(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 103729248 B
(45)授权公告日 2017.12.15

(21)申请号 201210392519.X

(22)申请日 2012.10.16

(65)同一申请的已公布的文献号
申请公布号 CN 103729248 A

(43)申请公布日 2014.04.16

(73)专利权人 华为技术有限公司
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼
专利权人 中国科学院计算技术研究所

(72)发明人 徐远超 范东睿 张浩 叶笑春

(51)Int.Cl.
*G06F 9/50*(2006.01)

(56)对比文件
US 2012/0180061 A1,2012.07.12,
CN 101419561 A,2009.04.29,
CN 102483703 A,2012.05.30,
CN 101706755 A,2010.05.12,
CN 102184125 A,2011.09.14,

审查员 唐佩

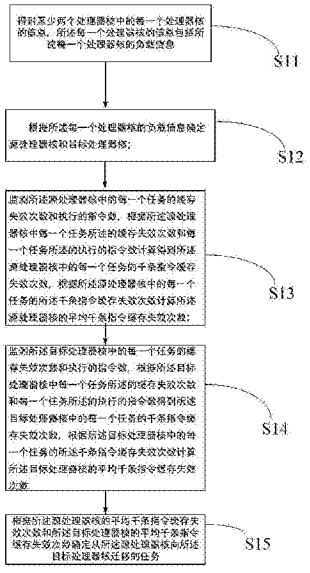
权利要求书4页 说明书16页 附图4页

(54)发明名称

一种基于缓存感知的确定待迁移任务的方法和装置

(57)摘要

本发明公开了一种基于缓存感知的确定待迁移任务的方法，包括：根据每一个处理器核的负载确定源处理器核和目标处理器核；监测源处理器核和目标处理器核中的每一个任务的缓存失效次数和执行的指令数，得到源处理器核和目标处理器核中的每一个任务的千条指令缓存失效次数；得到源处理器核和目标处理器核的平均千条指令缓存失效次数；根据源处理器核的平均千条指令缓存失效次数和目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务。根据本发明实施例的确定待迁移任务的方法，可以让操作系统感知程序的行为，从而在任务迁移时选择更加合理的任务。本发明还公开了一种基于任务感知确定待迁移任务的装置。




CN 103729248 B

1.一种基于缓存感知的确定待迁移任务的方法，其特征在于，所述方法包括：

得到至少两个处理器核中的每一个处理器核的信息，所述每一个处理器核的信息包括所述每一个处理器核的负载信息；

根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核；

监测所述源处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数计算得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的所述千条指令缓存失效次数计算所述源处理器核的平均千条指令缓存失效次数；

监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述目标处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，根据所述目标处理器核中的每一个任务的所述千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数；

根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务，包括：当所述源处理器核的平均千条指令缓存失效次数不小于所述目标处理器核的平均千条指令缓存失效次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最小的任务，将所述千条指令缓存失效次数最小的任务迁移到所述目标处理器核。

2.如权利要求1所述的确定待迁移任务的方法，其特征在于，所述根据每一个所述处理器核的负载信息确定源处理器核和目标处理器核包括：

将所述至少两个处理器核按照预设规则分为至少两个的调度组；

周期性的监测所述调度组的状态，所述调度组的状态包括所述调度组中每一个处理器核的负载；

根据所述调度组的状态得到负载最大的调度组，根据所述负载最大的调度组中每一个处理器核的负载得到负载最大的处理器核；

若所述调度组之间存在负载不均衡，则将所述负载最大的调度组中所述负载最大的处理器核确定为所述源处理器核，将正在监测的处理器核确定为所述目标处理器核。

3.如权利要求1或2所述的确定待迁移任务的方法，其特征在于，监测所述源处理器核的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得到所述源处理器核中的每一个任务的千条指令缓存失效次数包括：

在所述源处理器核中创建一个任务时，设置所述缓存失效次数计数器的初始值和指令计数器的初始值；所述源处理器核运行所述任务时，所述缓存失效次数计数器和所述指令计数器开始计数；

所述任务暂停运行时，暂停所述缓存失效次数计数器的计数和所述指令计数器的计数；根据所述缓存失效次数计数器的计数值得到任务的缓存失效次数，根据所述指令计数器的计数值得到任务的指令数；

根据所述任务的缓存失效次数和所述任务的指令数得到所述任务的千条指令缓存失

效次数；

重复上述步骤直至所述源处理器核中的全部任务处理完毕。

4.如权利要求1所述的确定待迁移任务的方法，其特征在于，所述根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务包括：当所述源处理器核的平均千条指令缓存次数小于所述目标处理器核的平均千条指令缓存次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最大的任务，将所述千条指令缓存失效次数最大的任务迁移到所述目标处理器核。

5.如权利要求1所述的确定待迁移任务的方法，其特征在于，得到所述源处理器核中的每一个任务的千条指令缓存失效次数，得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，还可以采用下列方式：

根据预测的每千条指令缓存失效次数值得到所述源处理器核中的每一个任务的千条指令缓存失效次数或所述目标处理器核中的每一个任务的千条指令缓存失效次数；其中，所述预测的每千条指令缓存失效次数是根据监测所述任务的缓存失效次数和所述任务的指令数得到的每千条指令缓存失效次数的当前值和暂存的每千条指令缓存失效次数的历史值，通过指数平滑公式计算得到。

6.如权利要求1所述的确定待迁移任务的方法，其特征在于，在根据所述源处理核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务之前，所述方法还包括：

所述处理器核的信息包括所述处理器核的物理CPU状态；

如果所述源处理器核和所述目标处理器核在不同的物理CPU上，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

7.如权利要求1所述的确定待迁移任务的方法，其特征在于，在根据所述源处理核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务之前，所述方法还包括：

所述处理器核的信息包括所述处理器核的性能，根据所述处理器核的性能判断所述处理器核是慢核或快核；

如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器是慢核，所述目标处理器是快核，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

8.如权利要求7所述的确定待迁移任务的方法，其特征在于，所述方法还包括：如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器是快核，所述目标处理器是慢核，则将所述源处理器核中千条指令缓存失效次数最大的任务确定为迁移到所述目标处理器核的待迁移任务。

9.一种基于缓存感知的确定待迁移任务的装置，其特征在于，所述装置包括：至少两个处理器核，得到所述至少两个处理器核中的每一个处理器核的信息，所述每一个处理器核的信息包括所述每一个处理器核的负载信息，根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核；

缓存失效次数生成模块，所述缓存失效次数生成模块监测所述源处理器核中的每一个

任务的缓存失效次数和执行的指令数，根据所述源处理器核中的每一个任务的所述缓存失效次数和所述指令数得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的千条指令缓存失效次数计算所述源处理器核的平均千条指令缓存失效次数；所述缓存失效次数生成模块监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述目标处理器核中的每一个任务的所述缓存失效次数和所述指令数得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，根据所述目标处理器核中的每一个任务的千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数；

任务迁移模块，用于根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务，包括：当所述源处理器核的平均千条指令缓存次数不小于所述目标处理器核的平均千条指令缓存次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最小的任务，将所述千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

10.如权利要求9所述的确定待迁移任务的装置，其特征在于，根据处理器核的负载情况确定源处理器核和目标处理器核包括：

将所述至少两个处理器核按照预设规则分为至少两个的调度组；

周期性的监测所述调度组的状态，所述调度组的状态包括所述调度组中每一个处理器核的负载信息；

根据所述调度组的状态得到负载最大的调度组，根据所述负载最大的调度组中每一个处理器核的负载信息得到负载最大的处理器核；

若所述调度组之间存在负载不均衡，则将所述负载最大的调度组中所述负载最大的处理器核确定为所述源处理器核，将正在监测的处理器核确定为所述目标处理器核。

11.如权利要求9或10所述的确定待迁移任务的装置，其特征在于，监测所述源处理器核的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中的每一个任务的所述缓存失效次数和所述指令数得到所述源处理器核中的每一个任务的千条指令缓存失效次数包括：

在所述源处理器核中创建一个任务时，根据所述任务的信息设置缓存失效次数计数器的初始值和指令计数器的初始值；所述源处理器核运行所述任务时，所述任务的缓存失效次数计数器和所述任务的指令计数器开始计数；

所述任务暂停运行时，暂停所述任务的缓存失效次数计数器的技术和所述任务的指令计数器的计数；根据所述任务的缓存失效次数计数器的计数值得到任务的缓存失效次数，根据所述任务的指令计数器的计数值得到任务的指令数；根据所述任务的缓存失效次数和所述任务的指令数得到所述任务的千条指令缓存失效次数；

重复上述步骤直至所述源处理器核中的全部任务处理完毕。

12.如权利要求9所述的确定待迁移任务的装置，其特征在于，所述装置还包括：

当所述源处理器核的平均千条指令缓存次数小于所述目标处理器核的平均千条指令缓存次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最大的任务，将所述千条指令缓存失效次数最大的任务确定为

迁移到所述目标处理器核的待迁移任务。

13.如权利要求9所述的确定待迁移任务的装置，其特征在于，所述装置还包括：

所述处理器核的信息包括所述处理器核的物理CPU状态；

如果所述源处理器核和所述目标处理器核在不同的物理CPU上，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

14.如权利要求9所述的确定待迁移任务的装置，其特征在于，所述装置还包括：

所述处理器核的信息包括所述处理器核的性能，根据所述处理器核的性能判断所述处理器核是慢核或快核；

如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器核是慢核，所述目标处理器核是快核，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

