU中，以获得较低的中断处理延时。

[29] 进一步地，步骤（332）中预测最快进入活跃状态的VCPU的依据是VCPU处于空闲状态。

[30] 处于空闲状态的VCPU可能最快进入活跃状态，这是因为CREDIT调度器设计允许IDLE状态的VCPU在需要应答事件时进行抢占，因此选定处于空闲状态的VCPU为目标VCPU，将虚拟中断映射到该VCPU中，以获得较低的中断处理延时。

[31] 进一步地，如果所述VCPU均未处于空闲状态，步骤（332）中预测最快进入活跃状态的VCPU的依据VCPU在等待队列中位置与剩余信用值。

[32] 对于所有的VCPU，优先考虑其在各自的等待队列中所处的位置，越是靠近队首的VCPU就越有可能最快进入活跃状态。然后，当有相同排位的VCPU时，再考虑VCPU的剩余信用值，选择剩余信用值大的作为目标VCPU，以获得较低的中断处理延时。

[33] 进一步地，步骤（32）中根据处于活跃状态的VCPU列表中VCPU的中断负载，选定目标VCPU的方法包括步骤：

[34] （323）读取当前虚拟机结构体维护的VCPU中断负载表；

[35] （334）比较处于活跃状态的VCPU列表中VCPU的中断负载，选定中断负载最小的VCPU为目标VCPU。

[36] 当处于活跃状态的VCPU列表中有多个VCPU，将虚拟中断映射给当前中断负载最低的活跃VCPU，在获得较低的中断处理延时的同时实现VCPU中断负载平衡。

[37] 进一步地，步骤（1）中虚拟硬件包括为虚拟设备和/或经VMM中断处理的物理设备。

[38] 进一步地，步骤（5）中通过VT-x的事件注入机制完成虚拟中断注入。

[39] 进一步地，在虚拟机SHARED_INFO结构的基础上增加成员变量sched_info，用于记录VCPU的调度状态。

[40] 进一步地，每当调度器发生一次VCPU切换，被换入的VCPU和被换出的VCPU完成上下文交换后，被换入的VCPU成为活跃VCPU，活跃VCPU所属的虚拟机SHARED_INFO结构的sched_info成员变量标记为活跃；被换出的VCPU的所属的虚拟机SHARED_INFO结构的sched_info成员变量标记为等待。

[41] 与现有技术相比，本发明提供的基于当前VCPU活跃状态的动态中断映射方法具有以下有益的技术效果：

[42] （1）根据当前VCPU的调度状态，将虚拟中断映射到活跃的VCPU，有效减少了因为调度延时而产生的中断请求处理延时，从而保证了虚拟中断及时注入目标VCPU中，获得较低的中断处理延时，提高了中断响应速度。

[43] （2）当多个VCPU处于活跃状态时，选定中断负载最小的VCPU为目标VCPU，进一步保证各个VCPU之间的中断处理负载均衡，使得SMP结构下VCPU的负载更加对称，从而促进SMP结构下各个VCPU整体性能的均衡发挥。

[44] （3）由于本发明的方案并没有要求减小调度周期的时间片长度、增加VCPU上下文切换频率或在VCPU之间引入新的抢占机制，因此，本发明在有效减少中断响应延时的同时，没有引入额外的计算能力的浪费，有利于计算密集型负载的运行，保证了较高的CPU利用率。

[45] （4）由于中断在操作系统中具有很重要的作用，提高中断处理速率能够使得虚拟机的整体响应能力也得到一定程度的提高。

[46] 附图说明

[47] 图1是物理平台上的中断处理结构示意图；

[48] 图2是现有的SMP虚拟机技术中虚拟中断处理器的结构示意图；

[49] 图3是本发明的一个实施例的虚拟中断处理器的示意图；

[50] 图4是图3所示的虚拟中断处理器的虚拟中断映射的建立过程示意图。

[51] 具体实施方式

[52] 以下将结合附图对本发明的构思、具体结构及产生的技术效果作进一步说明，以充分地了解本发明的目的、特征和效果。

[53] 图1是物理平台上的中断处理结构示意图。在物理平台上，I/O设备首先通过中断控制器（I/O APIC 或者PIC）发出中断请求，这个中断请求通过PCI总线发送到系统总线上，最后目标CPU的本地 APIC部件接收此中断，目标CPU开始处理该中断。

