(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 106033368 B
(45)授权公告日 2019.02.22

(21)申请号 201510102323.6

(22)申请日 2015.03.09

(65)同一申请的已公布的文献号
申请公布号 CN 106033368 A

(43)申请公布日 2016.10.19

(73)专利权人 北京大学
地址 100871 北京市海淀区颐和园路5号

(72)发明人 李春奇 任仕儒 谭乐 肖臻

(74)专利代理机构 北京万象新悦知识产权代理有限公司 11360
代理人 苏爱华

(51)Int.Cl.
G06F 9/455(2006.01)

(56)对比文件
CN 101563674 A,2009.10.21,
CN 104375819 A,2015.02.25,
CN 104246696 A,2014.12.24,
CN 104216764 A,2014.12.17,
US 2011078666 A1,2011.03.31,
Yufei Chen.Scalable Deterministic Replay in a Parallel Full-system Emulator.《ACM SIGPLAN symposium on Principies and Practice of Parallel Programming》.2013,

审查员 辛小霞

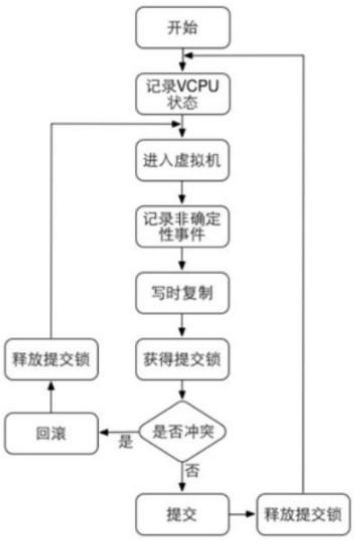
权利要求书2页 说明书7页 附图2页

(54)发明名称

一种多核虚拟机确定性重演的方法

(57)摘要

本发明公布了一种多核虚拟机确定性重演的方法，该方法将虚拟机中的虚拟中央处理单元(VCPU)的执行划分为多个执行块，每个执行块的执行过程包括记录阶段和重演阶段，记录阶段包括执行前记录当前虚拟中央处理单元的状态、执行过程中记录操作和执行结束后记录操作；重演阶段包括获得当前执行块的执行窗口、在虚拟机退出的位置设置断点并进入虚拟机、在断点处处理中断和非确定性事件的插入等操作。本发明在现有多核CPU(中央处理单元)架构下，可解决多核虚拟机内存访问的随机性问题，加快多核虚拟机确定性重演的记录速度，同时大大降低记录日志文件的大小。



CN 106033368 B

1.一种多核虚拟机确定性重演的方法，所述方法将虚拟机中的虚拟中央处理单元的指令执行序列划分为多个执行块，每个执行块的执行过程包括记录阶段和重演阶段，具体包括如下步骤：

对于记录阶段：

A.在每个执行块执行之前，记录当前虚拟中央处理单元的状态，执行如下操作：

A1.记录虚拟中央处理单元的寄存器的状态；

A2.导出并记录虚拟中央处理单元的浮点计算单元相关寄存器的状态；

A3.记录虚拟中央处理单元的输入输出高级可编程中断控制器的状态；

A4.记录虚拟中央处理单元相关机器状态寄存器的状态；

B.在每个执行块执行期间，执行如下操作：

B1.记录外部中断；

B2.记录非确定性事件；

B3.通过写时复制方法处理虚拟中央处理单元写的内存页面；

B4.记录虚拟中央处理单元在该执行块中读和写的内存页面；

C.在每个执行块执行结束后，执行如下操作：

C1.获得提交锁；

C2.检测该执行块与其他虚拟中央处理单元的执行是否有冲突；

C3.如果存在冲突，则执行回滚操作；

C4.如果不存在冲突，则执行提交操作；

C5.释放提交锁；

对于重演阶段：

D.在重演阶段，对于B中对应的每一个执行块，执行如下操作：

D1.获得当前执行块的执行窗口；

D2.在虚拟机退出的位置设置断点，并进入虚拟机；

D3.在断点处处理中断和非确定性事件的插入；

D4.重复执行步骤D2～D3，直到该执行块中的全部指令执行结束。

2.如权利要求1所述多核虚拟机确定性重演的方法，其特征是，所述执行块执行结束的时间点为以下三种中的一种：

1)虚拟中央处理单元需要执行输入输出指令时；

2)虚拟中央处理单元需要执行内存映射输入输出指令时；

3)虚拟中央处理单元已经执行指定条数的指令时。

3.如权利要求2所述多核虚拟机确定性重演的方法，其特征是，当所述执行块执行结束的时间点为虚拟中央处理单元需要执行输入输出指令时，通过Intel VT技术提供的虚拟机输入输出指令退出机制完成；当所述执行块执行结束的时间点为虚拟中央处理单元需要执行内存映射输入输出指令时，通过Intel VT技术提供的扩展页表配置失效机制完成；当所述执行块执行结束的时间点为虚拟中央处理单元已经执行指定条数的指令时，通过Intel VT技术提供的虚拟中央处理单元计时器机制完成。