15.如权利要求14所述的确定待迁移任务的装置，其特征在于，所述装置还包括：

如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器核是快核，所述目标处理器核是慢核，则将所述源处理器核中千条指令缓存失效次数最大的任务确定为迁移到所述目标处理器核的待迁移任务。

# 一种基于缓存感知的确定待迁移任务的方法和装置

## 技术领域

[0001] 本发明涉及计算机科学技术领域，尤其涉及一种基于缓存感知的确定待迁移任务的方法和系统。

## 背景技术

[0002] 任务调度是操作系统的核心功能之一，任务调度的好坏直接影响着程序运行的性能、公平性、以及实时性等等。对于只具有单个处理器核的操作系统而言，任务调度只需要解决不同任务之间的切换问题。而对于具有多个处理器核的操作系统而言，除了调度不同任务之间的切换外，还需要处理多个任务在多个处理器核上的分配以及任务在多个处理器核之间的迁移过程，以保证多个处理器核之间的负载均衡。

[0003] 在具有多个处理器核的操作系统中，多个任务需要争用共享缓存、存储器控制器、内存总线等诸多共享资源，不同的任务对资源的需求不尽相同，如果处理器在任务调度时不对上述资源加以考虑，就会造成部分资源(例如共享cache、共享访存等)争用时而其他处理器核的资源却没有被充分利用，从而对整个系统的性能产生不利影响。

[0004] 现有技术对任务调度的处理中，默认的是操作系统在迁移任务时按照优先级从高到低的顺序从包含有任务的最高优先级链表尾部顺序选择允许迁移的任务，并不分析待迁移任务的程序以及对该迁移目标处理器核上任务的影响，因此，整个系统的性能及服务质量无法得到保证，如果待迁移任务不适合在目标处理器核上运行，则系统的性能会变得很糟糕。

## 发明内容

[0005] 为解决上述问题，本发明实施例提供了一种基于缓存感知的确定待迁移任务的方法和装置，可以让操作系统感知程序的行为，从而在任务迁移时选择更加合理的任务，降低对处理器资源的争用，提高整个系统的性能。

[0006] 本发明一方面实施例公开了一种基于缓存感知的确定待迁移任务的方法，所述方法包括：

[0007] 得到至少两个处理器核中的每一个处理器核的信息，所述每一个处理器核的信息包括所述处理器核的负载信息；

[0008] 根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核；

[0009] 监测所述源处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数计算得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的所述千条指令缓存失效次数计算所述源处理器核的平均千条指令缓存失效次数；

[0010] 监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述目标处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得

到所述目标处理器核中的每一个任务的千条指令缓存失效次数，根据所述目标处理器核中的每一个任务的所述千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数；

[0011] 根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务。

[0012] 在本发明的第一方面的实施例的一种可能实现的方式中，所述根据处理器核的负载情况确定源处理器核和目标处理器核包括：

[0013] 将所述至少两个处理器核按照预设规则分为至少两个的调度组；

[0014] 周期性的监测所述调度组的状态，所述调度组的状态包括所述调度组中每一个处理器核的负载；

[0015] 根据所述调度组的状态得到负载最大的调度组，根据所述负载最大的调度组中每一个处理器核的负载得到负载最大的处理器核；

[0016] 若所述调度组之间存在负载不均衡，则将所述负载最大的调度组中所述负载最大的处理器核确定为所述源处理器核，将正在监测的处理器核确定为所述目标处理器核。

[0017] 结合本发明第一方面实施例和第一种可能实现的方式的第二种可能实现的方式中，监测所述源处理器核的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得到所述源处理器核中的每一个任务的千条指令缓存失效次数包括：

[0018] 在所述源处理器核中创建一个任务时，设置缓存失效次数计数器的初始值和指令计数器的初始值；所述源处理器核运行所述任务时，所述缓存失效次数计数器和所述指令计数器开始计数；

[0019] 所述任务暂停运行时，暂停所述缓存失效次数计数器的计数和所述指令计数器的计数；根据所述缓存失效次数计数器的计数值得到任务的缓存失效次数，根据所述指令计数器的计数值得到任务的指令数；

[0020] 根据所述任务的缓存失效次数和所述任务的指令数得到所述任务的千条指令缓存失效次数；

[0021] 重复上述步骤直至所述源处理器核中的全部任务处理完毕。

[0022] 结合上述本发明第一方面实施例的第三种可能实现的方式中，所述根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述目标处理器核向所述源处理器核迁移的任务包括：

[0023] 当所述源处理器核的平均千条指令缓存次数不小于所述目标处理器核的平均千条指令缓存次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最小的任务，将所述源处理器核中千条指令缓存失效次数最小的任务迁移到所述目标处理器核。

[0024] 结合上述本发明第一方面实施例的第四种可能实现的方式中，当所述源处理器核的平均千条指令缓存次数小于所述目标处理器核的平均千条指令缓存次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最大的任务，将所述源处理器核中千条指令缓存失效次数最大的任务迁移到所述目标处理器核。

[0025] 结合上述本发明第一方面实施例的第五种可能实现的方式中，得到所述源处理器核中的每一个任务的千条指令缓存失效次数，得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，还可以采用下列方式：

[0026] 根据预测的每千条指令缓存失效次数值得到所述源处理器核中的每一个任务的千条指令缓存失效次数或所述目标处理器核中的每一个任务的千条指令缓存失效次数。根据监测所述任务的缓存失效次数和所述任务的指令数得到的每千条指令缓存失效次数的当前值和暂存的每千条指令缓存失效次数的历史值，通过指数平滑公式计算得到所述预测的每千条指令缓存失效次数。

[0027] 结合上述本发明第一方面实施例的第六种可能实现的方式中，在根据所述源处理核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务之前，所述方法还包括：所述处理器核的信息包括所述处理器核的物理CPU状态；如果所述源处理器核和所述目标处理器核在不同的物理CPU上，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

[0028] 结合上述本发明第一方面实施例的第七种可能实现的方式中，在根据所述源处理核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务之前，所述方法还包括：所述处理器核的信息包括所述处理器核的性能，根据所述处理器核的性能判断所述处理器核是慢核或快核；如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器是慢核，所述目标处理器是快核，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

[0029] 结合上述本发明第一方面实施例的第八种可能实现的方式中，所述方法还包括：如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器是快核，所述目标处理器是慢核，则将所述源处理器核中千条指令缓存失效次数最大的任务确定为迁移到所述目标处理器核的待迁移任务。

[0030] 根据本发明实施例提供的基于缓存感知的确定待迁移任务的方法，可以让操作系统感知程序的行为，从而在任务迁移时选择更加合理的任务，降低资源争用的概率，提高整个系统的性能；同时可以让操作系统在迁移任务时能够结合源处理器核和目标处理器核所在的拓扑位置，从而选择合适的线程迁移。另外本发明的实施例除可以用于同构多核环境外，对性能不对称的异构多核处理器同样可以适用。

[0031] 本发明第二方面的实施例公开了一种基于缓存感知的确定待迁移任务的装置，所述装置包括：

[0032] 至少两个处理器核，得到所述至少两个处理器核中的每一个处理器核的信息，所述每一个处理器核的信息包括所述每一个处理器核的负载信息，根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核；

[0033] 缓存失效次数生成模块，所述缓存失效次数生成模块监测所述源处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中的每一个任务的所述缓存失效次数和所述指令数得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的千条指令缓存失效次数计算所述源处理器核的平

均千条指令缓存失效次数;所述缓存失效次数生成模块监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数,根据所述目标处理器核中的每一个任务的所述缓存失效次数和所述指令数得到所述目标处理器核中的每一个任务的千条指令缓存失效次数,根据所述目标处理器核中的每一个任务的千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数;

[0034] 任务迁移模块,用于根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务。

[0035] 其中处理器核和缓存失效次数生成模块相连,任务迁移模块分别和处理器核以及令缓存失效次数生成模块相连。

[0036] 根据本发明实施例提供的基于缓存感知的确定待迁移任务的装置,可以让操作系统感知程序的行为,从而在任务迁移时选择更加合理的任务,降低资源争用的概率,提高整个系统的性能;同时可以让操作系统在迁移任务时能够结合源处理器核和目标处理器核所在的拓扑位置,从而选择合适的线程迁移。另外本发明的实施例除可以用于同构多核环境外,对性能不对称的异构多核处理器同样可以适用。

[0037] 在本发明的第二方面的实施例的一种可能实现的方式中,将所述至少两个处理器核按照预设规则分为至少两个的调度组;

[0038] 周期性的监测所述调度组的状态,所述调度组的状态包括所述调度组中每一个处理器核的负载信息;

[0039] 根据所述调度组的状态得到负载最大的调度组,根据所述负载最大的调度组中每一个处理器核的负载信息得到负载最大的处理器核;

[0040] 若所述调度组之间存在负载不均衡,则将所述负载最大的调度组中所述负载最大的处理器核确定为所述源处理器核,将正在监测的处理器核确定为所述目标处理器核。

[0041] 结合本发明第二方面实施例和第二方面实施第一种可能实现的方式的第二种可能实现的方式中,在所述源处理器核中创建一个任务时,根据所述任务的信息设置缓存失效次数计数器的初始值和指令计数器的初始值;所述源处理器核运行所述任务时,所述任务的缓存失效次数计数器和所述任务的指令计数器开始计数;