[54] 图2是现有的SMP虚拟机技术中虚拟中断处理器的结构示意图。在虚拟机的环境下，VMM也需要为客户机操作系统呈现一个与物理中断架构类似的虚拟中断架构。和在真实的物理平台上一样，在虚拟SMP架构下，每一个VCPU也对应了一个虚拟本地 APIC用于接收中断，而VMM则提供虚拟I/O APIC用于发送中断。当某个虚拟设备，例如虚拟网卡等需要发送一个中断时，虚拟设备首先调用虚拟I/O APIC提供的接口发送中断。虚拟I/O APIC根据得到的中断请求，选择一个VCPU作为该中断的处理者，将中断交给该VCPU对应的本地APIC。最后，虚拟本地APIC利用VT-x的事件注入机制将中断注入到对应的目的VCPU中，由VCPU处理中断。类似于VCPU，虚拟中断处理器中涉及到的虚拟本地APIC、虚拟I/O APIC、虚拟PIC等也都是VMM维护的软件实体。

[55] 图3是本发明的一个实施例的虚拟中断处理器的示意图。与现有的虚拟中断处理器相比，本实施例中的虚拟中断处理器中增加了中断均衡分配器，中断均衡分配器处于原有的虚拟I/O APIC与虚拟本地APIC之间。

[56] 中断均衡分配器从VMM的调度器中读取到VCPU的基本调度信息，分析出当前时刻虚拟机的VCPU的调度状态，即哪些VCPU处于活跃状态，哪些VCPU处于调度等待状态。当虚拟I/O APIC需要发送一个中断请求时，中断均衡分配器截获此中断请求，根据已有的知识代替虚拟I/O APIC选择处理中断的目标VCPU，进

而将该中断发送至目标VCPU的虚拟本地APIC，发起中断注入。

[57] 图4是图3所示的虚拟中断处理器的虚拟中断映射的建立过程示意图。图中具体说明了中断均衡分配器的工作过程。当中断均衡分配器需要处理一个中断请求时，中断均衡分配器首先基于从调度器得到的调度信息来分析当前虚拟机的各个VCPU的调度和中断负载情况。

[58] 若当前只存在唯一的活跃VCPU，即该虚拟机的VCPU中只有一个VCPU目前取得了实际的运行权，在物理CPU上运行，则将此VCPU作为目标VCPU记录下来。

[59] 若当前没有VCPU处于活跃状态，即虚拟机的所有VCPU均处在调度器的等待队列中，则中断均衡分配器对各个VCPU进入活跃状态的可能性进行推测，选定最快能够进入活跃状态的VCPU作为目标VCPU。推测的依据来自调度器提供的信息，例如VCPU是否处于IDLE状态，CREDIT调度器下各个VCPU的剩余信用值等。一般来说，在CREDIT调度器中，处于IDLE状态的VCPU是最合适的目标选择，因为CREDIT调度器设计允许IDLE状态的VCPU在需要应答事件时进行抢占。对于所有的VCPU，优先考虑其在各自的等待队列中所处的位置，越是靠近队首的VCPU就越有可能最快进入活跃状态。然后，当有相同排位的VCPU时，再考虑VCPU的剩余信用值，选择剩余信用值大的作为目标VCPU，以获得较低的中断处理延时。

[60] 若当前有多个VCPU处于活跃状态，中断均衡分配器依次读取各个处于活跃状态的VCPU的中断负载情况，选取最小负载的VCPU作为目标VCPU。确定目标VCPU之后，中断均衡分配器继续中断处理流程，代替虚拟I/O APIC发起中断注入。

