

内存迁移方法、装置及计算设备

本申请要求 2019 年 09 月 25 日递交的申请号为 201910913238.6、发明名称为“内存迁移方法、装置及计算设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5

技术领域

本申请实施例涉及计算机应用技术领域，尤其涉及一种内存迁移方法、装置及计算设备。

10 背景技术

虚拟机 (Virtual Machine) 热迁移是指将正在运行的虚拟机从一台物理机器中迁移至另一台物理机器上，迁移过程中虚拟机还可以对外正常提供服务。虚拟机热迁移主要包括 CPU、内存以及 IO 的热迁移，CPU 以及 IO 的热迁移比较简单，传递状态即可，而内存迁移是其中最耗费时间的。

15 目前进行内存迁移时，会首先创建迁移线程，由迁移线程采用迭代方式拷贝脏页至目的端。随着迭代轮数增加，脏页会越来越少，当脏页数量收敛至某个阈值时，即可以挂起源端的虚拟机，将剩余脏页及 CPU 状态、IO 状态拷贝至目的端的虚拟机，拷贝完成之后，目的端恢复虚拟机运行，热迁移即结束。

其中，迁移线程是在将前一轮脏页全部发送至目的端之后，才会执行下一轮的迭代；
20 而迁移线程在每一次迭代过程中，需要执行多个环节的处理操作，若其中任一环节的处理操作发生瓶颈，就会影响内存迁移效率。

发明内容

本申请实施例提供一种内存迁移方法、装置及计算设备，用以提高内存迁移效率。

25 第一方面，本申请实施例中提供了一种内存迁移方法，包括：

创建处理线程、发送线程及针对内存区域的遍历线程；

由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，将脏页地址分配至所述处理线程；

由所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏
30 页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

由所述发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

第二方面，本申请实施例中提供了一种内存迁移方法，包括：

5 创建发送线程及针对内存区域的遍历线程；

由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，基于脏页地址读取脏页数据；为所述脏页数据添加轮数标记信息，生成待迁移脏页；

由所述发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；

10 其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自脏页数据还原至目的内存。

第三方面，本申请实施例中提供了一种内存迁移方法，包括：

创建处理线程、发送线程、遍历线程及针对内存区域的同步线程；

15 由所述同步线程迭代执行同步操作；所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至相应遍历线程；

由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历分配的脏页位图，将脏页地址分配至所述处理线程；

由所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

20 由所述发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

第四方面，本申请实施例中提供了一种内存迁移方法，包括：

接收源端传输的至少一个待迁移脏页；

25 解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数；

按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存。

第五方面，本申请实施例中提供了一种内存迁移装置，包括：

30 第一创建模块，用于创建处理线程、发送线程及针对内存区域的遍历线程；

第一执行模块，用于控制所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，将脏页地址分配至所述处理线程；

第二执行模块，用于控制所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

5 第三执行模块，用于控制所述送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

第六方面，本申请实施例中提供了一种内存迁移装置，包括：

第二创建模块，用于创建发送线程及针对内存区域的遍历线程；

10 第四执行模块，用于控制所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，基于脏页地址读取脏页数据；为所述脏页数据添加轮数标记信息，生成待迁移脏页；

第五执行模块，用于控制所述发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自脏页数据还原至目的内存。

15

第七方面，本申请实施例中提供了一种内存迁移装置，包括：

第三创建模块，用于创建处理线程、发送线程、遍历线程及针对内存区域的同步线程；

第六执行模块，用于控制所述同步线程迭代执行同步操作；所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至相应遍历线程；

20

第七执行模块，用于控制所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历分配的任一脏页位图，将脏页地分配至相应处理线程；

第八执行模块，用于控制所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

25 第九执行模块，用于控制所述发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

第八方面，本申请实施例中提供了一种内存迁移装置，包括：

解析模块，用于接收源端传输的至少一个待迁移脏页，并解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数；

30

还原模块，用于按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存。

第九方面，本申请实施例中提供一种计算设备，包括一个或多个处理组件以及一个或多个存储组件；所述一个或多个处理组件运行源虚拟机；

5 所述一个或多个存储组件存储一个或多个计算机指令；所述一个或多个计算机指令用以被所述一个或多个处理组件调用并执行；

所述一个或多个处理组件用于：

创建处理线程、发送线程及针对内存区域的遍历线程；

10 由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，将脏页地址分配至所述处理线程；

由所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

15 由所述至少一个发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

第十方面，本申请实施例中提供一种计算设备，包括一个或多个处理组件以及一个或多个存储组件；所述一个或多个处理组件运行源虚拟机；

所述一个或多个存储组件存储一个或多个计算机指令；所述一个或多个计算机指令用以被所述一个或多个处理组件调用并执行；

20 所述一个或多个处理组件用于：

创建发送线程及针对内存区域的遍历线程；由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，基于脏页地址读取脏页数据；为所述脏页数据添加轮数标记信息，生成待迁移脏页；

25 由所述至少一个发送线程分别迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自脏页数据还原至目的内存。

第十一方面，本申请实施例中提供一种计算设备，包括一个或多个处理组件以及一个或多个存储组件；所述一个或多个处理组件运行源虚拟机；

30 所述一个或多个存储组件存储一个或多个计算机指令；所述一个或多个计算机指令用以被所述一个或多个处理组件调用并执行；

所述一个或多个处理组件用于：

创建处理线程、发送线程、遍历线程及针对内存区域的同步线程；由所述同步线程迭代执行同步操作；所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至所述遍历线程；

- 5 由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历分配的脏页位图，将脏页地址分配至处理线程；

由所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

- 10 由所述发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

第十二方面，本申请实施例中提供一种计算设备，包括一个或多个处理组件以及一个或多个存储组件；所述一个或多个处理组件运行源虚拟机；

- 15 所述一个或多个存储组件存储一个或多个计算机指令；所述一个或多个计算机指令用以被所述一个或多个处理组件调用并执行；

所述一个或多个处理组件用于：

接收源端传输的至少一个待迁移脏页；

解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数；

- 20 按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存。

第十三方面，本申请实施例中提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时实现如上述第一方面所述的内存迁移方法。

- 25 第十四方面，本申请实施例中提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时实现如上述第二方面所述的内存迁移方法。

第十五方面，本申请实施例中提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时实现如上述第三方面所述的内存迁移方法。

第十六方面，本申请实施例中提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时实现如上述第四方面所述的内存迁移方法。

- 30 本申请实施例中，创建处理线程、发送线程以及针对内存区域的遍历线程，由不同

线程独立迭代执行各自操作，并在脏页中添加轮数标记信息，从而即便不同轮数的脏页未有序到达目的端，目的端也可以根据脏页中的轮数标记信息，将脏页数据依序还原，保证了内存一致性。通过创建多种线程，发送端可以无需考虑保序问题，在同一时间点不同线程可以并行执行不同处理操作，不同线程无需同步和等待，不会因为其中某个处理操作缓慢而阻塞整体迁移流程，从而提高了内存迁移效率和成功率。

