



(12) 发明专利申请

(10) 申请公布号 CN 105094686 A

(43) 申请公布日 2015. 11. 25

(21) 申请号 201410193960. 4

(22) 申请日 2014. 05. 09

(71) 申请人 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

申请人 中国科学院计算技术研究所

(72) 发明人 魏巍 张立新 熊劲 蒋德钧

(74) 专利代理机构 北京同立钧成知识产权代理
有限公司 11205

代理人 刘芳

(51) Int. Cl.

G06F 3/06(2006. 01)

G06F 12/08(2006. 01)

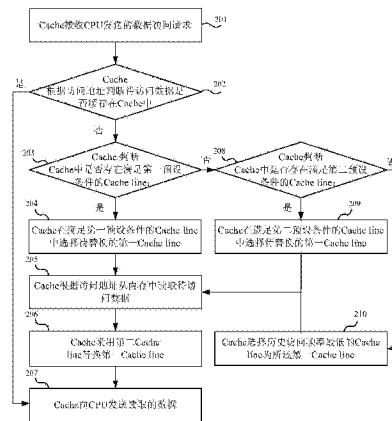
权利要求书3页 说明书10页 附图3页

(54) 发明名称

数据缓存方法、缓存和计算机系统

(57) 摘要

本发明实施例提供一种数据缓存方法、缓存和计算机系统。本发明实施例中,Cache 在访问请求不命中需要确定待替换的 Cache line 时,不仅需要考虑 Cache line 的历史访问频率,还要考虑 Cache line 对应的内存类型,从而可以优先替换与 DRAM 内存类型对应的 Cache line,减少 Cache 对存储于 DRAM 的数据的缓存量,从而使得 Cache 能够提高对存储于 NVM 中的数据的数据的缓存量,使得针对存储于 NVM 中的数据的数据的访问请求尽可能的在 Cache 中能够找到相应的数据,从而减少了从 NVM 中读取数据的情形,减少了从 NVM 中读取数据的延迟,有效的提高访问效率。



1. 一种数据缓存方法,所述方法由缓存 Cache 执行,其特征在于,包括:

接收 CPU 发送的数据访问请求,所述数据访问请求中包含访问地址;

根据所述访问地址判断待访问数据是否缓存在所述 Cache 中;

在确定所述待访问数据没有缓存在所述 Cache 中的情况下,根据所述 Cache 中的缓存线 Cache line 的历史访问频率以及 Cache line 对应的内存的类型,判断所述 Cache 中是否存在满足第一预设条件的 Cache line,其中,所述满足第一预设条件的 Cache line 包括历史访问频率低于预设频率且与动态随机访问存储器 DRAM 类型的内存对应的 Cache line,所述内存包括 DRAM 类型的内存和非易失性存储器 NVM 类型的内存;

在确定所述 Cache 中存在满足第一预设条件的 Cache line 的情况下,在满足第一预设条件的 Cache line 中选择待替换的第一 Cache line;

根据所述访问地址从所述内存中读取所述待访问数据;

采用第二 Cache line 替换所述第一 Cache line,所述第二 Cache line 包含所述访问地址和所述待访问数据;

所述 Cache 向所述 CPU 发送所述待访问数据。

2. 根据权利要求 1 所述的方法,其特征在于,所述判断所述 Cache 中是否存在满足第一预设条件的 Cache line,包括:

判断所述 Cache 的最近最少访问 LRU 链表中前 M 个 Cache line 中是否存在与 DRAM 类型的内存对应的 Cache line,其中所述 LRU 链表中前 M 个 Cache line 为历史访问频率低于预设频率的 Cache line;

所述在确定所述 Cache 中存在满足第一预设条件的 Cache line 的情况下,在满足第一预设条件的 Cache line 中选择待替换的第一 Cache line,包括:

在确定所述 LRU 链表中前 M 个 Cache line 中存在与 DRAM 类型的内存对应的 Cache line 的情况下,在确定的所述前 M 个 Cache line 中与 DRAM 类型的内存对应的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。

3. 根据权利要求 1 或 2 所述的方法,其特征在于,还包括:

在确定所述 Cache 中不存在满足第一预设条件的 Cache line 的情况下,判断所述 Cache 中是否存在满足第二预设条件的 cache line,其中所述满足第二预设条件的 cache line 包括历史访问频率低于所述预设频率、与 NVM 类型的内存对应并且未被修改过的 Cache line;

在确定所述 Cache 中存在满足第二预设条件的 cache line 的情况下,在所述满足第二预设条件的 cache line 中选择待替换的第一 Cache line。

4. 根据权利要求 3 所述的方法,其特征在于,所述判断所述 Cache 中是否存在满足第二预设条件的 cache line,包括:

判断所述 Cache 的 LRU 链表中前 N 个 Cache line 中是否存在与 NVM 类型的内存对应并且修改 Modify 标识位表示干净 clean 的 Cache line,其中所述前 N 个 Cache line 为历史访问频率低于预设频率的 Cache line;

所述在确定所述 Cache 中存在满足第二预设条件的 cache line 的情况下,在所述满足第二预设条件的 cache line 中选择待替换的第一 Cache line 包括:

在确定所述 LRU 链表中前 N 个 Cache line 中存在与 NVM 类型的内存对应且 Modify 位

为 clean 的 Cache line 的情况下,在确定的所述前 N 个 Cache line 中与 NVM 类型的内存对应且 Modify 位为 clean 的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。

5. 根据权利要求 4 所述的方法,其特征在于,还包括:

在确定所述 Cache 中不存在满足第二预设条件的 cache line 的情况下,确定所述 LRU 链表中最前端的 Cache line 为所述第一 Cache line。

6. 根据权利要求 1~5 中任一项所述的方法,其特征在于,所述采用第二 Cache line 替换所述第一 Cache line 之后,还包括:

根据所述访问地址指向的所述内存中的存储介质的类型,记录所述第二 Cache line 对应的内存的类型。

7. 一种缓存 Cache,其特征在于,包括:

接收模块,用于接收 CPU 发送的数据访问请求,所述数据访问请求中包含访问地址;

命中确定模块,用于根据所述访问地址判断待访问数据是否缓存在所述 Cache 中;

替换确定模块,用于在确定所述待访问数据没有缓存在所述 Cache 中的情况下,根据所述 Cache 中的缓存线 Cache line 的历史访问频率以及 Cache line 对应的内存的类型,判断所述 Cache 中是否存在满足第一预设条件的 Cache line,其中,所述满足第一预设条件的 Cache line 包括历史访问频率低于预设频率且与动态随机访问存储器 DRAM 类型的内存对应的 Cache line,所述内存包括 DRAM 类型的内存和非易失性存储器 NVM 类型的内存;在确定所述 Cache 中存在满足第一预设条件的 Cache line 的情况下,在满足第一预设条件的 Cache line 中选择待替换的第一 Cache line;

读取模块,用于根据所述访问地址从所述内存中读取所述待访问数据;

替换模块,用于采用第二 Cache line 替换所述第一 Cache line,所述第二 Cache line 包含所述访问地址和所述待访问数据;

发送模块,用于向所述 CPU 发送所述待访问数据。

8. 根据权利要求 7 所述的 Cache,其特征在于,所述替换确定模块,具体用于:

判断所述 Cache 的最近最少访问 LRU 链表中前 M 个 Cache line 中是否存在与 DRAM 类型的内存对应的 Cache line,其中所述 LRU 链表中前 M 个 Cache line 为历史访问频率低于预设频率的 Cache line;

在确定所述 LRU 链表中前 M 个 Cache line 中存在与 DRAM 类型的内存对应的 Cache line 的情况下,在确定的所述前 M 个 Cache line 中与 DRAM 类型的内存对应的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。

9. 根据权利要求 7 或 8 所述的 Cache,其特征在于,所述替换确定模块,还用于:

在确定所述 Cache 中不存在满足第一预设条件的 Cache line 的情况下,判断所述 Cache 中是否存在满足第二预设条件的 cache line,其中所述满足第二预设条件的 cache line 包括历史访问频率低于所述预设频率、与 NVM 类型的内存对应并且未被修改过的 Cache line;

在确定所述 Cache 中存在满足第二预设条件的 cache line 的情况下,在所述满足第二预设条件的 cache line 中选择待替换的第一 Cache line。

10. 根据权利要求 9 所述的 Cache,其特征在于,所述替换确定模块,具体用于:

判断所述 Cache 的 LRU 链表中前 N 个 Cache line 中是否存在与 NVM 类型的内存对应

并且修改 Modify 标识位表示干净 clean 的 Cache line,其中所述前 N 个 Cache line 为历史访问频率低于预设频率的 Cache line;

在确定所述 LRU 链表中前 N 个 Cache line 中存在与 NVM 类型的内存对应且 Modify 位为 clean 的 Cache line 的情况下,在确定的所述前 N 个 Cache line 中与 NVM 类型的内存对应且 Modify 位为 clean 的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。

11. 根据权利要求 10 所述的 Cache,其特征在于,所述替换确定模块,还用于:

在确定所述 Cache 中不存在满足第二预设条件的 cache line 的情况下,确定所述 LRU 链表中最前端的 Cache line 为所述第一 Cache line。

12. 根据权利要求 7 ~ 11 中任一项所述的 Cache,其特征在于,所述替换模块,还用于:

在采用第二 Cache line 替换所述第一 Cache line 之后,根据所述访问地址指向的所述内存中的存储介质的类型,记录所述第二 Cache line 对应的内存的类型。

13. 一种计算机系统,其特征在于,包括:处理器、内存和如权利要求 7 ~ 12 中任一项所述的缓存 Cache,所述内存包括 DRAM 和 NVM,所述处理器、所述内存与所述 Cache 通过总线连接。

数据缓存方法、缓存和计算机系统

技术领域

[0001] 本发明实施例涉及存储技术,尤其涉及一种数据缓存方法、缓存和计算机系统。

背景技术

[0002] 目前越来越多的应用以数据为中心,例如互联网应用、大数据应用等。这些应用需要强大的存储支持。

[0003] 在现有技术中,通常采用的动态随机访问存储器 (Dynamic Random-Access Memory,以下简称:DRAM) 作为计算机系统。然而,受工艺的限制,DRAM 的容量较小且能耗较大,很难满足应用对大容量低能耗的要求。今年来,非易失性存储器 (Non-Volatile Memory,以下简称:NVM) 被广泛采用,其具有存储量大、能耗低的优势,采用 NVM 取代 DRAM 作为计算机系统,能够满足应用对大容量低能耗的需求。但是与 DRAM 相比,NVM 的读写延迟较长。正是由于 DRAM 和 NVM 各有利弊,现有技术进一步采用 DRAM 和 NVM 组成的混合内存,以期望为应用提供大容量、低能耗、高性能的内存。

发明内容

[0004] 本发明实施例提供一种数据缓存方法、缓存和计算机系统。

[0005] 第一方面,本发明实施例提供一种数据缓存方法,所述方法由缓存 Cache 执行,包括:

[0006] 接收 CPU 发送的数据访问请求,所述数据访问请求中包含访问地址;

[0007] 根据所述访问地址判断待访问数据是否缓存在所述 Cache 中;

[0008] 在确定所述待访问数据没有缓存在所述 Cache 中的情况下,根据所述 Cache 中的缓存线 Cache line 的历史访问频率以及 Cache line 对应的内存的类型,判断所述 Cache 中是否存在满足第一预设条件的 Cache line,其中,所述满足第一预设条件的 Cache line 包括历史访问频率低于预设频率且与动态随机访问存储器 DRAM 类型的内存对应的 Cache line,所述内存包括 DRAM 类型的内存和非易失性存储器 NVM 类型的内存;