[61] 本发明基于当前VCPU活跃状态和VCPU中断处理负载分析进行中断均衡分配的动态中断映射方法具体步骤如下：

[62] 步骤1、每当VMM的VCPU调度器发生一次有效的VCPU切换，被换入的VCPU和被换出的VCPU完成上下文交换后，被换入的VCPU成为活跃VCPU，真正在物理CPU上运行，此时该VCPU所属的虚拟机结构体根据VCPU的ID将被换入的VCPU状态标记为活跃，同时把被换出的VCPU的状态标记为等待。

[63] 步骤2、每当虚拟设备产生一个虚拟中断需要处理时，在该虚拟中断被虚拟I/O APIC映射到VCPU之前拦截该虚拟中断，同时查看当前虚拟机的VCPU状态表，得出处于活跃状态的VCPU列表。若活跃VCPU列表中只有一个VCPU，则选中其为目标VCPU，转步骤5；若当前时刻没有VCPU处于活跃状态，转步骤3；若有多于一个VCPU处于活跃状态，转步骤4。

[64] 步骤3、当前时刻没有VCPU处于活跃状态，则根据调度器提供的各个VCPU调度信息推测最早能够进入活跃状态的VCPU，选中其为目标VCPU，转步骤5.

[65] 步骤4、当存在多个处于活跃状态的VCPU时，则进一步读取当前虚拟机结构体维护的VCPU中断负载表，比较各个活跃VCPU的当前中断负载，选取负载相对最小的VCPU作为目标VCPU。

[66] 步骤5、更新目标VCPU的中断负载状态，并且将当前虚拟中断映射到动态选取的目标VCPU中，最后继续完成中断注入。

[67] 具体地，本发明的中断映射方案具体实现步骤主要分为两部分，第一部分步骤如下：

[68] 步骤1、在原有虚拟机的SHARED_INFO结构的基础上增加一个新的成员变量sched_info，用于记录VCPU的调度状态。该步骤的主要代码实现如下：

[69] struct shared_info {

[70] …

[71] uint64_t sched_info;

[72] }

[73] 步骤2、每当VMM的VCPU调度器发生一次有效的VCPU切换，被换入的VCPU和被换出的VCPU完成上下文交换后，被换入的VCPU成为活跃VCPU，即真正在物理CPU上运行，此时该VCPU所属的虚拟机结构体根据VCPU的ID将被换入的VCPU状态标记为活跃，同时把被换出的VCPU的状态标记为等待。该步骤的主要代码实现如下：

[74] static void schedule(void) ()

[75] {

[76] …

[77] context_switch(prev, next);

[78] if (check_pre_ok(prev, next))

[79] {

[80] prev->domain->shared_info->native.sched_info=((prev->domain->shared_info->native.sched_info>>(prev->vcpu_id-1))-1)<<(prev->vcpu_id-1);

[81] next->domain->shared_info->native.sched_info=((next->domain->shared_info->native.sched_info>>(next->vcpu_id-1))+1)<<(next->vcpu_id-1);

[82] }

[83] context_switch(prev, next);

[84] }

[85] 本发明的具体实现步骤的第二部分修改现有虚拟I/O APIC对虚拟中断的中断映射方案，具体步骤如下：

[86] 步骤1、每当VMM产生一个虚拟中断时，虚拟I/O APIC在该虚拟中断被映射到VCPU之前拦截该虚拟中断，同时查看当前虚拟机的VCPU状态表，得出处于活跃状态的VCPU列表。若当前活跃VCPU列表中只有一个VCPU，则选中其为目标VCPU，转步骤4；若当前时刻没有VCPU处于活跃状态，转步骤2；若有多于一个VCPU处于活跃状态，转步骤3。

[87] 步骤2、当前时刻没有VCPU处于活跃状态，则根据调度器提供的各个VCPU调度信息推测最早能够进入活跃状态的VCPU，选中其为目标VCPU，转步骤4.