4.如权利要求1所述多核虚拟机确定性重演的方法，其特征是，通过开启磁盘直接内存访问的方法来加快虚拟机访问磁盘的速度，使得系统的性能提高。

5.如权利要求1所述多核虚拟机确定性重演的方法，其特征是，所述多核虚拟机确定性重演的方法用于Linux系统，在步骤A2中所述导出并记录虚拟中央处理单元的浮点计算单元相关寄存器状态之前，首先导出浮点计算单元的状态。

6.如权利要求1所述多核虚拟机确定性重演的方法，其特征是，步骤A4所述记录虚拟中央处理单元相关机器状态寄存器的状态，具体通过Intel VT的机器状态寄存器保存/恢复方法来实现。

7.如权利要求1所述多核虚拟机确定性重演的方法，其特征是，步骤B3通过所述写时复制方法处理虚拟中央处理单元写的内存页面时，提前对该内存页面进行冲突检测，若确定产生冲突，提前提交以避免该冲突的产生，从而减少执行块的回滚。

8.如权利要求1所述多核虚拟机确定性重演的方法，其特征是，步骤C3所述执行回滚操作过程具体如下：

a.销毁B3阶段写时复制的页面；

b.恢复虚拟中央处理单元的扩展页表，使其指向原页面；

c.恢复A1～A4阶段记录的虚拟中央处理单元状态；

d.插入B1阶段记录的所有中断。

9.如权利要求1所述多核虚拟机确定性重演的方法，其特征是，步骤C4所述执行提交操作过程具体如下：

a.将B3阶段写时复制的页面内容拷贝回原页面；

b.恢复虚拟中央处理单元的扩展页表，使其指向原页面；

c.销毁B3阶段写时复制的页面；

d.将B1、B2阶段记录的信息保存在日志文件中；

e.获得唯一的提交序号，将当前执行块的执行约束和虚拟中央处理单元的执行位置保存在日志文件中。

10.如权利要求9所述多核虚拟机确定性重演的方法，其特征是，步骤b延迟操作，在有其他虚拟中央处理单元访问该页面时，再进行步骤b恢复相应扩展页表的操作。

一种多核虚拟机确定性重演的方法

技术领域

[0001] 本发明属于计算机软件技术领域，涉及计算机操作系统与软件系统设计，尤其涉及一种多核虚拟机确定性重演的方法。

背景技术

[0002] 虚拟化技术是一种将物理硬件进行虚拟的抽象和隔离，使得计算在虚拟的环境上运行的一种技术。虚拟机指的是将CPU(中央处理单元)、内存、输入输出设备的虚拟抽象聚合起来，在这个完全隔离环境中运行的完整计算机系统。虚拟机技术能够在较少的牺牲性能的前提下提供良好的隔离型和可控性，使其能够广泛应用在大规模计算、入侵检测、程序调试、高可用性等多个领域。虚拟机技术已经成为目前云计算、信息安全等领域最为基础和重要的技术之一。

[0003] 虚拟机确定性重演指的是通过某种技术手段使得虚拟机在两次执行过程中能够执行完全相同的指令序列，从而获得完全相同的执行结果。该技术在多核程序调试、入侵分析等领域有着广泛和深入的应用。虚拟机确定性重演目前常用的技术一般分为两个部分：a)记录虚拟机在运行时刻所有非确定性的事件内容及发生位置，包括读取TSC(Time Stamp Counter，时间戳计数器)、I/O(输入输出)操作、软/硬中断等；b)在虚拟机的另一次执行中，在记录的位置插入完全同样的非确定性事件。由于虚拟机的其它执行流程中所执行的指令都是确定性的，因此该技术能够满足虚拟机的两次执行能够活的完全相同的执行结果。