[0009] 在确定所述 Cache 中存在满足第一预设条件的 Cache line 的情况下,在满足第一预设条件的 Cache line 中选择待替换的第一 Cache line;

[0010] 根据所述访问地址从所述内存中读取所述待访问数据;

[0011] 采用第二 Cache line 替换所述第一 Cache line,所述第二 Cache line 包含所述访问地址和所述待访问数据;

[0012] 所述 Cache 向所述 CPU 发送所述待访问数据。

[0013] 第二方面,本发明实施例提供一种缓存 Cache,包括:

[0014] 接收模块,用于接收 CPU 发送的数据访问请求,所述数据访问请求中包含访问地址;

[0015] 命中确定模块,用于根据所述访问地址判断待访问数据是否缓存在所述 Cache 中;

[0016] 替换确定模块,用于在确定所述待访问数据没有缓存在所述 Cache 中的情况下,

根据所述 Cache 中的缓存线 Cache line 的历史访问频率以及 Cache line 对应的内存的类型,判断所述 Cache 中是否存在满足第一预设条件的 Cache line,其中,所述满足第一预设条件的 Cache line 包括历史访问频率低于预设频率且与动态随机访问存储器 DRAM 类型的内存对应的 Cache line,所述内存包括 DRAM 类型的内存和非易失性存储器 NVM 类型的内存;在确定所述 Cache 中存在满足第一预设条件的 Cache line 的情况下,在满足第一预设条件的 Cache line 中选择待替换的第一 Cache line;

[0017] 读取模块,用于根据所述访问地址从所述内存中读取所述待访问数据;

[0018] 替换模块,用于采用第二 Cache line 替换所述第一 Cache line,所述第二 Cache line 包含所述访问地址和所述待访问数据;

[0019] 发送模块,用于向所述 CPU 发送所述待访问数据。

[0020] 第三方面,本发明实施例提供一种计算机系统,包括:处理器、混合内存和如前述第二方面所述的缓存 Cache,所述混合内存包括 DRAM 和 NVM,所述处理器、所述混合内存与所述缓存 Cache 通过总线连接。

[0021] 本发明实施例中,Cache 在访问请求不命中需要确定待替换的 Cache line 时,不仅需要考虑到 Cache line 的历史访问频率,还要考虑到 Cache line 对应的内存类型,从而可以优先替换与 DRAM 内存类型对应的 Cache line,减少 Cache 对存储于 DRAM 的数据的缓存量,从而使得 Cache 能够提高对存储于 NVM 中的数据的数据的缓存量,使得针对存储于 NVM 中的数据的访问请求尽可能的在 Cache 中能够找到相应的数据,从而减少到 NVM 中读取数据的情形,从而减少了从 NVM 中读取数据的延迟,有效的提高访问效率。

附图说明

[0022] 为了更清楚地说明本发明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性的前提下,还可以根据这些附图获得其他的附图。

[0023] 图 1 为本发明实施例提供的一种计算机系统的架构图;

[0024] 图 2 为本发明实施例提供的一种数据缓存方法的流程图;

[0025] 图 3 为本发明实施例提供的一种缓存的结构示意图。

具体实施方式

[0026] 为使本发明实施例的目的、技术方案和优点更加清楚,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,显然,所描述的实施例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,都属于本发明保护的范围。

[0027] 图 1 为本发明实施例提供的一种计算机系统的架构图,如图 1 所示,该计算机系统包括:处理器 (Processor) 11 和混合内存 (Hybrid Memory) 12,其中,Processor 11 可以包括 CPU 111、缓存 (Cache) 112 以及存储控制器 (Memory controller) 113,Hybrid Memory 12 可以包括 DRAM 121 和 NVM 122,Processor 11、Hybrid Memory 12 以及缓存 (Cache) 112 通过总线连接。Hybrid Memory 12 与 Memory controller 113 之间可以通过内存总线 (Memory Bus) 13 连接。

[0028] 在进行数据访问时,CPU 可以向 Cache 发出数据访问请求,该数据访问请求中包含访问地址。Cache 中缓存了 Hybrid Memory 中的部分数据以提高响应速度。因此,Cache 首先根据访问地址确定 CPU 请求访问的数据是否缓存在 Cache 中,换一种表达方式,Cache 首先根据访问地址判断是否能够 Cache 命中所述请求访问的数据。当 Cache 命中时,即在确定所述待访问数据缓存在所述 Cache 中的情况下,Cache 可以直接向 CPU 返回其请求访问的数据,当 Cache 没有命中时,即在确定所述待访问数据没有缓存在所述 Cache 中的情况下,该数据访问请求将会通过 Memory controller 发往 Hybrid Memory,以从 Hybrid Memory 中读取 CPU 请求访问的数据。

[0029] 由于 Cache 的缓存空间一般较小,因此,Cache 在数据访问过程中,需要根据访问情况不断更新其缓存内容,以满足不断变化的访问需求。具体来说,当本次数据访问在 Cache 中命中时,则不进行缓存线 (Cache line) 的替换更新,当本次数据访问在 Cache 中不命中时,Cache 需要从当前缓存的缓存线 (Cache line) 中确定一待替换的 Cache line,采用从内存中读取的新的 Cache line 来替换该 Cache line。

[0030] 本领域技术人员可以知道,Cache 与内存交换数据是通过缓存控制器 Cache controller 来执行的。其中,Cache line 是 Cache Controller 的最小操作单位。换一种表达方式,当 Cache Controller 向内存中写数据时,Cache Controller 是按照 Cache line 为单位将一行 line 数据写到内存中,当 Cache Controller 从内存中读数据时,也是按照 Cache line 来读数据的。为了描述方便,在本发明实施例中,一个 Cache line 可以表示一个 Cache line 的数据。本发明实施例中的“替换 Cache line”是指用从内存中读出的一个 Cache line 的数据来替换 Cache 中的一个 Cache line 的数据。在上述过程中,Cache 是从当前缓存的 Cache line 中查找访问频率最低的 Cache line,将该访问频率最低的 Cache line 确定为待替换的 Cache line。而 Cache 并不感知该待替换的 Cache line 对应的内存类型是 DRAM 类型还是 NVM 类型,即不感知该待替换的 Cache line 是来源于 DRAM 还是 NVM。