[88] 步骤3、当存在多个处于活跃状态的VCPU时，进一步读取当前虚拟机结构体维护的VCPU中断负载表，比较各个活跃VCPU的当前中断负载，选取负载相对最小的VCPU作为目标VCPU，转步骤4。

[89] 步骤4、更新目标VCPU的中断负载状态，并且将当前虚拟中断映射到动态选取的目标VCPU中，继续完成中断注入。

[90] 该部分的核心代码实现如下：

[91] static void vioapic_deliver(struct hvm_hw_vioapic *vioapic, int irq)

[92] {

[93] …

```
[94]    target = target_search();
[95]    ioapic_inj_irq(vioapic, target, vector, trig_mode, delivery_mode);
[96]    }
[97]    struct vlapic *target_search(struct domain *d)
[98]    {
[99]    int target_vcpu=0;
[100]   uint64_t tmp= d->vcpu[d->shared_info->native. sched_info;
[101]   if (tmp==0)
[102]   {
[103]   target_vcpu = most_likely_active(d);
[104]   return vcpu_vlapic(target_vcpu);
[105]   }
[106]   while ((tmp&1)!=1)
[107]   {
[108]   tmp = tmp>>1;
[109]   target_vcpu = target_vcpu+1;
[110]   }
[111]   if (tmp==0)
[112]   {
[113]   return vcpu_vlapic(target_vcpu); }
[114]   while (tmp!=0)
[115]   {
[116]   if ((tmp&1)==1)
[117]   saved_active_vcpu(target_vcpu);
[118]   tmp = tmp>>1;
[119]   target_vcpu = target_vcpu+1;
[120]   }
[121]   target_vcpu = min_load_vcpu(saved_active_vcpu()) ;
```

[122] return vcpu_vlapic(target_vcpu); }

[123] 本发明提供的基于当前VCPU活跃状态的动态中断映射方法，根据当前VCPU的调度状态，将虚拟中断映射到活跃的VCPU，有效减少了因为调度延时而产生的中断请求处理延时，从而保证了虚拟中断及时注入目标VCPU中，获得较低的中断处理延时，提高了中断响应速度；当多个VCPU处于活跃状态时，选定中断负载最小的VCPU为目标VCPU，进一步保证各个VCPU之间的中断处理负载均衡，使得SMP结构下VCPU的负载更加对称，从而促进SMP结构下各个VCPU整体性能的均衡发挥。

[124] 由于本发明的方案并没有要求减小调度周期的时间片长度、增加VCPU上下文切换频率或在VCPU之间引入新的抢占机制，因此，本发明在有效减少中断响应延时的同时，没有引入额外的计算能力的浪费，有利于计算密集型负载的运行，保证了较高的CPU利用率；由于中断在操作系统中具有很重要的作用，提高中断处理速率能够使得虚拟机的整体响应能力也得到一定程度的提高。

[125] 以上详细描述了本发明的较佳具体实施例。应当理解，本领域的普通技术人员无需创造性劳动就可以根据本发明的构思做出诸多修改和变化。因此，凡本技术领域中技术人员依本发明的构思在现有技术的基础上通过逻辑分析、推理或者有限的实验可以得到的技术方案，皆应在由权利要求书所确定的保护范围内。