[0004] 然而目前的虚拟机确定性重演技术在多核环境下遇到了巨大挑战。当前的主流的多核CPU(中央处理单元)架构中不同的核对相同的内存访问并不会保证访问顺序的一致性，即在虚拟机的两次执行过程中，即使在指定位置插入相同的非确定性事件，两个核对某内存的访问顺序也有可能不同，从而使虚拟机两次执行的最终结果有所不同。目前有一些简单的系统能够用来解决这个问题，然而这些系统均在某些方面有着重大缺陷，包括但不限于以下几方面：

[0005] 一、记录内存访问的日志过大，在双核环境下最少需要40GB/天的日志；

[0006] 二、无法使用硬件虚拟化支持导致性能较差，相比于在物理环境执行慢40倍以上；

[0007] 三、不能支持虚拟机全系统的确定性重演，只能支持对某个进程的确定性重演；

[0008] 四、不能支持精确的确定性重演，重演的结果可能存在一定的误差。

发明内容

[0009] 为了克服上述现有技术的不足，本发明提供一种多核虚拟机确定性重演的方法，在现有多核CPU(中央处理单元)架构下，针对多核内存访问顺序随机性问题，该方法能够记录不确定性事件，实现确定性重演，且记录的日志文件很小，是一种高效的多核确定性软件重演的新的虚拟机执行系统设计方法。

[0010] 为了便于说明，本文约定：采用“VCPU”表示虚拟机中的虚拟中央处理单元。

[0011] 本发明的原理是：将VCPU的指令执行序列人为划分为多个执行块，一般每个执行

块的大小为1万到10万条CPU指令；在记录阶段每个执行块执行之前，记录当前VCPU的相关状态；在每个执行块执行期间，记录所有的不确定性事件，并记录所有读写访问的内存页面，并且使用写时复制技术将所有写的内存页面写到独立的区域中；在每个执行块执行结束后，通过判断该块执行过程中读写的内存页面是否与其他VCPU写的内存页面存在冲突，如果存在冲突则回滚该执行块，恢复执行块执行前的VCPU状态，重新执行该执行块，如果不存在冲突则提交该执行块，记录该执行块的执行次序约束，将执行过程中写的内存页面复制回原地址空间，并将该执行块写的内存页面扩散到其他VCPU，以便其他VCPU进行冲突检测。在重演阶段，执行块严格按照记录阶段的大小划分；在记录阶段对应的VCPU上执行，执行顺序需要满足执行窗口的序号约定；在记录阶段对应位置插入相同的不确定性事件，从而得到完全相同的执行过程和执行结果。

[0012] 本发明提供的技术方案是：

[0013] 一种多核虚拟机确定性重演的方法，将虚拟机中的虚拟中央处理单元(VCPU)的执行划分为多个执行块，每个执行块的执行过程包括记录阶段和重演阶段，具体包括如下步骤：

[0014] 对于记录阶段：

[0015] A.在每个执行块执行之前，记录当前VCPU的状态，执行如下操作：

[0016] A1.记录VCPU的寄存器状态；

[0017] A2.导出并记录VCPU的FPU(浮点计算单元)相关寄存器状态；

[0018] A3.记录VCPU的I/O APIC(输入输出高级可编程中断控制器)状态；

[0019] A4.记录VCPU相关MSR(机器状态寄存器)状态；

[0020] B.在每个执行块执行期间，执行如下操作：

[0021] B1.记录外部中断；

[0022] B2.记录非确定性事件，如RDTSC(Read Time Stamp Counter，读时间戳计数器)指令、生成随机数指令、输入输出指令和虚拟机控制指令；