[0042] 所述任务暂停运行时,暂停所述任务的缓存失效次数计数器的技术和所述任务的指令计数器的计数;根据所述任务的缓存失效次数计数器的计数值得到任务的缓存失效次数,根据所述任务的指令计数器的计数值得到任务的指令数;根据所述任务的缓存失效次数和所述任务的指令数得到所述任务的千条指令缓存失效次数;重复上述步骤直至所述源处理器核中的全部任务处理完毕。

[0043] 结合上述实施例的本发明第二方面实施例的第三种可能实现的方式中,当所述源处理器核的平均千条指令缓存次数不小于所述目标处理器核的平均千条指令缓存次数时,根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最小的任务,将所述千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

[0044] 结合上述实施例的本发明第二方面实施例的第四种可能实现的方式中,当所述源处理器核的平均千条指令缓存次数小于所述目标处理器核的平均千条指令缓存次数时,根

据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最大的任务，将所述千条指令缓存失效次数最大的任务确定为迁移到所述目标处理器核的待迁移任务。

[0045] 结合上述实施例的本发明第二方面实施例的第五种可能实现的方式中，所述处理器核的信息包括所述处理器核的物理CPU状态；如果所述源处理器核和所述目标处理器核在不同的物理CPU上，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

[0046] 结合上述实施例的本发明第二方面实施例的第六种可能实现的方式中，所述处理器核的信息包括所述处理器核的性能，根据所述处理器核的性能判断所述处理器核是慢核或快核；如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器核是慢核，所述目标处理器核是快核，则将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的待迁移任务。

[0047] 结合上述实施例的本发明第二方面实施例的第七种可能实现的方式中，如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器核是快核，所述目标处理器核是慢核，则将所述源处理器核中千条指令缓存失效次数最大的任务迁移到所述目标处理器核。

[0048] 本发明的实施例还提供了一种基于缓存感知的确定待迁移任务的装置，所述装置包括CPU和存有可执行程度代码的存储器，所述可执行代码可用于得到至少两个处理器核中的每一个处理器核的信息，所述每一个处理器核的信息包括所述处理器核的负载信息；

[0049] 根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核；

[0050] 监测所述源处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数计算得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的所述千条指令缓存失效次数计算所述源处理器核的平均千条指令缓存失效次数；

[0051] 监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述目标处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，根据所述目标处理器核中的每一个任务的所述千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数；

[0052] 根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务。

## 附图说明

[0053] 为了更清楚地说明本发明实施例的技术方案，下面将对本发明实施例中所需要使用的附图作简单地介绍，显而易见地，下面所描述的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0054] 图1为本发明实施例的一种基于缓存感知的确定待迁移任务的方法的流程图。

[0055] 图2为根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法的具体实现的流程图。

[0056] 图3为根据本发明实施例的划分调度域和调度组的示意图。

[0057] 图4为根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法的具体示例。

[0058] 图5为根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法的另一种具体示例。

[0059] 图6为根据本发明实施例的一种基于缓存感知的确定待迁移任务的装置的结构图。

## 具体实施方式

[0060] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，可以理解的是，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0061] 任务迁移是指在具有多个处理器核的系统中，由于处理器核上的任务数出现不均衡现象，为了均衡不同处理器核之间的负载，需要将运行任务数较多的处理器核上的一部分任务迁移到任务数较少的处理器核上。缓存感知是指操作系统在进行任务迁移调度决策时，将与缓存相关的参数，如任务的缓存失效次数等作为决策依据。通过捕获任务在一段时间内的缓存失效次数然后除以该时间段内运行的指令数，再放大1000倍便得到该任务该段时间内的每千条指令缓存失效次数(cache Miss Per Kilo instructions，MPKI)，由于通常情况下缓存失效次数绝对数值很小，所以采用MPKI以便比较不同的处理器核的缓存失效次数。

[0062] 下面结合图1具体描述根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法。

[0063] 如图1所示，根据本发明实施例的基于缓存感知的确定待迁移任务的方法，包括：

[0064] S11：得到至少两个处理器核中的每一个处理器核的信息，所述每一个处理器核的信息包括所述每一个处理器核的负载信息。

[0065] S12：根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核。

[0066] S13：监测所述源处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数计算得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的所述千条指令缓存失效次数计算所述源处理器核的平均千条指令缓存失效次数。

[0067] S14：监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述目标处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，根据所述目标处理器核中的每一个任务的所述千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数。

[0068] S15:根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务。

[0069] 可以理解的是,上述计算源处理器核和目标处理器核的平均千条指令缓存失效次数并不包含有顺序的限制。既可以先计算源处理器核的平均千条指令缓存失效次数也可以先计算目标处理器核的平均千条指令缓存失效次数,或者两者并行计算。

[0070] 下面结合图1具体描述根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法。

[0071] 根据本发明实施例的基于缓存感知的确定待迁移任务的方法,包括:

[0072] 一、确定源处理器核和目标处理器核。

[0073] S11:得到至少两个处理器核中的每一个处理器核的信息,所述每一个处理器核的信息包括所述每一个处理器核的负载信息;

[0074] 在本发明的一个实施例中,操作系统周期性的监测每一个所述处理器核的状态,得到所述处理器核的信息,所述处理器核的信息包括所述处理器核的负载、快慢以及拓扑位置。

[0075] S12:根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核;

[0076] 在本步骤中,操作系统得到至少二个处理器核中的每一个处理器核的信息,所述处理器核的信息包括所述处理器核的负载,根据每一个所述处理器核的负载确定源处理器核和目标处理器核。

[0077] 若所述处理器核之间的负载不均衡,则将负载最大的所述处理器核确定为所述源处理器核,将负载最小的所述处理器核确定为所述目标处理器核。

[0078] 在本发明的一个实施例中,在Linux操作系统下,Linux 2.6内核定义的规则是,当系统中存在一个处理器核的负载超过另一个处理器核负载的25%时,将启动任务迁移,此时将负载低的处理器核作为目标处理器核,负载高的处理器核作为源处理器核。

[0079] 在本发明的一个实施例中,每一个处理器核都会以自己为中心,查找是否存在其他处理器核的负载大于自身(即以自己为中心的处理器核)负载25%以上,如果存在,则认为处理器核之间存在负载不均衡,就需要迁移任务,选择其中负载最大的处理器核为源处理器核,自身就是目标处理器核,源处理器核上的任务从源处理器核迁移到目标处理器核。该过程将周期性地遍历所有处理器核。

[0080] 在本发明的一个实施例中,将全部所述处理器核按照预设规则分为至少两个的调度组;

[0081] 操作系统周期性的监测所述调度组的状态,所述调度组的状态包括所述调度组中每一个处理器核的负载。

[0082] 根据所述调度组的状态得到负载最大的调度组,根据所述调度组中每一个处理器核的负载得到负载最大的处理器核。

[0083] 若所述调度组之间存在负载不均衡,则将负载最大的调度组中负载最大的处理器核确定为所述源处理器核,将正在监测的处理器核确定为所述目标处理器核。

[0084] 在本发明的一个实施例中,系统通过迭代遍历的方式寻找源处理器核和目标处理器核,当发现不同调度组之间的负载不均衡时,以一个处理器核为中心,先遍历和该处理器核不相同调度组的处理器核,寻找负载最大的调度组中负载最大的处理器核作为源处理器

核，如果没有找到，则继续在与该处理器核相同调度组的处理器核，寻找负载最大的处理器作为源处理器核，而将位于中心的处理器核作为目标处理器核。如果位于中心的处理器核在相同的调度组中也没有找到符合条件的负载最大的处理器核，则继续将下一个处理器核作为中心处理器核继续遍历。

[0085] 如图3所示，调度域(Scheduling Domain)是指具有相同属性的一组处理器核的集合，并且根据超线程(Hyper-threading)，多核(Multi-core)，多处理器(SMP)，非一致性内存访问结构(NUMA)这样的系统结构划分成不同的级别。不同级之间通过指针链接在一起，形成一种树状的关系。

[0086] Linux将所有同一级别的处理器核归为一个“调度组”，然后将同一级别的所有的调度组组成一个“调度域”，负载均衡在调度域的各个调度组之间进行。Linux在基于调度域进行负载均衡的时候采用的是自下而上的遍历方式，这样就优先在对高速缓存(cache)影响最小的处理器核之间进行负载均衡，有力的阻止了对高速缓影响很大的处理器核之间的负载均衡。调度域是从对高速缓存影响最小的最底层向高层构建的。调度域分为Numa_domain，Phy_d omain，Core_domain，SMT_domain(Cpu_domain)四个等级。可以理解的是，本图所示调度组的划分只是为了帮助理解本发明而做出的一种示例，而不是对处理器核调度组划分的一种限制。

[0087] 二、监测任务的缓存失效次数并计算MPKI。

[0088] S13：监测所述源处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数计算得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的所述千条指令缓存失效次数计算所述源处理器核的平均千条指令缓存失效次数。

[0089] S14：监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述目标处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，根据所述目标处理器核中的每一个任务的所述千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数。