权利要求书

[权利要求 1] 一种基于当前VCPU调度状态的动态中断均衡映射方法，其特征在于，所述方法包括以下步骤：

（1）每当虚拟硬件产生一个虚拟中断，虚拟I/O APIC接收所述虚拟中断，并将所述虚拟中断传送到目标VCPU的虚拟本地APIC；

（2）中断均衡分配器在所述虚拟中断被传送到所述目标VCPU的虚拟本地APIC之前拦截所述虚拟中断；

（3）所述中断均衡分配器分析调度器提供的调度状态信息，获得处于活跃状态的VCPU列表；

（4）所述中断均衡分配器根据所述处于活跃状态的VCPU列表，重新选定目标VCPU；

（5）所述中断均衡分配器将所述虚拟中断传送到步骤（3）中重新选定的目标VCPU的虚拟本地APIC；

（6）所述目标VCPU的虚拟本地APIC向所述目标VCPU进行所述虚拟中断注入。

[权利要求 2] 如权利要求1所述的方法，其特征在于，步骤（3）中选定目标VCPU的方法包括步骤：

（31）获得所述处于活跃状态的VCPU列表中VCPU的个数；

（32）如果所述个数为0，根据所述调度状态信息，选定所述目标VCPU；如果所述个数为1，选定所述处于活跃状态的VCPU列表中VCPU为所述目标VCPU；如果所述个数大于1，根据所述处于活跃状态的VCPU列表中VCPU的中断负载，选定所述目标VCPU。

[权利要求 3] 如权利要求2所述的方法，其特征在于，步骤（32）中根据调度状态信息，选定所述目标VCPU的方法包括步骤：

（321）读取所述调度状态信息；

（332）选定所有VCPU中预测最快进入活跃状态的VCPU为所述目标VCPU。

[权利要求 4] 如权利要求3所述的方法，其特征在于，步骤（332）中预测最快进入活跃状态的VCPU的依据是VCPU处于空闲状态。

[权利要求 5] 如权利要求3所述的方法，其特征在于，如果所述VCPU均未处于空闲状态，步骤（332）中预测最快进入活跃状态的VCPU的依据是VCPU在等待队列中位置与剩余信用值。

[权利要求 6] 如权利要求2所述的方法，其特征在于，步骤（32）中根据所述处于活跃状态的VCPU列表中VCPU的中断负载，选定目标VCPU的方法包括步骤:

（323）读取当前虚拟机结构体维护的VCPU中断负载表；

（334）比较所述处于活跃状态的VCPU列表中VCPU的中断负载，选定中断负载最小的VCPU为目标VCPU。

[权利要求 7] 如权利要求1所述的方法，其特征在于，步骤（1）中所述虚拟硬件包括为虚拟设备和/或经VMM中断处理的物理设备。

[权利要求 8] 如权利要求1所述的方法，其特征在于，步骤（5）中通过VT-x的事件注入机制完成所述虚拟中断注入。

[权利要求 9] 如权利要求1所述的方法，其特征在于，在虚拟机SHARED_INFO结构的基础上增加成员变量sched_info，用于记录VCPU的调度状态。

[权利要求 10] 如权利要求9所述的方法，其特征在于，每当调度器发生一次VCPU切换，被换入的VCPU和被换出的VCPU完成上下文交换后，被换入的VCPU成为活跃VCPU，所述活跃VCPU所属的虚拟机SHARED_INFO结构的sched_info成员变量标记为活跃；被换出的VCPU的所属的虚拟机SHARED_INFO结构的sched_info成员变量标记为等待。