[0023] B3.通过写时复制方法处理VCPU写的内存页面；

[0024] B4.记录VCPU在该执行块中读和写的内存页面；

[0025] C.在每个执行块执行结束后，执行如下操作：

[0026] C1.获得提交锁；

[0027] C2.检测该执行块与其他VCPU的执行是否有冲突；

[0028] C3.如果存在冲突，则执行回滚操作；

[0029] C4.如果不存在冲突，则执行提交操作；

[0030] C5.释放提交锁；

[0031] 对于重演阶段：

[0032] D.在重演阶段，对于B中对应的每一个执行块，执行如下操作：

[0033] D1.获得当前执行块的执行窗口；

[0034] D2.在虚拟机退出的位置设置断点，并进入虚拟机；

[0035] D3.在断点处处理中断和非确定性事件的插入；

[0036] D4.重复执行D2～D3过程，直到该执行块中的全部指令执行结束。

[0037] 上述多核虚拟机确定性重演的方法中，对VCPU的执行划分的执行块执行结束的时

间点为以下三种中的一种：

[0038] 1)VCPU需要执行I/O(输入输出)指令时；

[0039] 2)VCPU需要执行MMIO(内存映射输入输出)指令时；

[0040] 3)VCPU已经执行指定条数的指令时。

[0041] 针对执行块执行结束的时间点，进一步地，当VCPU需要执行I/O(输入输出)指令时划分执行块执行结束，该划分可以通过Intel VT技术提供的虚拟机I/O指令退出机制完成；当VCPU需要执行MMIO(内存映射输入输出)指令时执行块执行结束，该划分可以通过Intel VT技术提供的扩展页表配置失效(EPT Misconfiguration)机制完成；当VCPU已经执行指定条数的指令时执行块执行结束，该划分可以通过Intel VT技术提供的VCPU计时器(Preemption Timer)机制完成。

[0042] 上述多核虚拟机确定性重演的方法，优选地，可通过采用磁盘DMA(直接内存访问)的方法来加快虚拟机访问磁盘的速度，从而提高系统的性能；具体是开启磁盘DMA(直接内存访问)的支持，将磁盘DMA模块单独作为最高优先级的执行块来处理，DMA激活期间停止其它VCPU执行块的提交工作。

[0043] 上述多核虚拟机确定性重演的方法中，A1中的VCPU的寄存器包括通用寄存器、段寄存器和指令寄存器等。

[0044] 在本发明实施例中，上述多核虚拟机确定性重演的方法用于Linux系统；在A2中导出并记录VCPU的FPU(浮点计算单元)相关寄存器状态之前，需要首先导出FPU的状态，然后才可以记录VCPU的FPU(浮点计算单元)相关寄存器状态。在A4中记录VCPU相关MSR(机器状态寄存器)状态，具体通过Intel VT的MSR保存/恢复方法来实现。

[0045] 上述多核虚拟机确定性重演的方法中，在B1中记录的外部中断包括中断号和响应中断的VCPU号，将外部中断按照其触发的顺序存储，并记录该中断发生时的VCPU执行位置；VCPU的执行位置包括下述三元组：指令寄存器、转移指令寄存器和RCX寄存器。

[0046] 上述多核虚拟机确定性重演的方法中，优选地，在B3处理写时复制页面时，提前对该页面进行冲突检测，如果该页面与其他VCPU在该执行块执行过程中提交的页面有冲突，则在该位置提前截断该执行块，即进入步骤C执行块结束之后的提交操作。该提前检测方法能够在已经确定产生冲突的情况下，提前提交以避免该冲突的产生，能够有效降低冲突执行块的比例，从而减少执行块的回滚。

[0047] 在B4中记录VCPU在该执行块中读和写的内存页面，该内存页面记录以位图方式进行存储：将写过的页面存储在“写位图”中，将读写的页面存储在“读写位图”中。优选地，在B4中，可以通过Intel VT提供的扩展页表硬件辅助方法来完成对读写页面的记录；在开启扩展页表硬件辅助技术后，可以在执行块执行结束后完成扫描扩展页表(EPT)的Access(访问)位和Dirty(写脏)位来获得该执行块中VCPU对内存的读写情况。

[0048] 在C3中，执行回滚操作过程具体如下：

[0049] a.销毁B3阶段写时复制的页面；

[0050] b.恢复VCPU的EPT(扩展页表)使其指向原页面；

[0051] c.恢复A1～A4阶段记录的VCPU状态；

[0052] d.插入B1阶段记录的所有中断；

[0053] 在C4中，执行提交操作过程如下：

[0054] a.将B3阶段写时复制的页面内容拷贝回原页面；

[0055] b.恢复VCPU的EPT(扩展页表)使其指向原页面；

[0056] c.销毁B3阶段写时复制的页面；

[0057] d.将B1、B2阶段记录的信息保存在日志文件中；

[0058] e.获得唯一的提交序号，将当前执行块的执行约束和VCPU的执行位置三元组保存在日志文件中；执行块的执行约束指该执行块需要在哪个执行块之后被执行；

[0059] 上述C4(b)中，优选地，可以延迟恢复写时复制的页面对应的页表内容，在之后有需要的时候(如其他VCPU访问该页面等)再恢复对应页表。根据局部性原理，该技术能够减少在下一个执行块中写时复制页面的数量，从而提高系统的性能。