[0031] 经过对 DRAM 和几种 NVM 进行的分析发现,DRAM 和 NVM 的读写性能存在差异。其中,在本发明实施例中,NVM 包括但不限于:相变存储器 (Phase Change Memory,以下简称:PCM)、自旋转移矩磁性随机访问存储器 (Spin Transfer Torque-Magnetic Random Access Memory,以下简称:STT-MRAM)、阻变式随机访问存储器 (Resistive Random Access Memory,以下简称 RRAM)。DRAM 和 NVM 的读写性能具体可以如表 1 所示:

[0032] 表 1

[0033]

	DRAM	PCM	STT-MRAM	RRAM
读延迟	<10ns	12ns	35ns	<50ns
写延迟	<10ns	100ns	35ns	<0.3ns
写次数	>1E16	1.00E+09	>1E12	1.00E+12
保留时间	64ms	>10y	>10y	>10y

[0034] 从表 1 中可以看出,虽然 NVM 的存储容量大于 DRAM,且能耗低于 DRAM,但是 NVM 的

读写延迟大于 DRAM 的读写延迟,并且 NVM 有写次数的限制。

[0035] 因此,采用上述现有技术,如果 Cache 确定的待替换的 Cache line 对应的内存类型为 NVM 类型,则该待替换的 Cache line 将被从 Cache 中删除,后续访问请求都必须要从 Hybrid Memory 的 NVM 中调取,而 NVM 的读写延迟相对 DRAM 又较大,则必然带来访问延迟的问题,无法满足应用对访问延迟较高的需求。

[0036] 因此,本方法实施例中,Cache 确定待替换的 Cache line 时,不仅考虑当前缓存的各 Cache line 的历史访问频率,还要进一步考虑各 Cache line 对应的内存类型,优先替换内存类型为 DRAM 类型的 Cache line,即优先替换内存类型为 DRAM 的 Cache line。这样,即使 DRAM 的 Cache line 从 Cache 中删除,后续数据访问需要到 DRAM 中调取数据,其访问延迟不至于过大。其中,内存类型为 DRAM 类型的 Cache line,即表示该 Cache line 中的数据存储在内存中的 DRAM 上。内存类型为 NVM 类型的 Cache line,即表示该 Cache line 中的数据存储在内存中的 NVM 上。

[0037] 下面采用具体的实施例对本发明的技术方案进行详细说明。

[0038] 图 2 为本发明实施例提供的一种数据缓存方法的流程图,该方法可以由 Cache 来执行,具体的可以由 Cache 中的 Cache Controller 来执行。如图 1 和图 2 所示,本实施例的方法可以包括:

[0039] S201、Cache 接收 CPU 发送的数据访问请求,该数据访问请求中包含访问地址;

[0040] S202、Cache 根据所述访问地址判断待访问数据是否缓存在所述 Cache 中,若是,则执行 S207,若否,则执行 S203;

[0041] 具体来说,CPU 可以接收应用发送的数据访问请求,从而将该数据访问请求发送给 Cache。Cache 可以将 CPU 请求的访问地址与其缓存的各 Cache line 的地址进行比较,确定请求的数据是否缓存在 Cache 中,即是否命中。在具体实现时,如果 Cache 的映射策略是全相连,则 Cache 是在整个缓存范围内进行查找比较,如果 Cache 的映射策略不是全相连,而是采用组划分的方式,则 Cache 可以根据访问地址中的索引(index)位,确定该访问地址在 Cache 中所在的组,进而根据访问地址中的位(tag)位确定在该组中是否包含该访问地址,如果包含,则 Cache 可以根据有效位判断缓存的数据是否有效,如果数据有效,则可以根据访问地址中的数据偏移找到对应数据,并将该数据返回给 CPU。

[0042] S203、Cache 根据所述 Cache 中的缓存线 Cache line 的历史访问频率以及 Cache line 对应的内存的类型,判断所述 Cache 中是否存在满足第一预设条件的 Cache line,若是,则执行 S204,否则执行 S208;

[0043] 其中,满足第一预设条件的 Cache line 包括历史访问频率低于预设频率且与动态随机访问存储器 DRAM 类型的内存对应的 Cache line,所述内存包括 DRAM 类型的内存和非易失性存储器 NVM 类型的内存;

[0044] S204、在满足第一预设条件的 Cache line 中选择待替换的第一 Cache line;

[0045] 在待访问数据未缓存在 Cache 中,即未命中的情况下,Cache 需要从当前缓存的 Cache line 中确定一个 Cache line 作为待替换的 Cache line,该被确定作为待替换的 Cache line 即为第一 Cache line。在确定该第一 Cache line 时,既需要参考当前缓存的各 Cache line 的历史访问频率,也需要参考各 Cache line 对应的内存类型。其中,Cache line 的历史访问频率用于表征对应缓存数据的访问热度,Cache line 对应的内存类型则表

征该 Cache line 的来源是 DRAM 还是 NVM。

[0046] 本实施例中, Cache 确定第一 Cache line 的策略可以为在考虑历史访问频率的基础上, 尽可能替换内存类型为 DRAM 的 Cache line, 即优先替换内存类型为 DRAM 类型的 Cache line。

[0047] 举例来说, Cache 可以从缓存的所有 Cache line 中先选取历史访问频率低于预设频率的几个 Cache line, 然后再根据这几个 Cache line 中每个 Cache line 对应的内存类型, 确定与 DRAM 类型的内存对应的 Cache line 作为待替换的第一 Cache line。若这几个 Cache line 中包含两个或者更多个内存类型为 DRAM 的 Cache line, 则可以确定历史访问频率最低且内存类型为 DRAM 的 Cache line 为待替换的第一 Cache line。