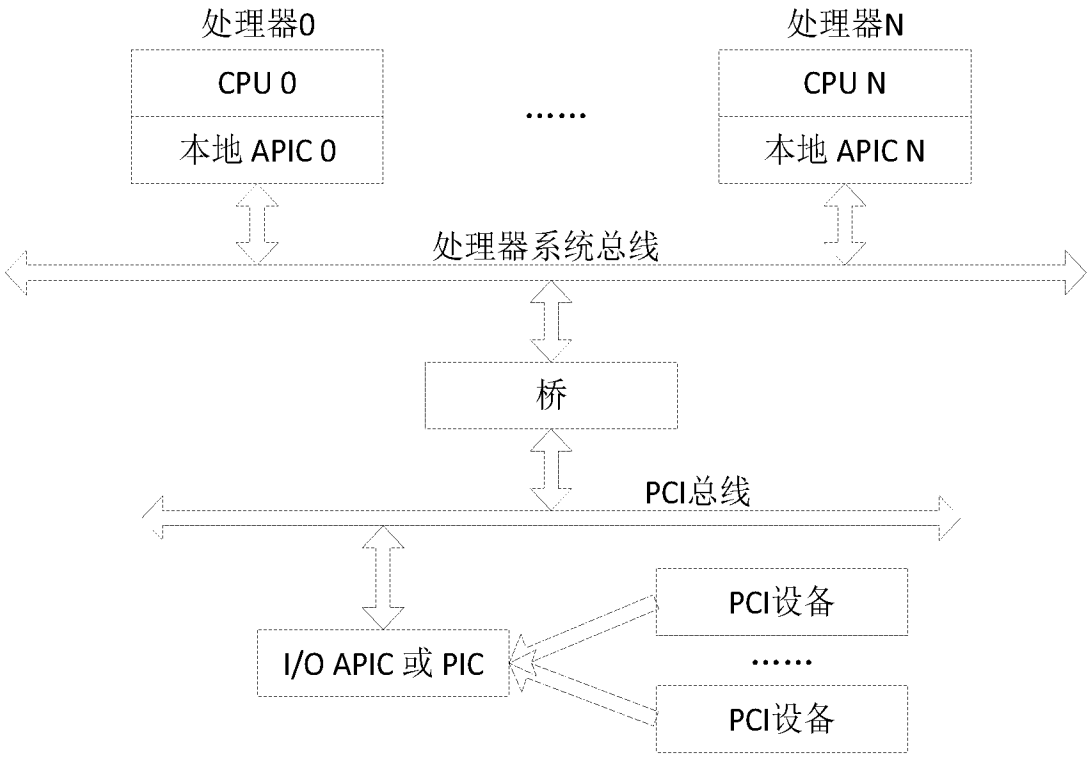

处理器0
处理器N
CPU 0
CPU N
本地 APIC 0
本地 APIC N
处理器系统总线
桥
PCI总线
I/O APIC 或 PIC
PCI设备
PCI设备

图 1

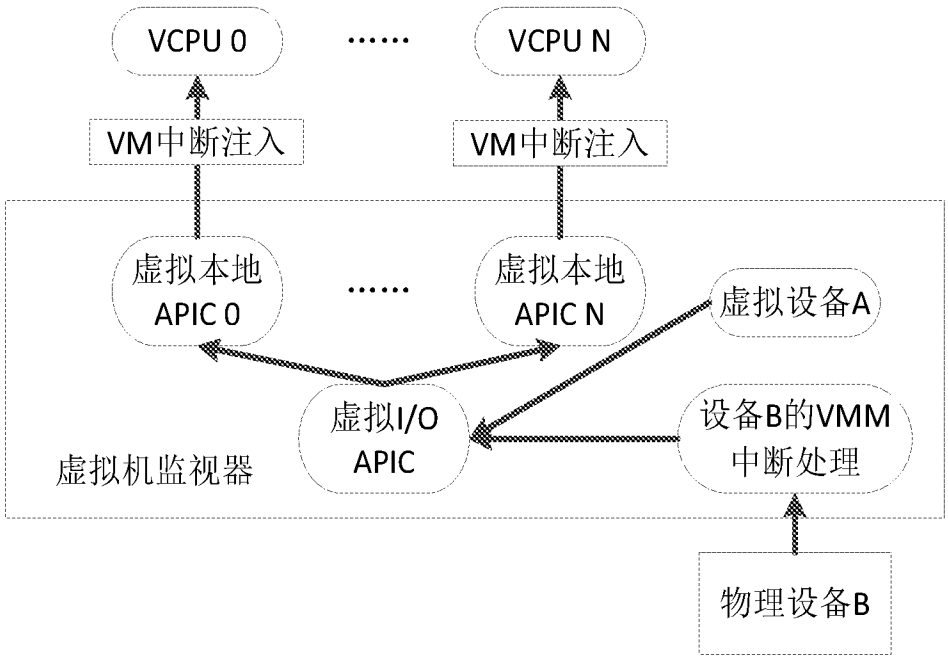

VCPU 0
VCPU N
VM中断注入
VM中断注入
虚拟本地 APIC 0
虚拟本地 APIC N
虚拟设备A
虚拟I/O APIC
设备B的VMM 中断处理
虚拟机监视器
物理设备B