本申请的这些方面或其他方面在以下实施例的描述中会更加简明易懂。

附图说明

为了更清楚地说明本申请实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍，显而易见地，下面描述中的附图是本申请的一些实施例，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。

- 图 1 示出了本申请提供的一种内存迁移方法一个实施例的流程图；
- 图 2 示出了本申请提供的一种内存迁移方法又一个实施例的流程图；
- 图 3 示出了本申请提供的一种内存迁移方法又一个实施例的流程图；
- 图 4 示出了本申请实施例的在一个实际应用中的内存迁移交互示意图；
- 图 5 示出了本申请实施例在一个可能实现情况中的内存迁移时间分配示意图；
- 图 6 示出了本申请实施例的在另一个实际应用中的内存迁移交互示意图；
- 图 7 示出了本申请实施例在又一个可能实现情况中的内存迁移时间分配示意图；
- 图 8 示出了本申请提供的一种内存迁移装置一个实施例的结构示意图；
- 图 9 示出了本申请提供的一种计算设备一个实施例的结构示意图；
- 图 10 示出了本申请提供的一种内存迁移装置又一个实施例的结构示意图；
- 图 11 示出了本申请提供的一种计算设备又一个实施例的结构示意图；
- 图 12 示出了本申请提供的一种内存迁移装置又一个实施例的结构示意图；
- 图 13 示出了本申请提供的一种计算设备又一个实施例的结构示意图。

具体实施方式

为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述。

在本申请的说明书和权利要求书及上述附图中的描述的一些流程中，包含了按照特

定顺序出现的多个操作，但是应该清楚了解，这些操作可以不按照其在本文中出现的顺序来执行或并行执行，操作的序号如 101、102 等，仅仅是用于区分开各个不同的操作，序号本身不代表任何的执行顺序。另外，这些流程可以包括更多或更少的操作，并且这些操作可以按顺序执行或并行执行。需要说明的是，本文中的“第一”、“第二”等描述，是用于区分不同的消息、设备、模块等，不代表先后顺序，也不限定“第一”和“第二”是不同的类型。

本申请实施例应用于虚拟机热迁移场景中，虚拟机热迁移对总迁移时间要求较高，希望总迁移时间尽可能短，而虚拟机热迁移过程中的内存迁移是最耗费时间的，因此如何优化内存迁移过程，提高内存迁移效率是本发明主要解决的技术问题。

10 为了方便理解，下面首先对本申请实施例中可能出现的技术术语进行相应解释：

虚拟机（Virtual Machine）：指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。

虚拟机热迁移：是指将正在运行的虚拟机从一台物理机器中迁移至另一台物理机器上，迁移过程中虚拟机还可以对外正常提供服务。虚拟机热迁移主要包括 CPU、内存以及 IO 的热迁移。

内存迭代拷贝：虚拟机热迁移过程中的一种内存迁移方式，由迁移线程第一轮拷贝所有内存页至目的端端，之后每一轮将拷贝前一轮的脏页至目的端，直至剩余脏页数量收敛至某个阈值，挂起源端虚拟机，将剩余脏页全部拷贝至目的端虚拟机；

脏页：在内存迭代拷贝过程中，前一轮产生的数据被修改的内存页。

20 脏页位图（dirty bitmap）：用于记录哪些内存页为脏页，脏页位图中每一位对应一个内存页，若内存页为脏页，标记为 1，否则标记为 0。

脏页标记：位于脏页头部的标识信息，用于标记脏页地址、数据长度等。

目前内存迁移是由迁移线程执行，完成一轮迭代之后进行下一轮迭代，每一轮迭代过程中需要执行遍历脏页位图、获取脏页数据、传输脏页数据等多个处理操作，任一个处理操作若发生问题，将会造成木桶效应，影响整体迁移效率。虽然，实际应用中为了降低数据传输量，提高数据传输效率，以此提高内存迁移效率，可以创建压缩线程对脏页数据进行压缩之后再传输，然后迁移线程需要等待压缩线程压缩完成之后，才能传输。压缩线程出现问题，也将影响整体迁移效率。且在实际业务运行过程中，往往存在超大规格的内存，迁移过程中会产生大量脏页，若内存迁移效率太低，将导致内存迭

代无法收敛，最终导致无法实现虚拟机热迁移。且如果压缩线程已经将当前一轮数据压缩完成，交由迁移线程进行传输，在迁移线程未传输完成的情况下，迁移线程也无法执行下一轮的迭代，导致压缩线程一直处于等待状态，得不到充分利用，从而导致资源浪费。

5 而发明人在研究中出现，内存迁移过程中的各轮迭代需要按序执行，因此迁移线程才需要执行完一轮之后才能进行下一轮的迭代操作，否则，在目的端会出现较早轮的脏页延迟到达目的端，从而覆盖了新一轮的脏页，导致源端虚拟机和目的端虚拟机出现内存不一致的问题，因此按照现有内存迁移方式，源端需要保证本轮脏页全部发出去，才能进行下一轮迭代。

10 为了提高内存迁移效率，且保证源端与目的端的内存一致性，发明人经过一系列研究提出了本申请的技术方案，在本申请实施例中，创建处理线程、发送线程以及针对内存区域的遍历线程，由不同线程独立迭代执行各自操作，并在脏页中添加标识迭代轮数的轮数标记信息，从而即便不同轮数的脏页未有序到达目的端，目的端也可以根据脏页中的轮数标记信息，将脏页数据依序还原，保证了内存一致性。通过设置标识迭代轮数的轮数标记信息以及创建多种线程，发送端可以无需考虑保序问题，在同一时间点不同线程可以并行执行不同处理操作，不同线程无需同步和等待，不会因为其中某个处理操作缓慢而阻塞整体迁移流程，去除了“木桶效应”，从而提高了内存迁移效率和成功率。且不同线程迭代执行各自操作，可以满足负荷运行，提高了资源利用率。

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

图 1 为本申请实施例提供的一种内存迁移方法一个实施例的流程图，该方法可以包括以下几个步骤：

25 101：创建处理线程、发送线程及针对内存区域的遍历线程。在一个可能实现方式中，源内存即作为该内存区域。

源内存即是指源端的虚拟机对应内存。源端可以是指提供源端虚拟机的宿主机，提供源端虚拟机运行环境的计算设备。

在又一个可能实现方式中，对于超大规格的源内存，源内存可以划分多个内存区

域，且不同内存区域之间互不重叠。从而针对每一个内存区域均可以按照本实施例的技术方案进行内容迁移。

其中，源内存若划分为多个内存区域，不同内存区域可以共用至少一个处理线程以及至少一个发送线程，当然，也可以针对每一个内存区域创建各自独立的一组处理线程以及一组发送线程。