[0048] S205、Cache 根据所述访问地址从所述内存中读取所述待访问数据;

[0049] Cache 在确定了待替换的第一 Cache line 之后, 即可根据访问地址从内存中读取待访问数据, 该待访问数据既可能存储在 DRAM 上, 也可能存储在 NVM 上。

[0050] S206、Cache 采用第二 Cache line 替换所述第一 Cache line, 所述第二 Cache line 包含所述访问地址和所述待访问数据;

[0051] Cache 在读取待访问数据之后, 可以采用第二 Cache line 替换第一 Cache line。需要说明的是, 本发明实施例中, 为了描述方便, 将从内存中读出的一个 Cache line 的数据称为第二 Cache line。采用第二 Cache line 替换所述第一 Cache line 是指将从内存中读出的数据缓存在 Cache 中, 并将第一 Cache line 的数据删除或写回内存中。在具体实现时, 可以在 Cache 中针对各个 Cache line 增加一个标识位 (Location), 通过该标识位来标识该 Cache line 对应的内存类型是 DRAM 类型还是 NVM 类型。

[0052] Cache 可以根据访问地址指向的内存中的存储介质的类型, 记录第二 Cache line 对应的内存的类型, 即设置该标志位。例如, 该标识位可以采用一个 bit, 该 bit 为 1 则表征对应的内存类型为 DRAM 类型, 该 bit 为 0 则代表对应的内存类型为 NVM 类型。

[0053] 在从 Cache 中删除第一 Cache line 时, 若待替换的第一 Cache line 的修改 (Modify) 位为干净 (clean), 则表示第一 Cache line 中的数据并未被修改过, 与内存中存储的数据一致, 则此时无需将第一 Cache line 的数据写回到内存中, 可以直接删除第一 Cache line 的数据。若待替换的第一 Cache line 的 Modify 位为脏 (dirty), 则表示第一 Cache line 中的数据被修改过, 与内存中存储的数据不一致, 则此时需要先将第一 Cache line 的数据写回到内存中, 再删除该第一 Cache line。

[0054] S207、Cache 向 CPU 发送待访问数据, 结束。

[0055] 在完成上述操作后, Cache 可以向 CPU 发送读取的待访问数据。

[0056] 综上, 本实施例的方法中, Cache 在访问请求不命中需要确定待替换的 Cache line 时, 不仅需要考虑 Cache line 的历史访问频率, 还要考虑 Cache line 对应的内存类型, 从而可以优先替换与 DRAM 内存类型对应的 Cache line, 减少 Cache 对存储于 DRAM 的数据的缓存量, 从而使得 Cache 能够提高对存储于 NVM 中的数据的数据的缓存量, 使得针对存储于 NVM 中的数据的数据的访问请求尽可能的在 Cache 中能够找到相应的数据, 从而减少到 NVM 中读取数据的情形, 从而减少了从 NVM 中读取数据的延迟, 有效的提高访问效率。

[0057] 上述实施例可以借助最近最少访问 (Least Recently Used, 以下简称: LRU) 链表来获取各 Cache line 的历史访问频率。

[0058] 具体来说,LRU 链表是按照访问频率从低到高的顺序记录 Cache line,Cache 在判断 Cache 中是否存在满足第一预设条件的 Cache line 时,具体可以为:

[0059] 判断所述 Cache 的最近最少访问 LRU 链表中前 M 个 Cache line 中是否存在与 DRAM 类型的内存对应的 Cache line,其中所述 LRU 链表中前 M 个 Cache line 为历史访问频率低于预设频率的 Cache line;

[0060] 在确定 LRU 链表中前 M 个 Cache line 中存在与 DRAM 类型的内存对应的 Cache line 的情况下,在确定的所述前 M 个 Cache line 中与 DRAM 类型的内存对应的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。

[0061] 其中,“第一个”是指的在 LRU 链表的前 M 个 cache line 中的与 DRAM 类型的内存对应的 Cache line 中位置最靠前的 Cache line。

[0062] 需要说明的是,若 M 的取值设置得较大时,则可以提高内存类型为 DRAM 类型的 Cache line 被替换的概率,但 M 的取值又不能设置得过大,否则 DRAM 中存储的数据将无法进入 Cache。本领域技术人员可以根据需求,自行设定 M 的取值。

[0063] S208、Cache 判断该 Cache 中是否存在满足第二预设条件的 cache line,若是,则执行 S209,否则执行 S210;

[0064] 其中该满足第二预设条件的 cache line 包括历史访问频率低于所述预设频率、与 NVM 类型的内存对应并且未被修改过的 Cache line;

[0065] 具体来说,Cache 在确定所述 Cache 中不存在满足第一预设条件的 Cache line 的情况下,此时,Cache 可能不得不替换内存类型为 NVM 的 Cache line。

[0066] 经过分析发现,Cache line 的 Modify 位为 clean,即 Cache line 中的数据并未被修改过,与内存中存储的数据一致,则在替换时无需将 Cache line 的数据写回到内存中,若 Cache line 的 Modify 位为 Dirty,即 Cache line 中的数据被修改过,与内存中存储的数据不一致,则在替换时需要先将 Cache line 的数据写回到内存中。但是,NVM 具有写次数限制,因此,当不得不替换内存类型为 NVM 的 Cache line 时,可以优先替换 Modify 位为 clean 的 Cache line,从而尽可能减少 NVM 的写次数。

[0067] S209、在满足第二预设条件的 cache line 中选择待替换的第一 Cache line,并执行 S205。

[0068] 其中,Cache 判断该 Cache 中是否存在满足第二预设条件的 cache line,可以具体为:

[0069] 判断所述 Cache 的 LRU 链表中前 N 个 Cache line 中是否存在与 NVM 类型的内存对应并且修改 Modify 标识位表示干净 clean 的 Cache line,其中所述前 N 个 Cache line 为历史访问频率低于预设频率的 Cache line;