[0060] 上述在D1中获得当前执行块的执行窗口，具体通过检查系统为当前VCPU维护的执行序号是否大于等于当前执行块的执行序号，如果是，则将系统中每个VCPU的执行序号加1，并获得执行窗口；否则，轮询等待系统的执行序号，直到其满足上述检查条件。

[0061] 在本发明实施例中，D2设置断点采用的是单步执行方法，即单步执行每一条虚拟机指令，直到执行到断点位置。

[0062] 与现有技术相比，本发明的有益效果是：

[0063] 本发明提供一种多核虚拟机确定性重演的方法，在现有多核CPU(中央处理单元)架构下，针对多核内存访问顺序随机性问题，通过本发明提供的技术方案，可以解决多核虚拟机内存访问的随机性问题，加快多核虚拟机确定性重演的记录速度，同时能够大大降低记录日志文件的大小，并且能够以不慢于记录的速度进行重演操作。

附图说明

[0064] 图1是本发明在记录阶段的流程框图。

[0065] 图2是本发明在重演阶段的流程框图。

[0066] 图3是本发明实施例一两个VCPU虚拟机系统中的记录阶段的一段指令运行片段；

[0067] 图4是本发明实施例二两个VCPU虚拟机系统中的重演阶段的一个执行片段；

[0068] 图3～图4中，两个VCPU名称分别为VCPU 0和VCPU 1；虚线框表示执行块的边界。

具体实施方式

[0069] 下面结合附图，通过实施例进一步描述本发明，但不以任何方式限制本发明的范围。

[0070] 本发明提供一种多核虚拟机确定性重演的方法，该方法将虚拟机中的虚拟中央处理单元(VCPU)的执行划分为多个执行块，每个执行块的执行过程包括记录阶段和重演阶段。

[0071] 在Linux系统上，对VCPU的执行划分的执行块，其执行结束的时间点为以下三种中的一种：

[0072] 1)VCPU需要执行I/O(输入输出)指令时，该划分可以通过Intel VT技术提供的虚拟机I/O指令退出机制完成；

[0073] 2)VCPU需要执行MMIO(内存映射输入输出)指令时，该划分可以通过Intel VT技术提供的扩展页表配置失效(EPT Misconfiguration)机制完成；

2)；

[0118] 4)VCPU 0首先结束执行块，检测冲突；由于此时没有其他执行块提交，不存在冲突，提交该执行块，将执行过程中复制的页面A写回到公共内存中；VCPU 0开始下一个执行块的执行；将VCPU 0写的页面(页面A)扩散给VCPU 1；

[0119] 5)VCPU 1随后结束执行块，检测冲突；由于VCPU 0刚刚写过内存页面A，与VCPU 1在该执行块中写的内存页面A冲突，则VCPU 1该执行块需要回滚；回滚需要如下三个操作：

[0120] a.摧毁VCPU 1上一个执行块写时复制的页面1；

[0121] b.恢复执行块执行之前记录的VCPU状态；

[0122] c.重新插入外部中断2；

[0123] 6)随后VCPU 1重新运行该执行块，并写时复制内存页面A；此次执行与VCPU 0的执行没有冲突，可以提交；

[0124] 7)该运行片段结束，VCPU 0与VCPU 1继续运行其后的执行块。

[0125] 实施例二：

[0126] 本实施例在一个拥有两个VCPU的虚拟机系统中使用本发明的确定性重演系统，以下描述的是在重演阶段的一个执行片段，见图4所示。两个VCPU分别命名为VCPU 0和VCPU 1；虚线框表示执行块的边界；该片段的执行情况如下：

[0127] 1)VCPU 0首先等待满足执行约束，其首先满足执行约束并运行执行块1；

[0128] 2)在执行块1运行过程中需要在适当位置退出虚拟机执行，并插入外部中断等异步事件；

[0129] 3)VCPU 1中执行块2的执行约束需要满足在执行块1结束之后，所以VCPU 1需要等待知道VCPU 0的执行块1执行结束才能够开始执行块2的运行；

[0130] 4)执行块1执行结束后，VCPU 0检测执行块3的执行约束，其满足执行约束，即可与执行块2并行执行；

[0131] 5)该运行片段结束，VCPU 0与VCPU 1继续运行其后的执行块。

[0132] 需要注意的是，公布实施例的目的在于帮助进一步理解本发明，但是本领域的技术人员可以理解：在不脱离本发明及所附权利要求的精神和范围内，各种替换和修改都是可能的。因此，本发明不应局限于实施例所公开的内容，本发明要求保护的范围以权利要求书界定的范围为准。