因此，所述创建处理线程、发送线程及针对任一内存区域的遍历线程可以包括：
创建针对所述内存区域的遍历线程、发送线程、以及处理线程。

在一个可能实现方式中，可以为一个内存区域创建一个处理线程以及一个发送线程，或者为每个内存区域对应创建一个处理线程以及一个发送线程。

10 在又一个可能实现方式中，为了进一步提高内存迁移效率，可以为一个内存区域创建多个处理线程以及多个发送线程，也即一组处理线程以及一组发送线程；或者为每个内存区域对应创建一组处理线程（包括多个处理线程）以及一组发送线程（包括多个发送线程）。

102：由所述遍历线程迭代执行遍历操作。

15 其中，所述遍历操作包括遍历所述内存区域的脏页位图，将脏页地分配至所述处理线程。

可选地，遍历操作可以具体包括：

遍历所述内存区域的脏页位图，确定脏页地址以及迭代轮数；
将脏页地址以及迭代轮数分配至所述处理线程。

20 由于遍历脏页位图可能获得多个脏页地址，如果存在多个处理线程，可以将多个脏页地址分发给多个处理线程。

其中，遍历线程每一次迭代均执行该遍历操作，每一轮迭代的遍历操作执行完成，也即将所述脏页地址分配至处理线程之后，即可以继续执行下一轮迭代，而无需与其它线程同步或者等待其它线程。其中，脏页地址可以包括块地址及内存地址，内存地址可以是指根据脏页位图换算获得的第多少页为脏页，块地址是指该脏页位于哪一个内存块等。

虚拟机是以内存页形式管理内存，该脏页位图中的每一位对应一个内存页，并按照内存页的存储位置设置，因此，根据脏页位图中的每一位可以确定对应内存页地址。而在脏页位图中可以标记脏页，比如某一位数值为 1，则表明对应内存页为脏页。

由遍历线程负责进行脏页位图遍历，且一个内存区域仅对应一个遍历线程，因此遍历线程也可以获知当前迭代轮数。

其中，脏页位图由虚拟机监视器控制，可选地，遍历线程可以从虚拟机监视器中同步该脏页位图，因此，该遍历操作中还可以包括同步脏页位图，同步脏页位图之后再
5 执行遍历所述脏页位图以及后续操作。

103: 由所述处理线程迭代执行处理操作。

其中，所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页。其中，所述论述标记信息用于标识其对应迭代轮数，迭代轮数可以由遍历线程确定并告知处理线程。

10 其中，创建多个处理线程时，多个处理线程并行运行，并分别迭代执行所述处理操作；每一个处理线程均迭代执行该处理操作。遍历线程可以负责向多个处理线程分别分配待处理任务，每一个处理线程分配获得的处理任务中即包括待处理的脏页地址以及对应的迭代轮数。因此，每一个处理线程即可以执行该处理操作，即基于分配的脏页地址读取脏页数据，并为所述脏页数据添加标识其对应迭代轮数的轮数标记信息，生成待迁
15 移脏页。

可选地，遍历线程可以是向存在空闲资源的一个或多个处理线程分别分配各自对应的待处理任务。每个待处理任务中可以包括至少一个脏页地址，可以结合每个处理线程剩余资源数量来确定每个处理线程分配多少待处理的脏页地址等。

每个处理线程执行处理操作时，遍历线程会一直迭代执行遍历操作，因此，同一个
20 处理线程可能会分配到不同迭代轮数的待处理的脏页地址。

其中，待迁移脏页是由脏页数据以及脏页标记构成，从脏页地址对应的内存页中读取脏页数据，在头部添加脏页标记即形成待迁移的脏页，目的端基于脏页标记中的脏页地址以及数据长度等信息，可以将脏页数据进行还原。因此，可选地，可以是在脏页标记中添加标识其对应迭代轮数的轮数标记信息，脏页标记与脏页数据即构成待迁移脏
25 页。

本申请实施例中，在脏页标记中增加了标识迭代轮数的轮数标记信息，使得可以根据该轮数标记信息确定迭代轮数。

作为一种可选方式，该轮数标记信息可以直接为表示迭代轮数的阿拉伯数字，迭代路数从数字 1 开始计数。

作为另一种可选方式，该轮数标记信息也可以为时间戳信息，该时间戳信息可以是指脏页位图同步时间或者遍历时间等，由于同步或者遍历需要执行完成一轮迭代之后才能执行下一轮，因此通过时间戳先后顺序也可以确定迭代轮数。

当然，还可以采用其它方式来标识迭代轮数，本申请不对此进行具体限制。

5 104：由所述发送线程迭代执行传输操作。

其中，所述传输操作包括发送至少一个待迁移脏页至目的端。每一个发送线程均可以发送至少一个待迁移脏页至目的端。

其中，每个处理线程可以将生成的待迁移脏页分配给发送线程。

10 可选地，创建多个发送线程时，可以结合每个发送线程的空闲资源为每个发送线程分配待迁移脏页。多个发送线程并行运行，并分别迭代执行所述传输操作。

此外，每个处理线程也可以将生成的待迁移脏页缓存至每个处理线程对应的脏页缓存队列中，从而每个发送线程可以负责发送各自对应的脏页缓存队列中的待迁移脏页，每一个脏页缓存队列可以对应至少一个处理线程以及至少一个发送线程等。

15 其中，所述目的端基于不同待迁移脏页的迭代轮数即可以依序将各自的脏页数据还原至目的内存。

目的端可以接收到不同发送线程发送的待迁移脏页，而每个发送线程可能发送不同迭代轮数的待迁移脏页。目的端只需对待迁移脏页中进行解析获知迭代轮数，即可以按照迭代轮数对应的先后顺序，依序将脏页数据还原至目的内存的。

20 本实施例中，通过为脏页数据添加标记迭代轮数的轮数标记信息，且内存迭代过程中的各个操作采用独立线程执行，使得内存迁移过程中各个操作可以以流水式形式并发执行，由于脏页中存在标记迭代轮数的轮数标记信息，各个独立线程之间不需要同步与等待，不会因为某个操作缓存而阻塞整体迁移流程，去除了“木桶效应”。且可以保证各个线程满负荷运行，大幅提升了内存迁移效率和迁移成功率。

25 此外，为了提高数据传输效率，降低数据传输量，处理线程执行的传输操作还可以包括对脏页数据进行压缩，也即处理线程可以执行压缩操作。

因此，在某些实施例中，所述处理线程具体迭代执行如下处理操作：

基于分配的脏页地址读取脏页数据；

将所述脏页数据进行压缩；

为压缩之后的所述脏页数据添加标识其对应迭代轮数的轮数标记信息，生成待迁移脏页。

每个处理线程均迭代执行上述处理操作。

也即待迁移脏页中存储的为压缩之后的脏页数据，以可以降低数据传输量。目的端
5 可以通过解压操作获得脏页数据之后，再进行还原。

结合上文描述可知，可以由遍历线程从虚拟机监视器同步脏页位图，而为了进一步提高内存迁移效率，在某些实施例中，所述方法还可以包括：

创建针对所述内存区域的同步线程；

10 由所述同步线程迭代执行同步操作。

其中，所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至所述遍历线程。此外还可以将迭代轮数发送至遍历线程。