[0070] 在确定所述 LRU 链表中前 N 个 Cache line 中存在与 NVM 类型的内存对应且 Modify 位为 clean 的 Cache line 的情况下,在确定的所述前 N 个 Cache line 中与 NVM 类型的内存对应且 Modify 位为 clean 的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。

[0071] 在具体实现时,M 和 N 的值可以根据应用行为调优。因为不同内存类型的 Cache line,其替换延迟存在如下关系:DRAM<NVM(clean)<NVM(dirty),因此,可以适当延长来自 NVM 的 Cache line 在 Cache 中的存储时间。因此,一般可以设定 $N \leq M$ 。

[0072] S210、Cache 选择历史访问频率最低的 Cache line 为所述第一 Cache line,并执行 S205。

[0073] 进一步的,在确定所述 Cache 中不存在满足第二预设条件的 cache line 的情况下,确定所述 LRU 链表中最前端的 Cache line 为所述第一 Cache line。

[0074] 具体来说,若 LRU 链表中前 N 个 Cache line 中也不存在内存类型为 NVM 类型且 Modify 位为 clean 的 Cache line,则 Cache 即可确定 LRU 链表中最前端的 Cache line 为第一 Cache line。此时确定的第一 Cache line 即为内存类型为 NVM 类型且 Modify 位为 dirty 的 Cache line。

[0075] 在本发明另一个实施例中,Cache 还可以根据访问地址所指向的内存中的存储介质的类型,记录所述第二 Cache line 对应的内存的类型。

[0076] 具体来说,该 Cache 可以采用以下两种实现方式来获取 Cache line 对应的内存的类型:

[0077] 方式一、根据访问地址在内存中所属的地址范围确定第二 Cache line 对应的内存类型并记录;

[0078] 具体来说,在该方式中,Hybrid Memory 中的物理地址是连续的,例如前 nGB 为 DRAM,后 nGB 为 NVM。

[0079] 因此,Cache 可以确定访问地址属于 DRAM 的地址范围还是属于 NVM 的地址范围;若访问地址属于 DRAM 的地址范围,则 Cache 可以记录第二 Cache line 对应的内存类型为 DRAM 类型;若访问地址属于 NVM 的地址范围,则 Cache 可以记录第二 Cache line 对应的内存类型为 NVM 类型。

[0080] 方式二、根据 Memory Controller 反馈的信息确定第二 Cache line 对应的内存类型并记录。

[0081] 具体来说,在该方式中,Memory Controller 中可以保存一映射表,该映射表记录有 DRAM 的地址范围或 NVM 的地址范围,或者,也可以同时记录 DRAM 的地址范围和 NVM 的地址范围,该地址范围可以连续,也可以不连续。Memory Controller 可以根据访问地址和该映射表确定访问地址在 Hybrid Memory 中的存储位置。而且,Memory Controller 和 Cache 的交互数据中可以增加一个位,Memory Controller 可以将访问地址在 Hybrid Memory 中的存储位置通过该增加的位发送给 Cache。

[0082] 在具体实现时,Cache 在未命中时,需要向 Memory Controller 发送数据读取请求,该数据读取请求中包含访问地址;Memory Controller 可以根据该访问请求到 Hybrid Memory 中读取待访问数据,并且 Memory Controller 可以根据该访问地址和映射表获取访问地址在 Hybrid Memory 中的存储位置,此后,Memory Controller 可以向 Cache 发送数据读取响应,该数据读取响应中包含待访问数据以及访问地址在 Hybrid Memory 中的存储位置;相应的,Cache 可以根据 Memory Controller 发送的数据读取响应中包含的存储位置,记录第二 Cache line 对应的内存类型。

[0083] 下面采用两个具体的实施例,对上述实施例的技术方案进行详细说明。

[0084] 假设:Hybrid Memory 中,前 2GB 为 DRAM,后 2GB 为 NVM,Cache 映射策略为全相连;CPU 向 Cache 请求数据,访问地址为 0x08000000,该访问地址位于 Hybrid Memory 第 1GB,数据总线为 32 位。

[0085] 例一 :Cache 在本地查找后,未找到对应数据,说明未命中,则 Cache 查找 LRU 链表前 5 个 Cache line,并根据这前 5 个 Cache line 的 location 位,确定未找到内存类型为 DRAM 的 Cache line,则 Cache 查看 LRU 链表前 3 个 Cache line 的 Modify 位,找到第 2 个 Cache line 的 Modify 位为 0,即表示 clean,则该 Cache line 即为待替换的第一 Cache line。Cache 可以将包含访问地址为 0x08000000 的数据的 Cache line 读入到 Cache 中替换第一 Cache line,并将读取到的数据返回给 CPU。并且,Cache 可以根据访问地址 (0x08000000) 判断数据存储在 DRAM 中,因此将新读入的 Cache line 即第二 Cache line 加入到 LRU 链表的最尾端,并且将其 location 位置 0 (表征该第二 Cache line 来源于 DRAM),并且 Modify 位置为 0 (表征该第二 Cache line 的数据未被修改过)。

[0086] 例二 :Cache 在本地查找后,未找到对应数据,说明未命中,则 Cache 查找 LRU 链表前 5 个 Cache line,并根据这前 5 个 Cache line 的 location 位,确定找到内存类型为 DRAM 的 Cache line,则该 Cache line 即为待替换的第一 Cache line。若该 DRAM 的 Modify 位为 1 (表征该第一 Cache line 的数据被修改过),则 Cache 先对第一 Cache line 进行数据写回,然后,Cache 可以将包含访问地址为 0x08000000 的数据的 Cache line 读入到 Cache 中以替换第一 Cache line,并将读取到的数据返回给 CPU。并且,Cache 可以根据访问地址 (0x08000000) 判断数据存储在 DRAM 中,因此将新读入的 Cache line 即第二 Cache line 加入到 LRU 链表的最尾端,并且将其 location 位置 0 (表征该第二 Cache line 来源于 DRAM),并且 Modify 位置为 0 (表征该第二 Cache line 的数据未被修改过)。