[0090] 在本发明的一个实施例中，在所述源处理器核或目标处理器核中创建一个任务时，设置缓存失效次数计数器的初始值和指令计数器的初始值；所述源处理器核或目标处理器核运行所述任务时，所述缓存失效次数计数器和所述指令计数器开始计数；所述任务暂停运行时，暂停所述缓存失效次数计数器的计数和所述指令计数器的计数；根据所述缓存失效次数计数器的计数值得到任务的缓存失效次数，根据所述指令计数器的计数值得到任务的指令数，根据所述任务的缓存失效次数和所述任务的指令数得到所述任务的千条指令缓存失效次数；重复上述步骤直至所述源处理器核或目标处理器核中的全部任务处理完毕。

[0091] 在本发明的一个实施例中，缓存失效次数计数器或指令计数器既可以用硬件实现也可以用软件实现。计数器既可以伴随着任务的创建而创建，也可以是由于任务的创建调用而来。可以理解的是，本发明实施例有关计数器的举例并不是对本发明技术方案的一种限制，计数器的使用方式也包括本领域普通技术人员无需创造性劳动即可实现的其它方

式。

[0092] 在本发明的一个实施例中，任务暂停运行可以指的是任务的挂起状态。这种挂起状态既可以是由于时间片耗尽产生的，也可以是由缺乏运行任务所需要的资源而产生的。可以理解的是，任务暂停运行也包括本领域普通技术人员无需创造性劳动即可实现的其它方式。

[0093] 在本发明的一个实施例中，根据预测的每千条指令缓存失效次数值得到所述源处理器核中的每一个任务的千条指令缓存失效次数或所述目标处理器核中的每一个任务的千条指令缓存失效次数。根据监测所述任务的缓存失效次数和所述任务的指令数得到的每千条指令缓存失效次数的当前值和暂存的每千条指令缓存失效次数的历史值，通过指数平滑公式计算得到所述预测的每千条指令缓存失效次数。

[0094] 对于尚未执行的指令代码，如果不提前进行离线分析，一般很难准确知晓其具体行为。通常是在代码执行过程中在线收集监测数据，通过监测数据来预测程序未来一个时间片或一段时间内的行为，随着代码的不断运行，预测数据也不断更新，操作系统正是基于这些预测数据来进行调度决策的，在本发明的一个实施例中，本发明使用一次指数平滑公式预测每个任务最新的MPKI。$MPKI_{n+1}(i,k) = (1-\alpha)*MPKI_n(i,k)+\alpha*T_n(i,k)$，其中，$MPKI_n(i,k)$为任务$T_i^k$在第n个时间片的每千条指令cache失效次数，$T_n(i,k)$为第n个时间片的平均每千条指令cache失效次数。a是一个常数加权因子$(0\leqslant\alpha\leqslant1)$，用于调节$MPKI_n(i,k)$与$T_n(i,k)$的权重系数。在本发明的一个实施例中定义$a=0.8$，此时最近的4个观测值对预测值起主导作用，这与程序的时间局部性原理和空间局部性原理是吻合的。可以理解的是a的定义不仅限于本发明实施例的给出的数值，而是包括其它本领域普通技术人员不需要经过创造性劳动即可实现的其它方式。

[0095] 在本发明的一个实施例中，监测源处理器核或目标处理器核的每个任务的缓存失效次数并计算MPKI，需要给进程描述符增加变量用于记录每个任务的缓存失效次数以及提交的指令数，以便用于计算MPKI。缓存失效次数的测量可以通过硬件性能计数器完成，本发明提供的一个实施例可以通过改写Perfctr驱动程序实现在Linux内核中实现对每个任务缓存失效次数的在线测量，在一个任务创建时，首先设置计数事件，并启动计数，当该任务切换时，暂停前一任务的计数事件，并将结果记录进程描述符中新增的变量中，同时开启下一个任务的计数事件。计算任务的MPKI需要两个参数，除了该任务的缓存失效次数，还需要测量每个任务的指令数，因此可以设置两个计数事件data_cac he_miss和retired_instructions，data_cache_miss用来测量任务的缓存失效次数，retired_instructions用来测量任务的指令数。

[0096] 在本发明的一个实施例中，在系统刚刚启动的过程中，处理器核内没有任务，此时的运行队列的平均缓存失效次数为0，当任务创建的时候，操作系统会从父进程复制相应的内容给子进程，而此时并不能够确定父子进程的缓存失效次数是否相关，所以在任务初始化时将缓存失效次数置为0。

[0097] 对于每个处理器核而言，准确地的计算出每个运行队列所有任务的平均MPKI是调度决策的关键，当任务发生切换时，运行队列的平均MPKI就发生了改变，然而如果每一次任务切换就更新运行队列的平均MPKI，开销会很大，周期性地更新运行队列的平均MPKI又无法保证任务迁移时的MPKI是最新的，为此，本发明实施例提供的方法是采取当有任务需要

迁移时，才重新计算源处理器核和目标处理器核运行队列的平均MPKI。

[0098] 三、任务迁移。

[0099] S15：根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务。

[0100] 在本发明的一个实施例中，当所述源处理器核的平均千条指令缓存次数小于所述目标处理器核的平均千条指令缓存次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最大的任务，将所述源处理器核中千条指令缓存失效次数最大的任务确定为迁移到所述目标处理器核的任务。

[0101] 在本发明的一个实施例中，当所述源处理器核的平均千条指令缓存次数不小于所述目标处理器核的平均千条指令缓存次数时，根据所述源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最小的任务，将所述源处理器核中千条指令缓存失效次数最小的任务确定为迁移到所述目标处理器核的任务。

[0102] 在本发明的一个实施例中，任务迁移分为两种：拉任务和推任务。推任务发生在强制迁移或拉任务失败时，拉任务是主要的任务迁移方式。从当前处理器核的角度看，如果是把自己的任务迁移到其他处理器核上，就是推任务，如果是将其他处理器核上的任务迁移到当前处理器核上，就是拉任务。

[0103] 在当前处理器核完全空闲时，即运行队列为空，没有任务在该处理器核上执行，造成这样的原因可能是系统刚刚初始化或当前处理器核上的任务全部运行结束，此时当前处理器核的平均缓存失效次数为0，可以迁移MPKI较大的任务，原因在于减少了源处理器核内缓存失效次数较大的任务数量，提高了缓存的公平使用，所以选择向空核迁移缓存失效次数较大的程序。

[0104] 当前处理器核上仍有任务在运行，操作系统内核会周期性的判断处理器核之间的负载是否均衡，如果不均衡，本发明实施例提供的缓存感知调度方法会在此时遍历当前处理器核和源处理器核的运行队列，更新平均MPKI值，随后迁移任务，力求待迁移任务与当前处理器核的平均MPKI最接近且与源处理器核的平均MPKI相差最远。

[0105] 在本发明的一个实施例中，具体的任务迁移规则为，当源处理器核的平均MPKI $\overline{MPKI}$ 大于等于目标处理器核的平均MPKI $\overline{MPKI}$ 时，优先迁移源处理器核上MPKI小的任务；当源处理器核的 $\overline{MPKI}$ 小于目标处理器核的 $\overline{MPKI}$ 时，优先迁移源处理器核上MPKI大的任务。推导过程如下：

[0106] 如果系统中出现了负载不均衡，编号为src的处理器核负载较小，需要从其他核上迁移任务，经过find_busiest_group以及find_busiest_cpu操作后，找到了负载最大的核dst，此时需要从src(源处理器核)上迁移部分任务到dst(目标处理器核)上。

[0107] 在本发明的一个实施例中，Linux现有调度算法将所有处理器核分成多个组，首先寻找负载最大的组，然后从负载最大的组中寻找负载最大的处理器核。寻找负载最大的组由Linux内核中find_busiest_group函数完成，从某一个组中寻找负载最大的处理器核由find_busiest_cpu函数完成。

[0108] 可以理解的是，find_busiest_group以及find_busiest_cpu是在Linux系统下为寻找负载最大的处理器核而设置的函数。在其它操作系统，例如IOS、Wind ows、Unix操作系统下可以采用别的函数寻找负载最大的处理器核，本发明实施例所使用的函数只是对解释

本发明所做的一种示例，而不是对本发明的一种限制。

[0109] 寻找待迁移任务的流程如下：

[0110] a.使用公式(1)计算源处理器核src上的每个任务$T_{src}^{k}$与目标处理器核dst的距离，使用公式(2)计算源处理器核src上的每个任务$T_{src}^{k}$与源处理器核src的距离。

[0111] $$D_{src}^{k}=\left|MPKI(src,k)-\overline{MPKI(dst)}\right|,k\in[1,N_{src}] \quad (1)$$

[0112] $$D_{src}^{k}=\left|MPKI(src,k)-\overline{MPKI(src)}\right|,k\in[1,N_{src}] \quad (2)$$

[0113] b.待迁移任务除满足默认调度策略can_migrate_task规定的约束条件外，还需要满足公式(3)，即要求任务$T_{src}^{m}$与目标处理器核的距离尽可能小同时与源处理器核的距离尽可能大，即使公式(3)的目标函数取最大值。

[0114]

$$m=\underset{k\in[1,N_{task}(src)]}{\arg\max}\left(\left[MPKI(src,k)-\overline{MPKI(src)}\right]^{2}-\left[MPKI(src,k)-\overline{MPKI(dst)}\right]^{2}\right) \quad (3)$$