也即可以由独立线程迭代执行同步操作，从而可以使得同步操作和遍历操作也可以并行执行。

15

在某些实施例中，创建多个处理线程时，所述遍历线程具体迭代执行如下遍历操作：

遍历所述内存区域的脏页位图，确定脏页地址及迭代轮数；

20 基于脏页处理数量，向存在空闲资源的一个或多个处理线程分别分配各自对应的待处理任务；其中，所述待处理任务包括所述迭代轮数以及待处理的至少一个脏页地址。

由遍历线程负责分配待处理任务时，可以结合当前一轮的脏页处理数量进行，考虑负载均衡，可以向存在空闲资源的一个或多个处理线程分别分配各自对应的待处理任务，因此，每个处理线程获得的待处理的脏页地址可能不同。

25 其中，脏页处理数量通过遍历脏页位图确定，例如可以是指脏页位图中数值为 1 的数量。

在某些实施例中，处理线程具体迭代执行如下处理操作：

基于分配的脏页地址读取脏页数据；

为所述脏页数据添加标识其对应迭代轮数的轮数标记信息，生成待迁移脏页；

将所述待迁移脏页缓存至所述处理线程对应的脏页缓存队列中。

因此，发送线程以具体迭代执行如下发送操作：

将所述脏页缓存队列中的至少一个待迁移脏页传输至目的端。

若创建多个处理线程以及多个发送线程时，每个处理线程可以对应脏页缓存队列，每个发送线程可以负责一个或多个脏页缓存队列，每个发送线程可以将其对应脏页缓存队列中的至少一个待迁移脏页传输至目的端。

每个发送线程只要发现脏页缓存队列中缓存待迁移脏页，即可以将该脏页缓存队列中至少一个待迁移脏页发送至目的端。每个发送线程发送的待迁移脏页数量可以由每个发送线程的发送性能决定等。

10

在实际应用中，由于虚拟机运行在物理机器中，而一个物理机器中可以包括一个或多个处理组件，一个或多个处理组件例如可以包括 CPU、FPGA、FAT 卡等硬件，可以由 CPU 执行所有线程，当然也可以由多个处理组件分别执行。因此，在某些实施例中，所述创建至少一个发送线程、至少一个处理线程及针对任一内存区域的遍历线程可以包

15

括：

在一个或多个处理组件中创建发送线程、处理线程及针对内存区域的遍历线程。

其中，在多个处理组件中创建发送线程、个处理线程及针对内存区域的遍历线程的情况下，每个处理组件可以仅创建一种类型的线程，当然也可以创建一种或多种类型的线程。

20

在实际应用中，如果不需要进行数据压缩，获取脏页数据以及为脏页数据添加脏页标记，效率会很快，不会存在瓶颈问题，因此，在这种情况下，也可以不需要创建独立的处理线程负责处理操作，可以由遍历线程执行。因此，作为又一个实施例，如图 2 所示，为本申请实施例提供的一种内存迁移方法又一个实施例的流程图，该方法可以包括

25

以下几个步骤：

201：创建至发送线程及针对内存区域的遍历线程。

在一个可能实现方式中，源内存即可以作为该内存区域。

在又一个可能实现方式中，对于超大规格的源内存，源内存可以划分为多个内存区域，且不同内存区域之间互不重叠。从而针对每一个内存区域均可以按照本实施例的技

术方案进行内存迁移。

其中，源内存若划分为多个内存区域，不同内存区域可以共用发送线程，当然，也可以针对每一个内存区域创建各自独立的发送线程。因此，所述创建至发送线程及针对内存区域的遍历线程可以包括：

- 5 创建针对所述内存区域的遍历线程、以及发送线程。

其中，具体可以为每个内存区域创建多个发送线程。

202：由所述遍历线程迭代执行遍历操作；

其中，所述遍历操作包括遍历所述内存区域的脏页位图，基于所述脏页地址读取脏页数据；为所述脏页数据添加轮数标记信息，生成待迁移脏页。

- 10 遍历脏页位图可以确定脏页地址以及迭代轮数，从而可以基于脏页地址读取脏页数据，并为脏页数据添加标识所述迭代轮数的轮数标记信息，以生成待迁移脏页。

可选地，遍历线程可以是虚拟机监视器中同步该脏页位图，因此，该遍历操作中还可以包括同步脏页位图，同步脏页位图之后再执行遍历所述内存区域的脏页位图及后续操作。

- 15 203：由所述发送线程迭代执行传输操作。

其中，所述传输操作包括发送所述待迁移脏页至目的端。可选地，可以结合发送性能，发送一个或多个待迁移脏页至目的端。

其中，所述目的端用于基于不同待迁移脏页的迭代轮数依序将各自脏页数据还原至目的内存。

- 20 创建多个发送线程时，多个发送线程可以并行运行并分别迭代执行所述传输操作。

遍历线程可以将生成的多个待迁移脏页分配给多个发送线程，可选地，可以结合每个发送线程的空闲资源进行分配，每个发送线程的分配的脏页数量可以结合每个发送线程的发送性能决定。

- 25 此外，遍历线程可以将生成的待迁移脏页缓存至存在空闲资源的脏页缓存队列中，从而发送线程可以发送脏页缓存队列中的待迁移脏页。

其中，可以设置多个脏页缓存队列，当创建多个发送线程时，每一个脏页缓存队列可以对应至少一个发送线程等。从而每个发送线程可以负责发送各自对应的脏页缓存队列中的待迁移脏页，每一个脏页缓存队列可以对应至少一个发送线程等。

本实施例中，通过为脏页数据添加标记迭代轮数的轮数标记信息，且内存迭代过程

中的各个操作采用独立线程执行，使得内存迁移过程中各个操作可以以流水式形式并发执行，由于脏页中存在标记迭代轮数的轮数标记信息，各个独立线程之间不需要同步与等待，不会因为某个操作缓存而阻塞整体迁移流程，去除了“木桶效应”。且可以保证各个线程满负荷运行，大幅提升了内存迁移效率和迁移成功率。

5 在某些实施例中，所述方法还可以包括：

创建针对所述内存区域的同步线程；

由所述同步线程迭代执行同步操作；所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至所述遍历线程。此外，还可以将迭代轮数发送至遍历线程。