图 2

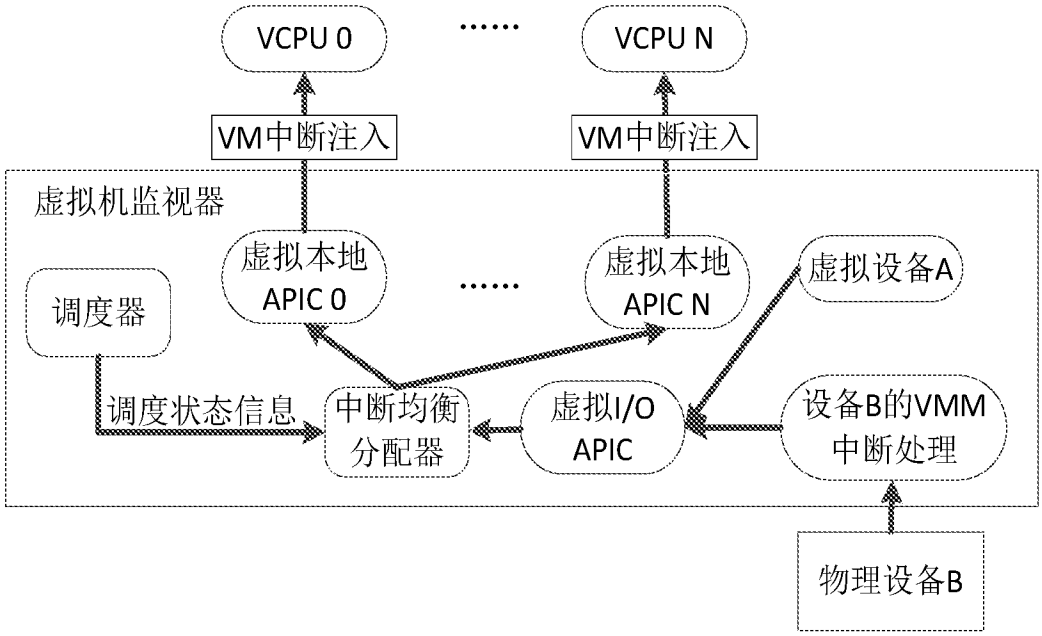

VCPU 0
……
VCPU N
VM中断注入
VM中断注入
虚拟机监视器
虚拟本地
APIC 0
……
虚拟本地
APIC N
虚拟设备A
调度器
调度状态信息
中断均衡
分配器
虚拟I/O
APIC
设备B的VMM
中断处理
物理设备B