[0115] 对公式(3)针对变量MPKI(src,k)进行求导，可以看到这是一个单调递增或递减函数，取决于目标处理器核与源处理器核的$\overline{MPKI}$孰大孰小，最大值出现在MPKI(src,k)最大或最小的时候。当$\overline{MPKI(src)}\geq\overline{MPKI(dst)}$，即源处理器核的$\overline{MPKI}$大于等于目标处理器核的$\overline{MPKI}$时，MPKI(src,k)取值最小才使得公式(3)中的目标函数取最大值，当$\overline{MPKI(src)}<\overline{MPKI(dst)}$时，即源处理器核的$\overline{MPKI}$小于目标处理器核的$\overline{MPKI}$时，MPKI(src,k)取值最大才使得公式(3)中的目标函数取最大值。当允许迁移多个任务时，再依照以上规则寻找次接近任务。

[0116] 在本发明的一个实施例中，默认调度策略can_migrate_task是指：在选择待迁移任务时，有几类任务是不允许被选择的：

[0117] 当前正在执行的进程；

[0118] 通过cpus_allowed明确表示不能迁移该CPU的进程；

[0119] 被原来CPU切换下来且时间间隔小于cache_decay_ticks的任务，这说明cache仍然是活跃的；

[0120] 评价某个任务是否属于以上几种就是由函数can_migrate_task函数完成的。

[0121] 可以理解的是，上述公式只是为了更加清楚的解释本发明实施例而给出的一种示例，而不是对本发明的实施例的一种限制。

[0122] 在本发明的一个实施例中，在步骤S15之前还包括：

[0123] 拓扑感知。

[0124] 拓扑感知是指操作系统在进行调任务迁移的度决策时，需要考虑源处理器核和目标处理器核所在的相对位置，如两者是否共享二级缓存，是否在同一个芯片内，或者两个处理器核是否位于不同的处理器等。

[0125] 在本发明的一个实施例中，在Linux系统环境下，Linux内核在启动过程中可自动识别处理器核的物理位置关系，即识别出哪些处理器核在一个物理CPU上，以及哪些处理器核在另外一个物理CPU上。

[0126] 首先判断所述源处理器核和所述目标处理器核是否在相同的物理CPU上，如果所述源处理器核和所述目标处理器核在不相同的物理CPU上，此时并不进行源处理器核和目标处理器核之间的MPKI的比较，而是直接将源处理器核中千条指令缓存失效次数最小的任

务迁移到所述目标处理器核。

[0127] 在本发明的一个实施例中，如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器是慢核，所述目标处理器是快核，此时并不进行源处理器核和目标处理器核之间的MPKI的比较，而是将所述源处理器核中千条指令缓存失效次数最小的任务迁移到所述目标处理器核。

[0128] 在本发明的一个实施例中，慢核与快核可以根据处理器核的时钟频率高低、有序或乱序执行、缓存大小等区分。例如处理器核的时钟频率高于某一值时将该处理器核认定为快核，处理器核的时钟频率低于某一值时将该处理器核认定为慢核。可以理解的是，快核与慢核的区分并不限于本实施例的举例，也包括任何本领域普通技术人员无需创造性劳动即可实现的区分快核和慢核的方法。

[0129] 在本发明的一个实施例中，如果所述源处理器核和所述目标处理器核在相同的物理CPU上，所述源处理器是快核，所述目标处理器是慢核，此时并不进行源处理器核和目标处理器核之间的MPKI的比较，而是将所述源处理器核中千条指令缓存失效次数最大的任务迁移到所述目标处理器核。

[0130] 图3描述了根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法的实现。如图3所示，首先以当前处理器核所在的调度组为中心，从整个调度域中寻找最繁忙的调度组。然后再判断符合条件的调度组是否存在，如不存在，则转向下一个处理器核，将其作为当前处理器核重新开始寻找最繁忙的调度组。

[0131] 如果符合条件的调度组存在，则继续在最繁忙的调度组中寻找最繁忙(可以是负载最大)的处理器核，如果该调度组中所有处理器核的负载都不超过预设的范围，则可以认为该组不满足预设的条件，则转向下一个处理器核重新开始遍历搜索；否则将符合条件的处理器核确定为源处理器核，当前处理器核为目标处理器核。具体寻找调度域和调度组的方式可以采用如图4所示的方法。

[0132] 然后判断源处理器核与目标处理器核是否在同一个物理CPU上，若两者不在同一个物理CPU上，则选择源处理器核中MPKI最小的任务准备迁移；在正式迁移前需要判断系统是否允许该任务迁移，判断的原则可以依据前面所述的默认调度策略can_migrate_task；如果判断的结果是该任务不允许被迁移，则继续选择下一个任务重复上述判断，如果判断的结果是允许该任务迁移，则选择该任务作为待迁移任务。

[0133] 如果源处理器核和目标处理器核在同一个物理CPU上，则继续判断目标处理器核与源处理器核的主频高低，主频高的可以被认为是快核，主频低的可以被认为是慢核；

[0134] 当目标处理器是快核，源处理器核是慢核时，此时选择源处理器核中MP KI值最小的任务准备迁移，在正式迁移前需要判断系统是否允许该任务迁移，判断的原则可以依据前面所述的默认调度策略can_migrate_task；如果判断的结果是该任务不允许被迁移，则继续选择下一个任务重复上述判断，如果判断的结果是允许该任务迁移，则选择该任务作为待迁移任务。

[0135] 当目标处理器是慢核，源处理器核是快核时，此时选择源处理器核中MP KI值最大的任务准备迁移，在正式迁移前需要判断系统是否允许该任务迁移，判断的原则可以依据前面所述的默认调度策略can_migrate_task；如果判断的结果是该任务不允许被迁移，则继续选择下一个任务重复上述判断，如果判断的结果是允许该任务迁移，则选择该任务作

为待迁移任务。

[0136] 当源处理核和目标处理器核的主频相等时，此时需要判断目标处理器核的平均MPKI和源处理器核的平均MPKI：

[0137] 当目标处理器核的平均MPKI小于源处理器核的平均MPKI时，根据源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最小的任务，将千条指令缓存失效次数最小的任务迁移到目标处理器核，在正式迁移前需要判断系统是否允许该任务迁移，判断的原则可以依据前面所述的默认调度策略can_migrate_task；如果判断的结果是该任务不允许被迁移，则继续选择下一个任务重复上述判断，如果判断的结果是允许该任务迁移，则选择该任务作为待迁移任务。

[0138] 当目标处理器核的平均MPKI大于或等于源处理器核的平均MPKI时，根据源处理器核中的每一个任务的千条指令缓存失效次数得到源处理器核中千条指令缓存失效次数最大的任务，将千条指令缓存失效次数最大的任务迁移到目标处理器核，在正式迁移前需要判断系统是否允许该任务迁移，判断的原则可以依据前面所述的默认调度策略can_migrate_task；如果判断的结果是该任务不允许被迁移，则继续选择下一个任务重复上述判断，如果判断的结果是允许该任务迁移，则选择该任务作为待迁移任务。

[0139] 图4和图5给出了根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法的具体示例。

[0140] 如图4所示，CPU1运行有3个任务，CPU2运行有2个任务，CPU3运行有4个任务，其中CPU1上的任务1-1的MPKI的值是10，任务1-2的MPKI的值是8，任务1-3的MPKI的值是6，则CPU1的平均MPKI值为10+8+6/3＝8；同理求得CPU2的平均MPKI值为10，CPU3的平均MPKI值为9。

[0141] 若此时CPU1的负载为250，CPU2的负载为300，CPU3的负载为500，则将CPU1作为目标处理器核，CPU3作为源处理器核，开始准备任务迁移。因为CPU3的平均MPKI大于CPU1的平均MPKI，则在CPU3中选择MPKI最小的任务迁移到CPU1上，即将任务3-1迁移到CPU1上。

[0142] 若此时CPU1的负载为500，CPU2的负载为300，CPU3的负载为350，则将CPU2作为目标处理器核，CPU1作为源处理器核，开始准备迁移。因为CPU1的平均MPKI小于CPU2的平均MPKI，则在CPU1中选择MPKI最大的任务迁移到CPU2上，即将任务1-1迁移到CPU2上。

[0143] 如图5所示，CPU1和CPU2在同一个物理CPU-A上，CPU3和CPU4在另一个物理CPU-B上，且CPU1的主频大于CPU2，可以认为CPU1为快核，CPU2为慢核。CPU1运行有3个任务，CPU2运行有2个任务，CPU3运行有4个任务，其中CPU1上的任务1-1的MPKI的值是10，任务1-2的MPKI的值是8，任务1-3的MPKI的值是6，则CPU1的平均MPKI值为10+8+6/3＝8；同理求得CPU2的平均MPKI值为10，CPU3的平均MPKI值为9。

[0144] (1)当CPU1作为目标处理器核，CPU3作为源处理器核时，因为CPU1和CPU3在不同的物理CPU上，此时将CPU3中MPKI值最小的任务迁移到CPU1上，选择CPU3中MPKI值最小的任务3-1作为待迁移任务。

[0145] (2)当CPU1作为目标处理器核，CPU2作为源处理器核时，因为CPU1和CPU2在相同的物理CPU-A上，且CPU2为慢核，CPU1为快核，此时将CPU2中MPKI值最小的任务2-2作为待迁移任务。