10 在某些实施例中，发送线程包括多个时，所述遍历线程具体迭代执行如下遍历操作：

遍历所述内存区域的脏页位图，确定脏页地址及迭代轮数；

基于所述脏页地址读取脏页数据；

为所述脏页数据添加标识所述迭代轮数的轮数标记信息，生成待迁移脏页；

15 基于脏页处理数量，向存在空闲资源的一个或多个发送线程分别分配至少一个待迁移脏页。

在某些实施例中，源内容划分为多个内存区域时，所述创建发送线程及针对内存区域的遍历线程包括：

20 创建针对所述内存区域的遍历线程以及发送线程。也即为每个内存区域均创建各自对应的遍历线程以及发送线程。

在某些实施例中，所述创建发送线程及针对内存区域的遍历线程可以包括：

在一个或多个处理组件中创建发送线程及针对内存区域的遍历线程。

此外，本申请实施例还提供了一种内存迁移方法，可以包括：

25 创建处理线程、发送线程、遍历线程及针对内存区域的同步线程；其中，源内存可以划分为至少一个内存区域；不同内存区域互不重叠；

由所述同步线程迭代执行同步操作；所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至相应遍历线程；

由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历分配的脏页位图，将所

述脏页地址分配至处理线程；

由所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加标识其对应迭代轮数的轮数标记信息，生成待迁移脏页；

- 5 由所述发送线程迭代执行传输操作；所述传输操作包括发送待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页中的迭代轮数依序将各自的脏页数据还原至目的内存。

也即为内存区域创建同步线程的情况下，遍历线程可以创建多个以实现并发执行，进一步提高内存迁移效率。

- 10 当然，在一种实现情况中，由于遍历操作不会花费太长时间，资源消耗少，可以快速完成，因此，可以针对每一个内存区域创建一个遍历线程，具体实现方式可以参见上文中所述，当然也可以只创建一个遍历线程，多个内存区域共用一个遍历线程。

其中，可以具体创建多个处理线程、多个发送线程以及多个遍历线程；

- 多个遍历线程并行运行，并分别迭代执行所述遍历操作；多个处理线程并行运行，
15 并分别迭代执行所述处理操作；多个发送线程并行运行，并分别迭代执行所述发送操作。

- 图 3 为本申请实施例提供的一种内存迁移方法又一个实施例的流程图，本实施例从目的端角度对本申请技术方案进行描述，目的端可以是指目的端虚拟机的宿主机，提供
20 目的端虚拟机运行环境的计算设备。

该方法可以包括以下几个步骤：

301：接收源端传输的至少一个待迁移脏页。

可选地，可以是接收源端的发送线程传输的至少一个待迁移脏页。若源端存在多个发送线程时，可以分别接收每个发送线程传输的待迁移脏页。

- 25 其中，源端的待迁移脏页的获得以及发送可以参见上文各个实施例中所述，在此将不再重复赘述。

其中，一种可能的实现方式，待迁移脏页可以由脏页标记以及脏页数据构成，其中，脏页标记中包括标识脏页数据对应的迭代轮数的标记信息。

另一种可能的实现方式，待迁移脏页可以由脏页标记以及压缩之后的脏页数据构

成，其中，脏页标记中包括标识脏页数据对应的迭代轮数的轮数标记信息。

此外，脏页标记中还可以包括原有脏页信息，例如脏页地址、数据长度等等。

302：解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数。

- 5 303：按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存。

通过解析待迁移脏页，即可以获得脏页数据，以及基于轮数标记信息确定的该脏页数据对应的迭代轮数。

- 10 从而即可以按照迭代轮数表示的先后顺序，依序将各自对应的脏页数据还原至目的内存。

- 本实施例中，通过为脏页数据添加标记迭代轮数的轮数标记信息，目的端接收到待迁移脏页之后，根据该轮数标记信息即可以确定迭代轮数，从而可以依序将待迁移脏页中的脏页数据还原至目的内存，保证了内存一致性。且源端在内存迭代过程中的各个操作采用独立线程执行，使得内存迁移过程中各个操作可以以流水式形式并发执行，不会
15 因为某个操作缓存而阻塞整体迁移流程，大幅提升了内存迁移效率和迁移成功率。

其中，目的端也可以创建一个或多个线程执行上述步骤 301~步骤 303 的操作。作为一种可选方式，所述方法还可以包括：

创建接收线程及还原线程；

- 20 其中，由所述接收线程分别迭代执行所述接收源端传输的至少一个待迁移脏页，及解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数的接收操作；

其中，由所述还原线程迭代执行所述按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存的还原操作。

- 25 其中，可以创建多个接收线程以及多个还原线程。多个接收线程可以并行运行，并分别迭代执行所述接收操作；多个还原线程可以并行运行，并分别迭代执行所述还原操作。

为了保证还原准确性，多个还原线程需同时针对同一迭的待迁移脏页进行脏页数据的还原，同一轮的脏页数据还原完成之后，再由多个还原线程执行下一轮的脏页数据的

还原。也即同一轮的多个待迁移脏页可以分发给多个还原线程进行脏页数据的还原，保证在同一时间不同还原线程是对同一轮的待迁移脏页进行还原即可。

而多个接收线程可以分别针对不同轮的待迁移脏页进行解析，无需关注待迁移脏页的迭代轮数。

5

在某些实施例中，所述按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存可以包括：

将所述至少一个待迁移脏页缓存至各自迭代轮数对应的接收缓存队列中；

按照迭代轮数依序将不同接收缓存队列中的待迁移脏页中的脏页数据还原至目的内存。

10

因此，上述接收线程可以迭代执行所述接收源端传输的至少一个待迁移脏页，及解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数，及将所述至少一个待迁移脏页放入各自迭代轮数对应的接收缓存队列中的接收操作；

所述还原线程迭代执行所述按照迭代轮数依序将不同接收缓存队列中的待迁移脏页中的脏页数据还原至目的内存的还原操作。

15

其中，若待迁移脏页中的脏页数据进行了压缩，可选地，所述按照迭代轮数依序将不同接收缓存队列中的待迁移脏页的脏页数据还原至目的内存可以包括：

按照迭代轮数依序将不同接收缓存队列中待迁移脏页中的脏页数据解压并还原至目的内存。

20

为了便于理解，下面从源内存分别划分为一个内存区域及多个内存区域（在下文中多以两个内存区域为例进行说明）的角度，对本申请实施例的技术方案进行描述。

其中，对于内存规格不大的源内存，迭代拷贝过程中，脏页数量不会大批量增加，因此，源内存可以作为一个内存区域进行内存迁移，不会影响整体迁移效率。

25

假设需要进行数据压缩操作，因此源端可以创建遍历线程、一组压缩线程以及一组发送线程，此外还可以创建同步线程。

其中，一组压缩线程中可以包括一个或多个压缩线程，一组发送线程中可以包括一个或多个发送线程。

参见图 4 所示的内存迁移交互示意图中，

对于源端的源内存 40，同步线程 401 负责迭代执行同步操作；同步操作包括从虚拟机监视器 41 中同步所述源内存的脏页位图，并将脏页位图以及对应的迭代轮数发送至遍历线程 402。其中，同步线程 401 每一次同步操作即可以累计一次迭代轮数。