[0087] 由上述过程可知,本发明实施例中,Cache 在确定待替换的 Cache line 时,优先替换内存类型为 DRAM 的 Cache line,从而尽量保留内存类型为 NVM 的 Cache line,以避免从 NVM 中读取数据的延迟;在没有可替换的内存类型为 DRAM 的 Cache line 而只能替换内存类型为 NVM 的 Cache line 时,优先替换内存类型为 NVM 且 Modify 位为 clean 的 Cache line,从而尽可能减少对 NVM 的写次数,提高内存的使用寿命。

[0088] 图 3 为本发明实施例提供的一种缓存 Cache 的结构示意图,如图 3 所示,本实施例的 Cache 可以包括:

[0089] 接收模块 31,用于接收 CPU 发送的数据访问请求,所述数据访问请求中包含访问地址;

[0090] 命中确定模块 32,用于根据所述访问地址判断待访问数据是否缓存在所述 Cache 中;

[0091] 替换确定模块 33,用于用于在确定所述待访问数据没有缓存在所述 Cache 中的情况下,根据所述 Cache 中的缓存线 Cache line 的历史访问频率以及 Cache line 对应的内存的类型,判断所述 Cache 中是否存在满足第一预设条件的 Cache line,其中,所述满足第一预设条件的 Cache line 包括历史访问频率低于预设频率且与动态随机访问存储器 DRAM 类型的内存对应的 Cache line,所述内存包括 DRAM 类型的内存和非易失性存储器 NVM 类型的内存;在确定所述 Cache 中存在满足第一预设条件的 Cache line 的情况下,在满足第一预设条件的 Cache line 中选择待替换的第一 Cache line;

[0092] 读取模块 34,用于根据所述访问地址从所述内存中读取所述待访问数据;

[0093] 替换模块 35,用于采用第二 Cache line 替换所述第一 Cache line,所述第二 Cache line 包含所述访问地址和所述待访问数据;

- [0094] 发送模块 36,用于向所述 CPU 发送所述待访问数据。
- [0095] 进一步的,替换确定模块 33,具体用于:
- [0096] 替换确定模块,具体用于:
- [0097] 判断所述 Cache 的最近最少访问 LRU 链表中前 M 个 Cache line 中是否存在与 DRAM 类型的内存对应的 Cache line,其中所述 LRU 链表中前 M 个 Cache line 为历史访问频率低于预设频率的 Cache line;
- [0098] 在确定所述 LRU 链表中前 M 个 Cache line 中存在与 DRAM 类型的内存对应的 Cache line 的情况下,在确定的所述前 M 个 Cache line 中与 DRAM 类型的内存对应的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。
- [0099] 进一步的,替换确定模块 33,还用于:
- [0100] 在确定所述 Cache 中不存在满足第一预设条件的 Cache line 的情况下,判断所述 Cache 中是否存在满足第二预设条件的 cache line,其中所述满足第二预设条件的 cache line 包括历史访问频率低于所述预设频率、与 NVM 类型的内存对应并且未被修改过的 Cache line;
- [0101] 在确定所述 Cache 中存在满足第二预设条件的 cache line 的情况下,在所述满足第二预设条件的 cache line 中选择待替换的第一 Cache line。
- [0102] 进一步的,替换确定模块 33,具体用于:
- [0103] 判断所述 Cache 的 LRU 链表中前 N 个 Cache line 中是否存在与 NVM 类型的内存对应并且修改 Modify 标识位表示干净 clean 的 Cache line,其中所述前 N 个 Cache line 为历史访问频率低于预设频率的 Cache line;
- [0104] 在确定所述 LRU 链表中前 N 个 Cache line 中存在与 NVM 类型的内存对应且 Modify 位为 clean 的 Cache line 的情况下,在确定的所述前 N 个 Cache line 中与 NVM 类型的内存对应且 Modify 位为 clean 的 Cache line 中选择第一个 Cache line 为所述第一 Cache line。
- [0105] 更进一步的,替换确定模块 33,还用于:
- [0106] 在确定所述 Cache 中不存在满足第二预设条件的 cache line 的情况下,确定所述 LRU 链表中最前端的 Cache line 为所述第一 Cache line。
- [0107] 进一步的,替换模块 35,还用于:
- [0108] 在采用第二 Cache line 替换所述第一 Cache line 之后,根据所述访问地址指向的所述内存中的存储介质的类型,记录所述第二 Cache line 对应的内存的类型。
- [0109] 本实施例的 Cache 可以用于执行上述方法实施例的技术方案,其实现原理和技术效果类似,此处不再赘述。
- [0110] 本发明实施例,还可以提供一计算机系统实施例,该计算机系统实施例的结构示意图可以参考图 1 所示架构,即包括处理器 (Processor)11 和混合内存 (Hybrid Memory)12,其中,Processor11 可以包括 CPU111、CPU 缓存 (Cache)112 以及存储控制器 (Memory controller)113,Hybrid Memory12 可以包括 DRAM121 和 NVM122,Hybrid Memory12 与 Memory controller113 之间可以通过内存总线 (Memory Bus)13 连接;
- [0111] 其中,Cache112 可以采用上述 Cache 实施例所述的结构,且该 Cache112 可以执行上述方法实施例的技术方案。其实现原理和技术效果类似,此处不再赘述。

[0112] 本领域普通技术人员可以理解：实现上述各方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成。前述的程序可以存储于一计算机可读取存储介质中。该程序在执行时，执行包括上述各方法实施例的步骤；而前述的存储介质包括：ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0113] 最后应说明的是：以上各实施例仅用以说明本发明的技术方案，而非对其限制。本申请所提供的实施例仅仅是示意性的。所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中未详述的部分，可以参见其他实施例的相关描述。在本发明实施例、权利要求以及附图中揭示的特征可以独立存在也可以组合存在。