[0146] (3)当CPU2作为目标处理器核，CPU1作为源处理器核时，因为CPU1和CPU2在相同的

物理CPU-A上，且CPU2为慢核，CPU1为快核，此时将CPU1中MPKI值最大的任务1-1作为待迁移任务。

[0147] 根据本发明实施例的一种基于缓存感知的确定待迁移任务的方法，先监测每个任务的MPKI，再通过聚合算法最终让缓存失效次数相似的任务基本聚合到一个处理器核上，这样不仅可以保证缓存失效次数相似的任务不会同时运行，从而争用共享缓存和访存的带宽资源，也避免了缓存失效次数较大的任务对缓存失效次数较小的任务产生的不良影响，保证了处理器核内私有缓存的公平使用。

[0148] 下面根据图6描述根据本发明实施例的一种基于缓存感知的确定待迁移任务的装置60。

[0149] 如图6所示，根据本发明实施例的一种基于缓存感知的确定待迁移任务的装置60包括：至少两个处理器核601，根据处理器核601的负载确定源处理器核601-1和目标处理器核601-2；缓存失效次数生成模块602，缓存失效次数生成模块602分别监测源处理器核601-1和目标处理器核601-2中的每一个任务的缓存失效次数和执行的指令数，得到源处理器核601-1和目标处理器核601-2中的每一个任务的千条指令缓存失效次数；用于根据源处理器核601-1和目标处理器核601-2中的每一个任务的千条指令缓存失效次数分别得到源处理器核601-1和目标处理器器核601-2的平均千条指令缓存失效次数；任务迁移模块603，用于根据源处理器核601-1的平均千条指令缓存失效次数和目标处理器核601-2的平均千条指令缓存失效次数确定从源处理器核601-1向目标处理器核601-2迁移的任务。

[0150] 其中处理器核601和缓存失效次数生成模块602相连，任务迁移模块603分别和处理器核601以及缓存失效次数生成模块602相连。

[0151] 根据本发明实施例的一种基于缓存感知的确定待迁移任务的装置60，先监测每个任务的MPKI，再通过聚合算法最终让缓存失效次数相似的任务基本聚合到一个处理器核上，这样不仅可以保证缓存失效次数相似的任务不会同时运行，从而争用共享缓存和访存的带宽资源，也避免了缓存失效次数较大的任务对缓存失效次数较小的任务产生的不良影响，保证了处理器核内私有缓存的公平使用。

[0152] 在本发明的一个实施例中，将全部处理器核601按照预设规则分为至少两个的调度组；

[0153] 周期性的监测所述调度组的状态，所述调度组的状态包括所述调度组中每一个处理器核601的负载；

[0154] 根据所述调度组的状态得到负载最大的调度组，根据所述调度组中每一个处理器核601的负载得到负载最大的处理器核；

[0155] 若调度组之间存在负载不均衡，则将所述负载最大的调度组中所述负载最大的处理器核确定为源处理器核601-1，将正在监测的处理器核确定为目标处理器核601-2。

[0156] 在本发明的一个实施例中，在源处理器核601-1中创建一个任务时，设置任务的缓存失效次数计数器的初始值和任务的指令计数器的初始值；源处理器核601-1运行所述任务时，所述任务的缓存失效次数计数器和所述任务的指令计数器开始计数；

[0157] 所述任务暂停运行时，暂停所述任务的缓存失效次数计数器的技术和所述任务的指令计数器的计数；根据所述任务的缓存失效次数计数器的计数值得到任务的缓存失效次数，根据所述任务的指令计数器的计数值得到任务的指令数；根据所述任务的缓存失效次

数和所述任务的指令数得到所述任务的千条指令缓存失效次数；

[0158] 重复上述步骤直至源处理器核601-1中的全部任务处理完毕。

[0159] 在本发明的一个实施例中，当源处理器核601-1的平均千条指令缓存次数不小于目标处理器核601-2的平均千条指令缓存次数时，根据源处理器核601-1中的每一个任务的千条指令缓存失效次数得到源处理器核601-1中千条指令缓存失效次数最小的任务，将千条指令缓存失效次数最小的任务迁移到目标处理器核601-1。

[0160] 在本发明的一个实施例中，当源处理器核601-1的平均千条指令缓存次数小于目标处理器核601-2的平均千条指令缓存次数时，根据源处理器核601-1中的每一个任务的千条指令缓存失效次数得到源处理器核601-1中千条指令缓存失效次数最大的任务，将千条指令缓存失效次数最大的任务迁移到目标处理器核601-2。

[0161] 在本发明的一个实施例中，操作系统得到的处理器核601的信息包括处理器核601的物理CPU状态；如果源处理器核601-1和目标处理器核601-2在不同的物理CPU上，则将源处理器核601-1中千条指令缓存失效次数最小的任务迁移到目标处理器核601-2。

[0162] 在本发明的一个实施例中，操作系统得到的处理器核601的信息包括处理器核601的性能，根据处理器核601的性能判断处理器核601是慢核或快核；如果源处理器核601-1和目标处理器核601-2在相同的物理CPU上，源处理器核601-1是慢核，目标处理器核601-2是快核，则将源处理器核601-1中千条指令缓存失效次数最小的任务迁移到目标处理器核601-2。

[0163] 在本发明的一个实施例中，如果源处理器核601-1和目标处理器核601-2在相同的物理CPU上，源处理器核601-1是快核，目标处理器核601-2是慢核，则将源处理器核601-1中千条指令缓存失效次数最大的任务迁移到目标处理器核601-2。

[0164] 根据本发明实施例的一种基于缓存感知的确定待迁移任务的装置60，先监测每个任务的MPKI，再通过聚合算法最终让缓存失效次数相似的任务基本聚合到一个处理器核上，这样不仅可以保证缓存失效次数相似的任务不会同时运行，从而争用共享缓存和访存的带宽资源，也避免了缓存失效次数较大的任务对缓存失效次数较小的任务产生的不良影响，保证了处理器核内私有缓存的公平使用。

[0165] 在本发明的一个实施例中，上述的操作既可以依靠多个处理器核完成，也可以是在一个或多个处理器核上依赖软件程序完成。例如可以是在一个处理器核上利用可执行程序控制所有处理器核之间的任务迁移。

[0166] 所属领域的技术人员可以清楚地了解到，为描述的方便和简洁，上述描述的装置的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

[0167] 在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

[0168] 另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。

[0169] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备(可以是个人计算机，服务器，或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U盘、移动硬盘、只读存储器(ROM，Read-Only Memory)、随机存取存储器(RAM，Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0170] 以上所述，仅为本发明较佳的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到的变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应该以权利要求的保护范围为准。