5 遍历线程 402 负责迭代执行遍历操作；遍历操作包括遍历脏页位图，确定脏页地址及迭代轮数，并将所述脏页地址及所述迭代轮数分配至相应压缩线程；

其中，压缩线程 403 可以包括多个，每一个压缩线程 403 负责迭代执行压缩操作；压缩操作可以包括基于分配的任一脏页地址读取脏页数据；将所述脏页数据进行压缩；为压缩之后的所述脏页数据添加脏页标记，生成待迁移脏页；并可以将待迁移脏页缓存至相应的脏页缓存队列中。其中，脏页标记中包括标识迭代轮数的轮数标记信息。

10 如图 4 中可知，遍历线程 402 可以将同一轮中的多个脏页地址分发给多个压缩线程 403 处理，可以结合剩余资源确定每个压缩线程的处理数量。

每个压缩线程 403 压缩完成之后，即将待迁移脏页缓存至各自对应的脏页缓存队列 42 中，由于每个压缩线程 403 可以处于不同轮的待迁移脏页，脏页缓存队列中也可以包括不同迭代轮数对应的待迁移脏页。

之后，每个发送线程 404 可以负责迭代执行发送操作，也即将对应脏页缓存队列 42 中的至少一个待迁移脏页发送至目的端。

对于目的端，可以创建接收线程 405 以及还原线程 406 等，其中，接收线程 405 以及还原线程 406 可以包括多个。

20 每个接收线程 405 负责迭代执行接收操作；其中，接收操作可以包括接收源端发送的至少一个待迁移脏页，解析每个待迁移脏页获得轮数标记信息以及脏页数据；将脏页数据缓存至轮数标记信息对应的接收缓存队列 43 中。

从而同一轮的脏页数据缓存同一接收缓存队列 43 中，同一接收缓存队列 43 中只缓存同一轮的脏页数据。

25 还原线程 406 可以负责迭代执行还原操作；其中，还原操作可以包括：按照迭代轮数依序从接收缓存队列 43 中将脏页数据还原至目的端的目的内存 44。

其中，若存在多个还原线程，针对同一个接收缓存队列 43 中的脏页数据，可以分发给多个还原线程进行还原。多个还原线程将一个接收缓存队列中的脏页数据全部还原完成之后，再对下一个接收缓存队列中的脏页数据进行还原。多个还原线程可以分别还

原同一个接收缓存队列中的至少部分的脏页数据。

由于不同种类的线程可以在同一时间点并行执行内存迭代过程中的不同操作，使得内存迁移过程中各个操作可以以流水式形式并发执行，大幅提升了内存迁移效率和迁移成功率。

5 如图 5 所示的内存迭代过程中的时间分配示意图中，假设每一种线程执行各自操作所花费时间相同，由图 5 中可知，例如在 t3 时间段，发送线程执行第 1 轮的发送操作，压缩线程可以并行执行第 2 轮的压缩操作，遍历线程可以并行执行第 3 轮的遍历操作，同步线程可以并行执行第 4 轮的同步操作，从 t0-t7 时间段可以完成五轮迭代操作，而采用现有方式最多只能完成两轮迭代操作。因此，通过本申请实施例的技术方案，各个
10 独立线程之间不需要同步与等待，不会因为某个操作缓存而阻塞整体迁移流程，可以保证各个独立线程满负荷运行，因此大幅提升了内存迁移效率和迁移成功率。且对于发送线程以及处理线程可以创建多个，使得多个处理线程或者多个发送线程也可以并发执行，以进一步提高内存迁移效率。

15 对于超大规格的源内存，为了保证内存迁移效率，可以将源内存划分为多个内存区域，多个内存区域之间互不重叠。每个内存区域均可以按照图 4 所示方案进行内存迭代，假设源内存划分为两个内存区域：区域 1 以及区域 2，如图 6 中示出了的内存迭代交互示意图中，示出了源端针对每个内存区域的内地迭代过程。

 针对每个内存区域均可以创建同步线程 601、遍历线程 602、一组压缩线程（包括至少一个压缩线程 603）以及一组发送线程（包括至少一个发送线程 604）。

 每个同步线程只负责针对其对应内存区域迭代执行同步操作；该同步操作也即是指同步其对应内存区域的脏页位图；

 每个遍历线程只负责针对其对应内存区域迭代执行遍历操作；该遍历操作也即是指遍历其对应内区域的脏页位图，确定脏页地址及迭代轮数，并将所述脏页地址及所述迭代轮数分配至其对应内存区域所对应的相应压缩线程；
25

 而同一内存区域对应的压缩线程，每个压缩线程可以只负责针对其对应内存区域迭代执行压缩操作，也即基于分配的脏页地址读取脏页数据；将所述脏页数据进行压缩；为压缩之后的所述脏页数据添加脏页标记，生成待迁移脏页；并可以将待迁移脏页缓存至相应的脏页缓存队列中。其中，脏页标记中包括标识迭代轮数的轮数标记信息。

而同一内存区域对应的发送线程，每个发送线程可以只负责针对其对应内存区域迭代执行发送操作，也即将其对应脏页缓存队列中的至少一个待迁移脏页发送至目的端。

对于目的端来说，虽然源端将源内存划分为多个内存区域，但是多个内存区域互不重叠，每个脏页的脏页地址并不会发生变化，目的端只需要知道脏页地址即可以进行脏页数据的还原，最终脏页数据会被还原到目的内存的不同内存区域中，不会出现覆盖。因此，目的端执行操作与图 4 所示的目的端执行操作相同，并无需获知每个待迁移脏页所属内存区域，无需分区域还原，具体还原过程可以参见图 4 中所示，在此将不再重复赘述。

其中，内存迭代过程中判断内存中剩余脏页数量是否收敛至某个阈值，可以是判断每个内存区域的剩余脏页数量是否均收敛至某个阈值，若是，确认内存迭代收敛成功，此时再挂起源端的虚拟机，执行后续热迁移操作，后续热迁移操作与现有技术相同，在此将不再赘述。

由于通过划分多个内存区域，多个内存区域可以并发的进行内存迭代迁移，可以将内存迭代效率提升多倍，例如划分两个内存区域，相较于图 4 所示方案，在相同时间单位下可以完成 2 轮迭代，内存迭代效率提升 2 倍。

如图 7 所示的内存迭代过程中的时间分配示意图中，假设每一种线程执行各自操作所花费时间相同，假设源内存划分为两个内存区域，由图 6 中可知，在每个单位时间内，可以并行执行两个相同操作，各个内存区域互相独立互不影响，每个内存区域中的各个独立线程之间也不需要同步与等待，不会因为某个操作缓存而阻塞整体迁移流程，可以保证各个独立线程满负荷运行，因此大幅提升了内存迁移效率和迁移成功率。