图 3

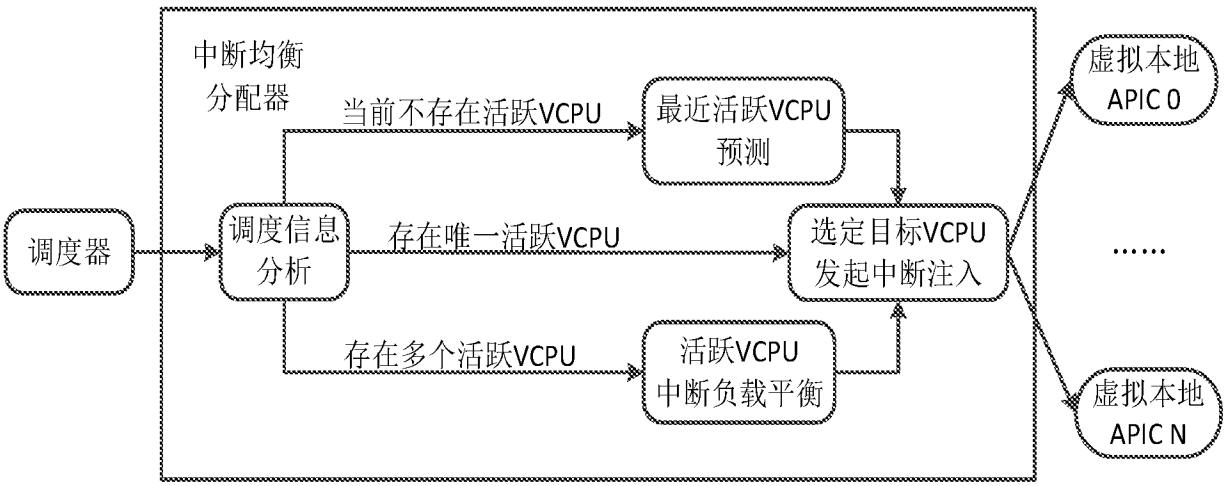

中断均衡
分配器
当前不存在活跃VCPU
最近活跃VCPU
预测
调度器
调度信息
分析
存在唯一活跃VCPU
选定目标VCPU
发起中断注入
存在多个活跃VCPU
活跃VCPU
中断负载平衡
虚拟本地
APIC 0
……
虚拟本地
APIC N

图 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/075253

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/45 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CPRSABS, CNKI, DWPI, SIPOABS: intercept, selected, virtual, dummy, despatch, dispatch, marshal, schedule, deploy, invoke, invocation, call, assign, allocate, assignation, interrupt, interruption, break, abort, disruption, disrupt, map, mapping, halting, capture, seizure, acquisition, list, choose, choice, select, elect, convey, transfer, transmit

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011202699 A1 (RED HAT INC.), 18 August 2011 (18.08.2011), the whole document	1-10
A	CN 101354663 A (LENOVO (BEIJING) CO., LTD.), 28 January 2009 (28.01.2009), the whole document	1-10
A	CN 101382923 A (LENOVO (BEIJING) CO., LTD.), 11 March 2009 (11.03.2009), the whole document	1-10

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 September 2014 (16.09.2014)	**30 September 2014 (30.09.2014)**
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer **DENG, Juan** Telephone No.: (86-10) **62411644**

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2014/075253

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
US 2011202699 A1	18 August 2011	US 8312195 B2	13 November 2012
CN 101354663 A	28 January 2009	US 2009031304 A1	29 January 2009
		US 8191064 B2	29 May 2012
CN 101382923 A	11 March 2009	CN 101382923 B	14 April 2010

国际检索报告

国际申请号
PCT/CN2014/075253

A. 主题的分类

G06F 9/45 (2006.01) i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用））

CPRSABS,CNKI,DWPI,SIPOABS：虚拟，调度，调用，分派，中断，映射，拦截，截获，捕获，列表，选择，选定，传送，传输，virtual, dummy, despatch, dispatch, marshal,schedule, deploy, invoke, invocation, call, assign, allocate, assignation, interrupt, interruption, break, abort, disruption, disrupt, map, mapping, halting, capture, seizure, acquisition, list, choose, choice, select, elect, convey, transfer, transmit

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	US 2011202699 A1 (RED HAT INC) 2011年 8月 18日 (2011-08-18) 全文	1-10
A	CN 101354663 A (联想北京有限公司) 2009年 1月 28日 (2009-01-28) 全文	1-10
A	CN 101382923 A (联想北京有限公司) 2009年 3月 11日 (2009-03-11) 全文	1-10

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2014年 9月 16日	2014年 9月 30日

ISA/CN的名称和邮寄地址	受权官员
中华人民共和国国家知识产权局(ISA/CN) 北京市海淀区蓟门桥西土城路6号 100088 中国	邓隽
传真号 (86-10)62019451	电话号码 (86-10)62411644

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2014/075253

检索报告引用的专利文件			公布日（年/月/日）	同族专利			公布日（年/月/日）
US	2011202699	A1	2011年 8月 18日	US	8312195	B2	2012年 11月 13日
CN	101354663	A	2009年 1月 28日	US	2009031304	A1	2009年 1月 29日
				US	8191064	B2	2012年 5月 29日
CN	101382923	A	2009年 3月 11日	CN	101382923	B	2010年 4月 14日