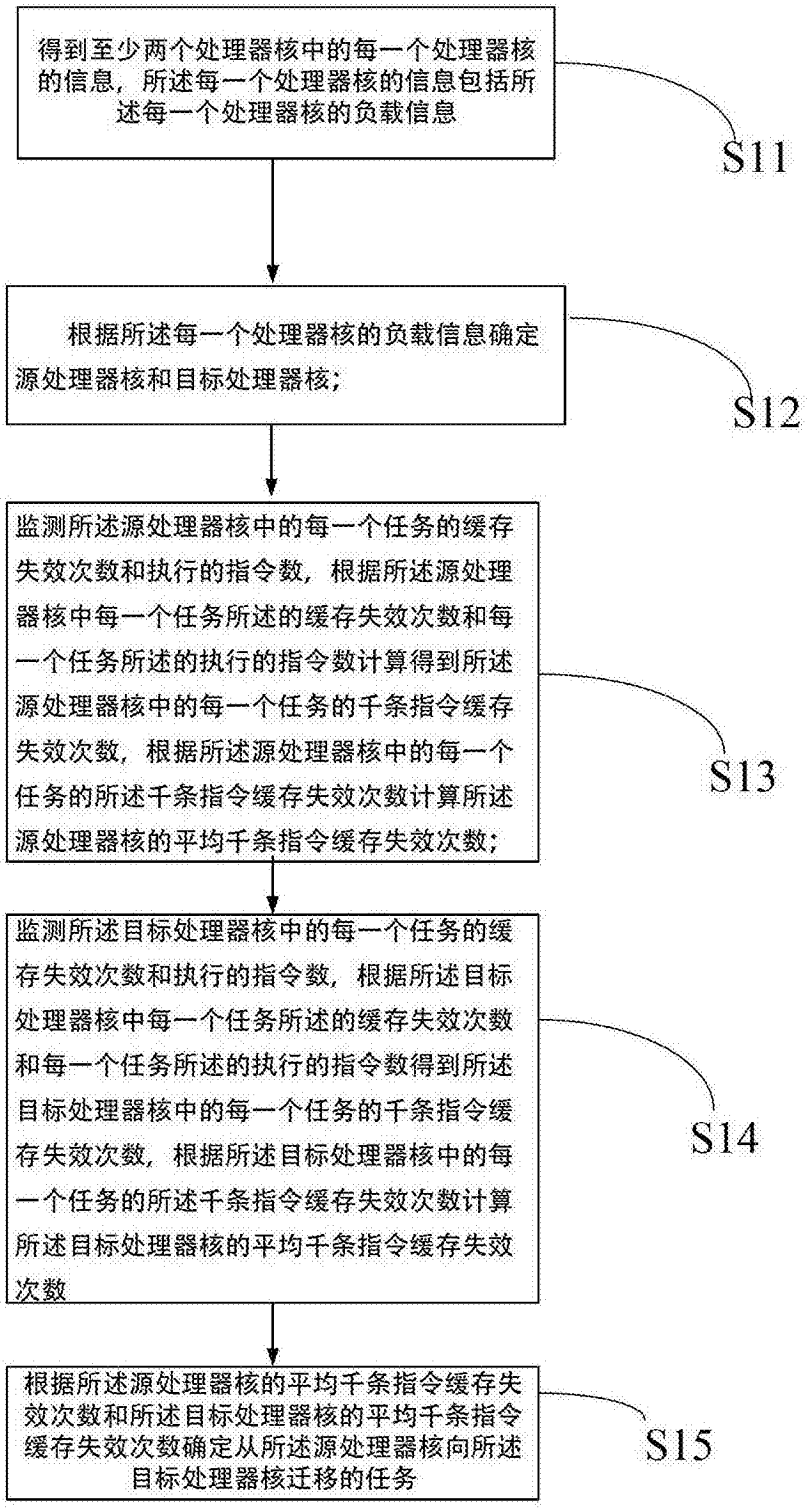

得到至少两个处理器核中的每一个处理器核的信息，所述每一个处理器核的信息包括所述每一个处理器核的负载信息
S11
根据所述每一个处理器核的负载信息确定源处理器核和目标处理器核；
S12
监测所述源处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述源处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数计算得到所述源处理器核中的每一个任务的千条指令缓存失效次数，根据所述源处理器核中的每一个任务的所述千条指令缓存失效次数计算所述源处理器核的平均千条指令缓存失效次数；
S13
监测所述目标处理器核中的每一个任务的缓存失效次数和执行的指令数，根据所述目标处理器核中每一个任务所述的缓存失效次数和每一个任务所述的执行的指令数得到所述目标处理器核中的每一个任务的千条指令缓存失效次数，根据所述目标处理器核中的每一个任务的所述千条指令缓存失效次数计算所述目标处理器核的平均千条指令缓存失效次数
S14
根据所述源处理器核的平均千条指令缓存失效次数和所述目标处理器核的平均千条指令缓存失效次数确定从所述源处理器核向所述目标处理器核迁移的任务
S15

图1

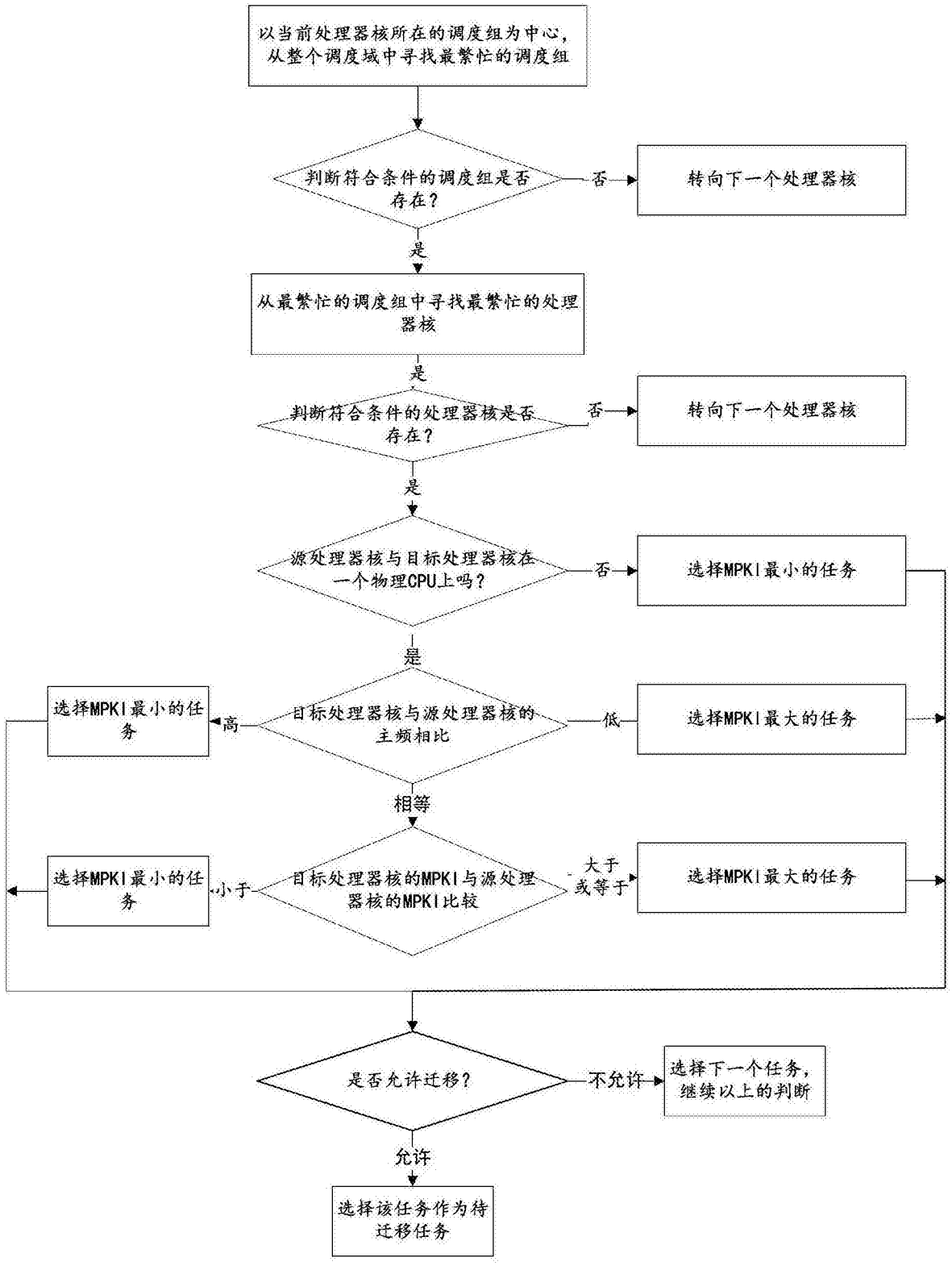

以当前处理器核所在的调度组为中心，从整个调度域中寻找最繁忙的调度组
判断符合条件的调度组是否存在？
否
转向下一个处理器核
是
从最繁忙的调度组中寻找最繁忙的处理器核
是
判断符合条件的处理器核是否存在？
否
转向下一个处理器核
是
源处理器核与目标处理器核在一个物理CPU上吗？
否
选择MPKI最小的任务
是
目标处理器核与源处理器核的主频相比
高
选择MPKI最小的任务
低
选择MPKI最大的任务
相等
目标处理器核的MPKI与源处理器核的MPKI比较
小于
选择MPKI最小的任务
大于或等于
选择MPKI最大的任务
是否允许迁移？
不允许
选择下一个任务，继续以上的判断
允许
选择该任务作为待迁移任务

图2

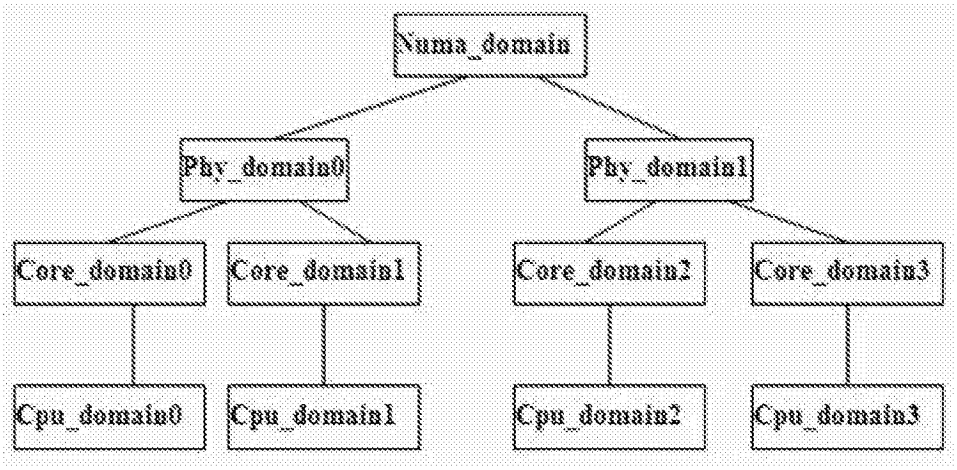

Numa_domain
Phy_domain0
Phy_domain1
Core_domain0
Core_domain1
Core_domain2
Core_domain3
Cpu_domain0
Cpu_domain1
Cpu_domain2
Cpu_domain3