图 8 为本申请实施例提供的一种内存迁移装置一个实施例的结构示意图，该装置可以包括：

第一创建模块 801，用于创建处理线程、发送线程及针对内存区域的遍历线程；
第一执行模块 802，用于控制所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，将脏页地址分配至所述处理线程；

第二执行模块 803，用于控制所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

第三执行模块 804，用于控制所述发送线程迭代执行传输操作；所述传输操作包括发送所述待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

在某些实施例中，所述第二执行模块控制所述至少一个处理线程分别迭代执行的处
5 理操作可以包括：

基于分配的脏页地址读取脏页数据；

将所述脏页数据进行压缩；

为压缩之后的所述脏页数据添加标识其对应迭代轮数的轮数标记信息，生成待迁移脏页。

10 在某些实施例中，该第一创建模块还用于创建针对所述内存区域的同步线程；

该装置还可以包括：

同步执行模块，用于控制所述同步线程迭代执行同步操作；所述同步操作包括获取所述任一内存区域的脏页位图，并将所述脏页位图发送至所述遍历线程。

在某些实施例中，所述第一执行模块控制所述遍历线程执行的遍历操作具体包括：

15 遍历所述内存区域的脏页位图，确定脏页地址及迭代轮数；

基于脏页处理数量，向存在空闲资源的一个或多个处理线程分别分配各自对应的待处理任务；其中，所述待处理任务包括所述迭代轮数以及待处理的至少一个脏页地址。

在某些实施例中，所述第二执行模块控制所述处理线程迭代执行的处理操作可以包括：

20 基于分配的脏页地址读取脏页数据；

为所述脏页数据添加标识其对应迭代轮数的轮数标记信息，生成待迁移脏页；

将所述待迁移脏页缓存至所述处理线程对应的脏页缓存队列中。

在某些实施例中，所述第三执行模块控制所述发送线程分别迭代执行的传输操作可以具体包括：

25 将脏页缓存队列中的至少一个待迁移脏页传输至目的端。

在某些实施例中，所述第一创建模块具体用于创建针对所述内存区域的遍历线程、发送线程、以及处理线程。

在某些实施例中，所述第一创建模块具体用于在一个或多个处理组件中创建发送线程、个处理线程及针对内存区域的遍历线程。

其中，可以具体创建多个处理线程、多个发送线程以及多个遍历线程；

多个遍历线程并行运行，并分别迭代执行所述遍历操作；多个处理线程并行运行，并分别迭代执行所述处理操作；多个发送线程并行运行，并分别迭代执行所述发送操作。

5 其中，源内存划分为多个所述内存区域，不同内存区域互不重叠。

图 8 所述的内存迁移装置可以执行图 1 所示实施例所述的内存迁移方法，其实现原理和技术效果不再赘述。对于上述实施例中的内存迁移装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

10 在一个可能的设计中，图 8 所示实施例的内存迁移装置可以实现为计算设备，该计算设备为源虚拟机，也即待迁移虚拟机的宿主机，如图 9 所示，该计算设备可以包括一个或多个存储组件 901 以及一个或多个处理组件 902；

所述一个或多个存储组件 901 存储一条或多条计算机指令，其中，所述一条或多条计算机指令供所述一个或多个处理组件 902 调用执行。

15 所述一个或多个处理组件 902 用于：

创建处理线程、发送线程及针对内存区域的遍历线程；

由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，将脏页地址分配至处理线程；

20 由所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

由所述发送线程迭代执行传输操作；所述传输操作包括发送待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

其中，处理线程、发送线程以及遍历线程可以分别在不同处理组件中创建。

25 其中，该一个或多个处理组件 902 可以包括中央处理器（CPU）、数字信号处理器（DSP）、数字信号处理设备（DSPD）、可编程逻辑器件（PLD）、现场可编程门阵列（FPGA）、控制器、微控制器、微处理器或其他电子元件等。

该一个或多个存储组件 901 被配置为存储各种类型的数据以支持在计算设备的操作。该一个或多个存储组件可以由任何类型的易失性或非易失性存储设备或者它们的组

合实现，如静态随机存取存储器（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），磁存储器，快闪存储器，磁盘或光盘。

当然，计算设备必然还可以包括其他部件，例如输入/输出接口、通信组件等。

- 5 输入/输出接口为处理组件和外围接口模块之间提供接口，上述外围接口模块可以是输出设备、输入设备等。

通信组件被配置为便于计算设备和其他设备之间有线或无线方式的通信等。

本申请实施例还提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时可以实现上述图 1 所示实施例的内存迁移方法。

10

图 10 为本申请实施例提供的一种内存迁移装置又一个实施例的结构示意图，该装置可以包括：

第二创建模块 1001，用于创建发送线程及针对内存区域的遍历线程；其中，源内存可以划分为至少一个内存区域；不同内存区域互不重叠；

- 15 第四执行模块 1002，用于控制所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，基于脏页地址读取脏页数据；为所述脏页数据添加轮数标记信息，生成待迁移脏页；

第五执行模块 1003，用于控制所述发送线程迭代执行传输操作；所述传输操作包括发送待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自脏页数据还原至目的内存。

20

在某些实施例中，所述第二创建模块还可以用于创建针对所述任一内存区域的同步线程；

该装置还可以包括：

同步执行模块，用于控制所述同步线程迭代执行同步操作；所述同步操作包括获取

25

所述任一内存区域的脏页位图，并将所述脏页位图发送至所述遍历线程。

在某些实施例中，所述第四执行模块控制所述遍历线程迭代执行的遍历操作可以包括：

遍历所述内存区域的脏页位图，确定脏页地址及迭代轮数；

基于所述脏页地址读取脏页数据；

为所述脏页数据添加标识所述迭代轮数的轮数标记信息，生成待迁移脏页；

基于脏页处理数量，向存在空闲资源的一个或多个发送线程分别分配至少一个待迁移脏页。

在某些实施例中，所述第二创建模块具体用于创建针对所述内存区域的遍历线程以及发送线程。

在某些实施例中，所述第二创建模块具体用于在一个或多个处理组件中创建发送线程及针对内存区域的遍历线程。

图 10 所述的内存迁移装置可以执行图 2 所示实施例所述的内存迁移方法，其实现原理和技术效果不再赘述。对于上述实施例中的内存迁移装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

在一个可能的设计中，图 10 所示实施例的内存迁移装置可以实现为计算设备，该计算设备为源虚拟机，也即待迁移虚拟机的宿主机，如图 11 所示，该计算设备可以包括一个或多个存储组件 1101 以及一个或多个处理组件 1102；

所述一个或多个存储组件 1101 存储一条或多条计算机指令，其中，所述一条或多条计算机指令供所述一个或多个处理组件 1102 调用执行。

所述一个或多个处理组件 1102 用于：

创建发送线程及针对内存区域的遍历线程；其中，源内存可以划分为至少一个内存区域；不同内存区域互不重叠；

由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历所述内存区域的脏页位图，基于脏页地址读取脏页数据；为所述脏页数据添加轮数标记信息，生成待迁移脏页；