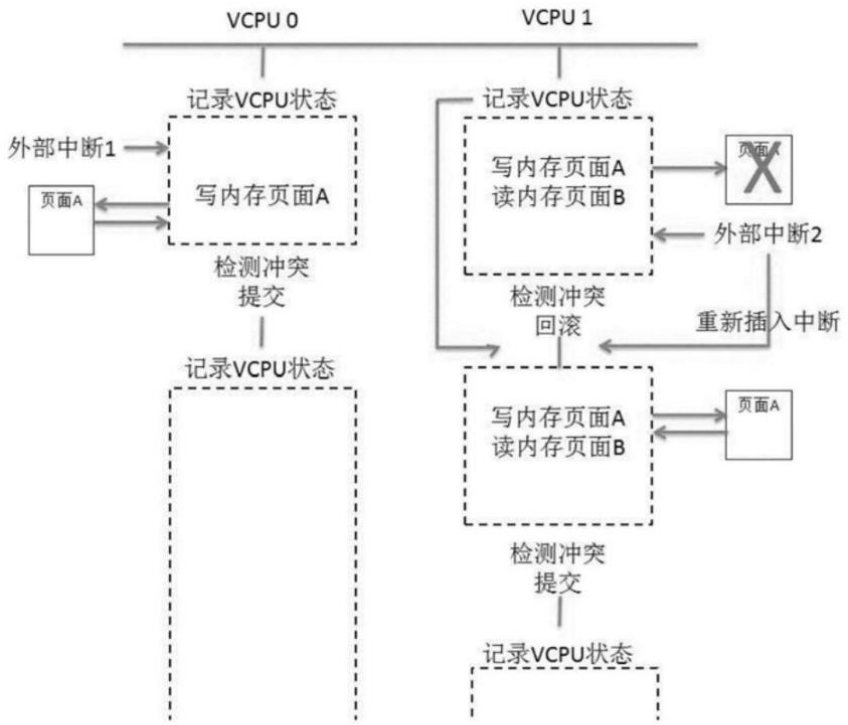

VCPU 0
VCPU 1
记录VCPU状态
外部中断1
写内存页面A
页面A
检测冲突
提交
记录VCPU状态
记录VCPU状态
写内存页面A
读内存页面B
外部中断2
检测冲突
回滚
重新插入中断
写内存页面A
读内存页面B
检测冲突
提交
记录VCPU状态

图3

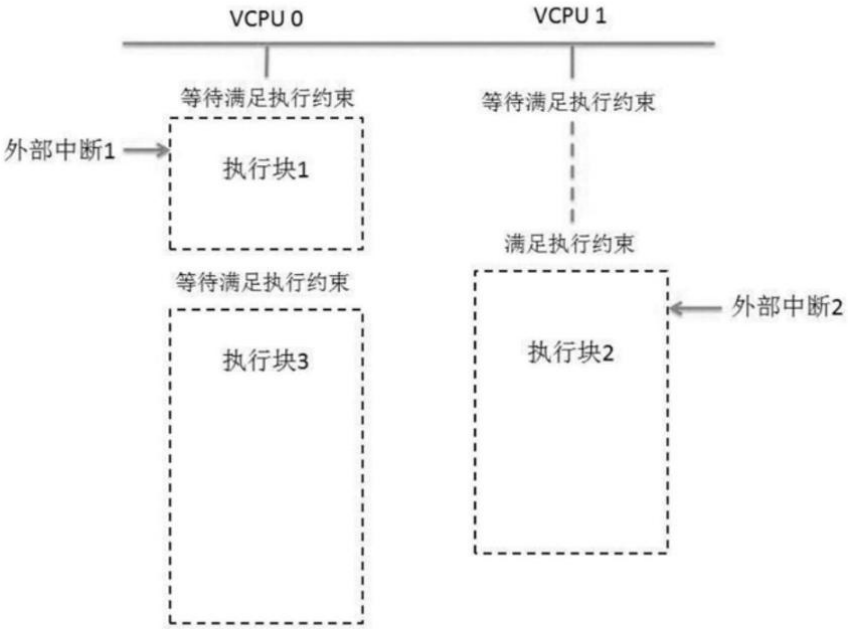

VCPU 0
VCPU 1
等待满足执行约束
外部中断1
执行块1
等待满足执行约束
执行块3
等待满足执行约束
满足执行约束
外部中断2
执行块2

图4