图3

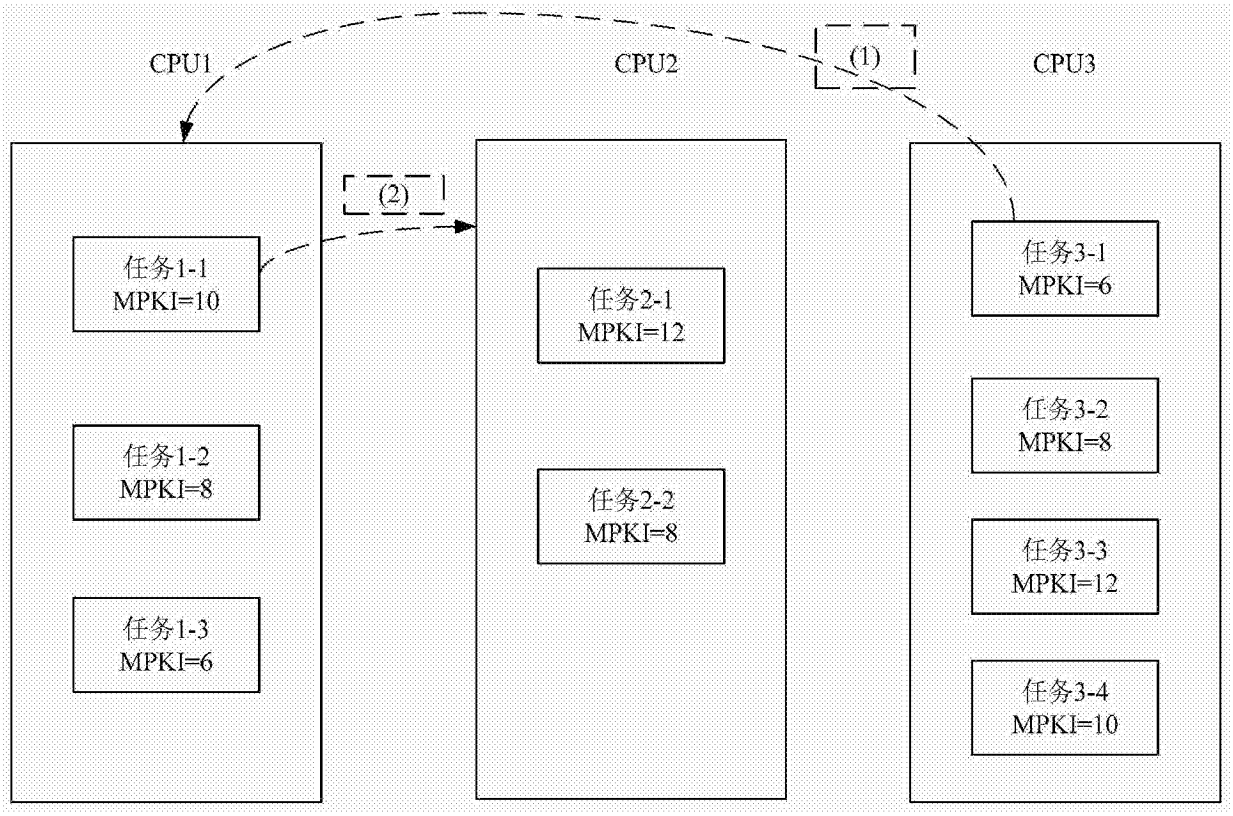

CPU1
CPU2
CPU3
(1)
(2)
任务1-1
MPKI=10
任务1-2
MPKI=8
任务1-3
MPKI=6
任务2-1
MPKI=12
任务2-2
MPKI=8
任务3-1
MPKI=6
任务3-2
MPKI=8
任务3-3
MPKI=12
任务3-4
MPKI=10

图4

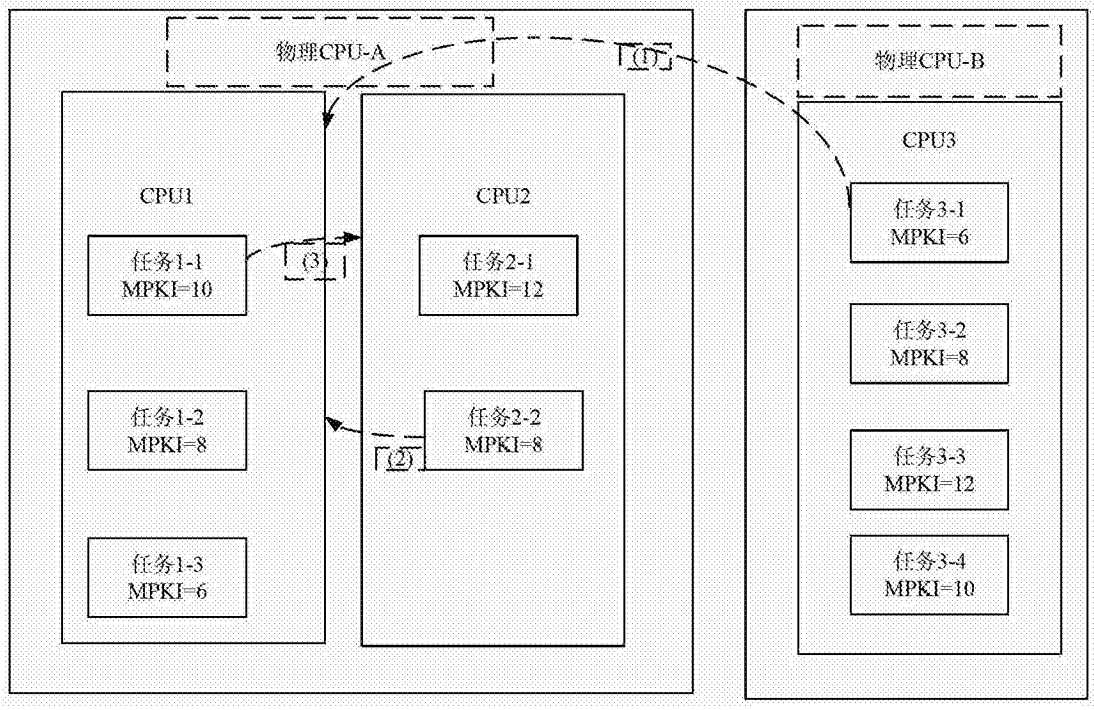

物理CPU-A
物理CPU-B
CPU1
CPU2
CPU3
任务1-1
MPKI=10
任务1-2
MPKI=8
任务1-3
MPKI=6
任务2-1
MPKI=12
任务2-2
MPKI=8
任务3-1
MPKI=6
任务3-2
MPKI=8
任务3-3
MPKI=12
任务3-4
MPKI=10
(1)
(2)
(3)

图5

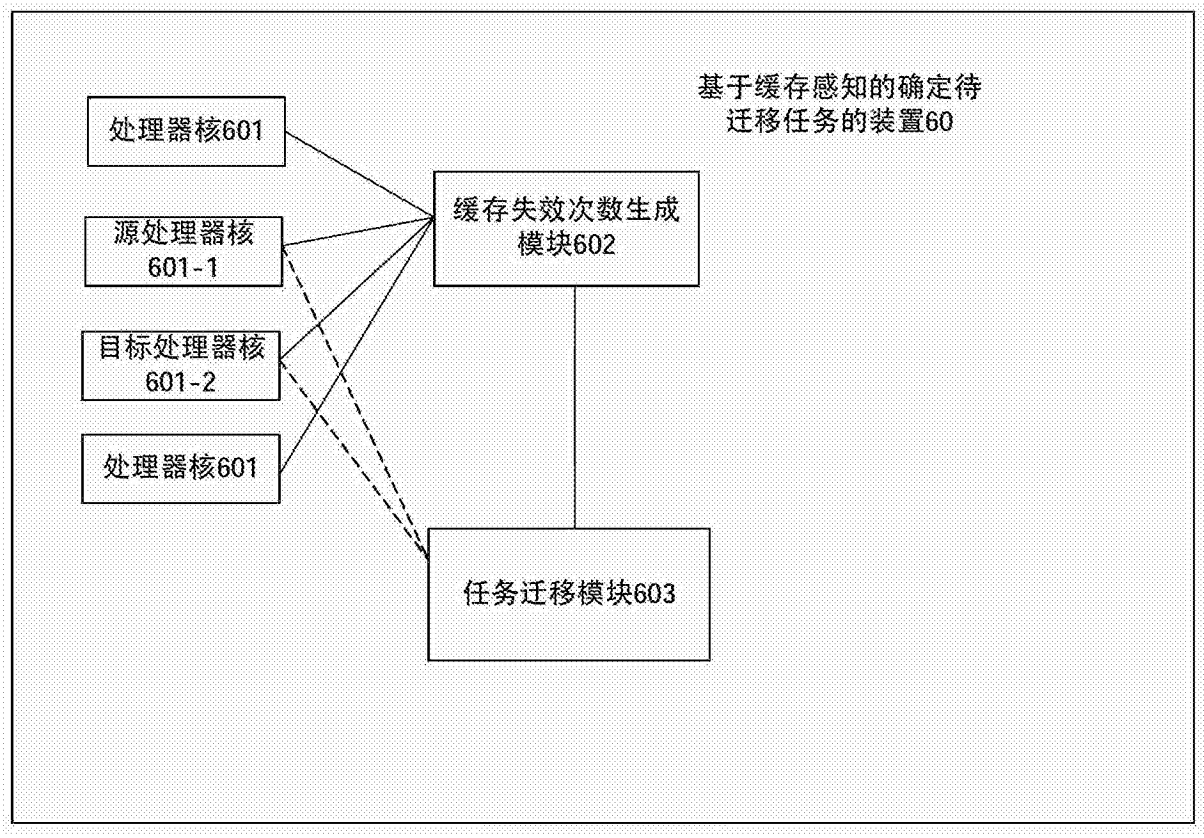

基于缓存感知的确定待迁移任务的装置60
处理器核601
源处理器核601-1
目标处理器核601-2
处理器核601
缓存失效次数生成模块602
任务迁移模块603

图6