由所述发送线程迭代执行传输操作；所述传输操作包括发送待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自脏页数据还原至目的内存。

其中，发送线程以及遍历线程可以分别在不同处理组件中创建。

其中，该一个或多个处理组件 1102 可以包括中央处理器（CPU）、数字信号处理器（DSP）、数字信号处理设备（DSPD）、可编程逻辑器件（PLD）、现场可编程门阵列（FPGA）、控制器、微控制器、微处理器或其他电子元件等。

该一个或多个存储组件 1101 被配置为存储各种类型的数据以支持在计算设备的操作。该一个或多个存储组件可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），

5 磁存储器，快闪存储器，磁盘或光盘。

当然，计算设备必然还可以包括其他部件，例如输入/输出接口、通信组件等。

输入/输出接口为处理组件和外围接口模块之间提供接口，上述外围接口模块可以是输出设备、输入设备等。

通信组件被配置为便于计算设备和其他设备之间有线或无线方式的通信等。

10 本申请实施例还提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时可以实现上述图 2 所示实施例的内存迁移方法。

此外，本申请实施例还提供了一种内存迁移装置，包括：

第三创建模块，用于创建处理线程、发送线程、遍历线程及针对内存区域的同步线程；其中，源内存可以划分为至少一个内存区域；不同内存区域互不重叠；

15 第六执行模块，用于控制所述同步线程迭代执行同步操作；所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至遍历线程；

第七执行模块，用于控制所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历分配的脏页位图，将所述脏页地址分配至处理线程；

20 第八执行模块，用于控制所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

第九执行模块，用于控制所述至少一个发送线程分别迭代执行传输操作；所述传输操作包括发送待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数
25 依序将各自的脏页数据还原至目的内存。

在一个可能的设计中，该装置可以实现了为计算设备，该计算设备为源虚拟机，也即待迁移虚拟机的宿主机，该计算设备可以包括一个或多个存储组件以及一个或多个处理组件；

所述一个或多个存储组件存储一条或多条计算机指令，其中，所述一条或多条计算

机指令供所述一个或多个处理组件调用执行。

所述一个或多个处理组件用于：

创建处理线程、发送线程、遍历线程及针对内存区域的同步线程；其中，源内存可以划分为至少一个内存区域；不同内存区域互不重叠；

- 5 由所述同步线程迭代执行同步操作；所述同步操作包括获取所述内存区域的脏页位图，并将所述脏页位图发送至遍历线程；

由所述遍历线程迭代执行遍历操作；所述遍历操作包括遍历分配的脏页位图，并将所述脏页地址分配至相应处理线程；

- 10 由所述处理线程迭代执行处理操作；所述处理操作包括基于分配的脏页地址读取脏页数据，并为所述脏页数据添加轮数标记信息，生成待迁移脏页；

由所述发送线程迭代执行传输操作；所述传输操作包括发送待迁移脏页至目的端；其中，所述目的端基于不同待迁移脏页的迭代轮数依序将各自的脏页数据还原至目的内存。

- 15 图 12 为本申请实施例提供的一种内存迁移装置又一个实施例的结构示意图，该装置可以包括：

解析模块 1201，用于接收源端传输的至少一个待迁移脏页，并解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数；

- 20 还原模块 1202，用于按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存。

在某些实施例中，所述还原模块具体用于将所述至少一个待迁移脏页缓存至各自迭代轮数对应的接收缓存队列中；按照迭代轮数依序将不同接收缓存队列中的待迁移脏页中的脏页数据还原至目的内存。

- 25 在某些实施例中，所述还原模块具体用于按照迭代轮数依序将不同接收缓存队列中待迁移脏页中的脏页数据解压并还原至目的内存。

在某些实施例中，还包括：

第四创建模块，用于创建接收线程及还原线程；

所述解析模块具体控制所述接收线程迭代执行所述接收源端传输的至少一个待迁移脏页，及解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏

页对应的迭代轮数，及将所述至少一个待迁移脏页放入各自迭代轮数对应的接收缓存队列中的接收操作；

所述还原模块具体控制所述还原线程迭代执行所述按照迭代轮数依序将不同接收缓存队列中的待迁移脏页中的脏页数据还原至目的内存的还原操作。

- 5 图 12 所述的内存迁移装置可以执行图 3 所示实施例所述的内存迁移方法，其实现原理和技术效果不再赘述。对于上述实施例中的内存迁移装置其中各个模块、单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述，此处将不做详细阐述说明。

- 10 在一个可能的设计中，图 12 所示实施例的内存迁移装置可以实现为计算设备，该计算设备为目的虚拟机的宿主机，如图 13 所示，该计算设备可以包括一个或多个存储组件 1301 以及一个或多个处理组件 1302；

所述一个或多个存储组件 1301 存储一条或多条计算机指令，其中，所述一条或多条计算机指令供所述一个或多个处理组件 1302 调用执行。

所述一个或多个处理组件 1302 用于：

- 15 接收源端传输的至少一个待迁移脏页；

解析所述至少一个待迁移脏页中的轮数标记信息，确定所述至少一个待迁移脏页对应的迭代轮数；

按照所述至少一个待迁移脏页的迭代轮数，将所述至少一个待迁移脏页中的脏页数据依序还原至目的内存。

- 20 其中，该一个或多个处理组件 1302 可以包括中央处理器（CPU）、数字信号处理器（DSP）、数字信号处理设备（DSPD）、可编程逻辑器件（PLD）、现场可编程门阵列（FPGA）、控制器、微控制器、微处理器或其他电子元件等。

- 25 该一个或多个存储组件 1301 被配置为存储各种类型的数据以支持在计算设备的操作。该一个或多个存储组件可以由任何类型的易失性或非易失性存储设备或者它们的组合实现，如静态随机存取存储器（SRAM），电可擦除可编程只读存储器（EEPROM），可擦除可编程只读存储器（EPROM），可编程只读存储器（PROM），只读存储器（ROM），磁存储器，快闪存储器，磁盘或光盘。

当然，计算设备必然还可以包括其他部件，例如输入/输出接口、通信组件等。

输入/输出接口为处理组件和外围接口模块之间提供接口，上述外围接口模块可以

是输出设备、输入设备等。

通信组件被配置为便于计算设备和其他设备之间有线或无线方式的通信等。

本申请实施例还提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被计算机执行时可以实现上述图 3 所示实施例的内存迁移方法。

5 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，上述描述的系统，装置和单元的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

以上所描述的装置实施例仅仅是示意性的，其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性的劳动的情况下，即可以理解并实施。

10

通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到各实施方式可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件。基于这样的理解，上述技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品可以存储在计算机可读存储介质中，如 ROM/RAM、磁碟、光盘等，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）执行各个实施例或者实施例的某些部分所述的方法。

15

最后应说明的是：以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

20