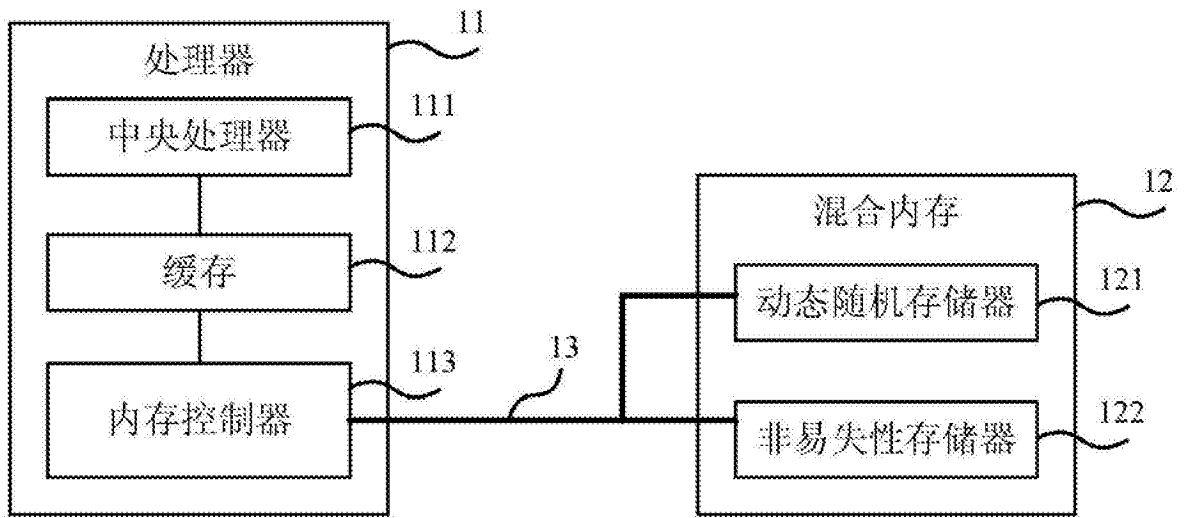


图 1

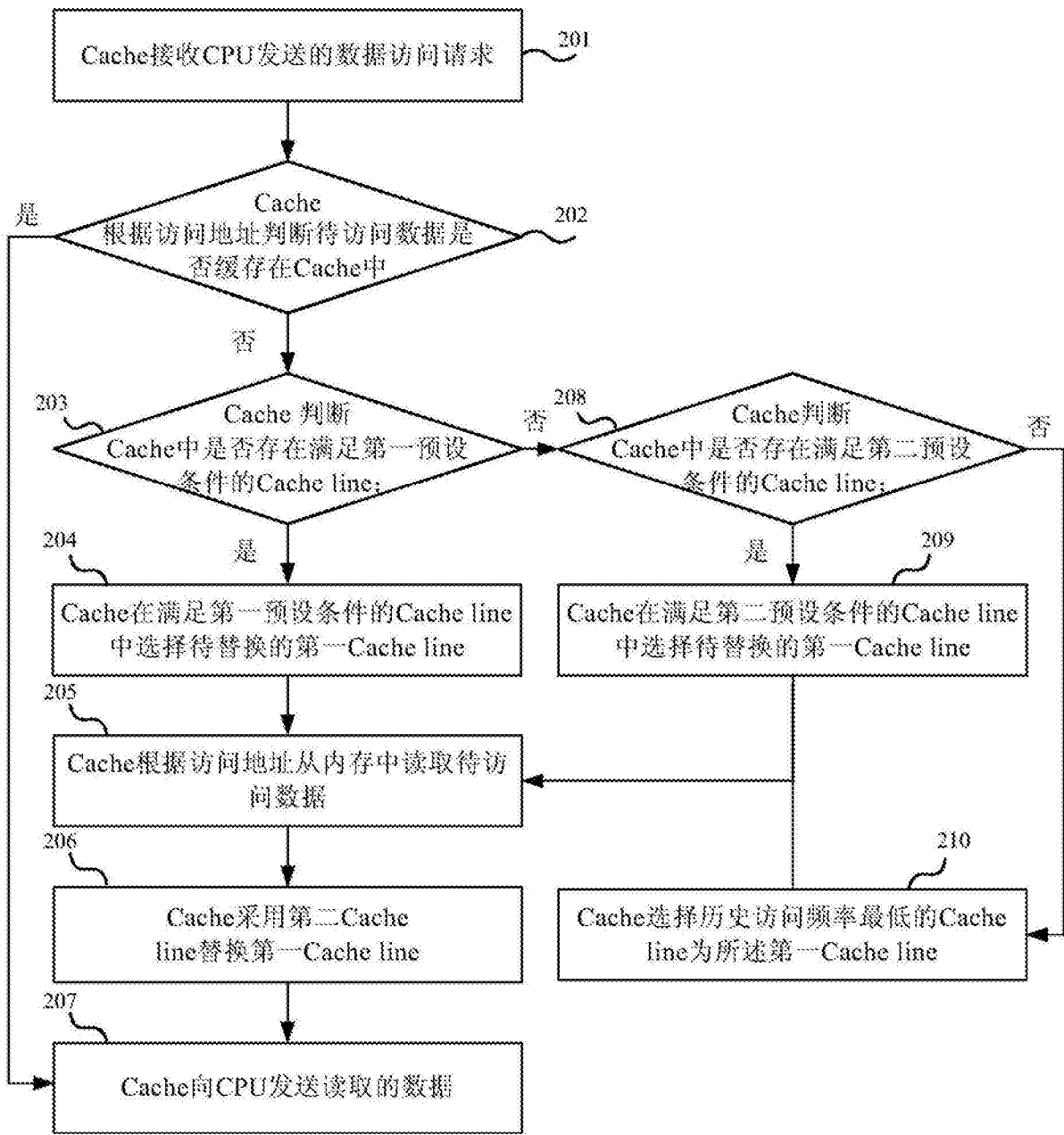


图 2

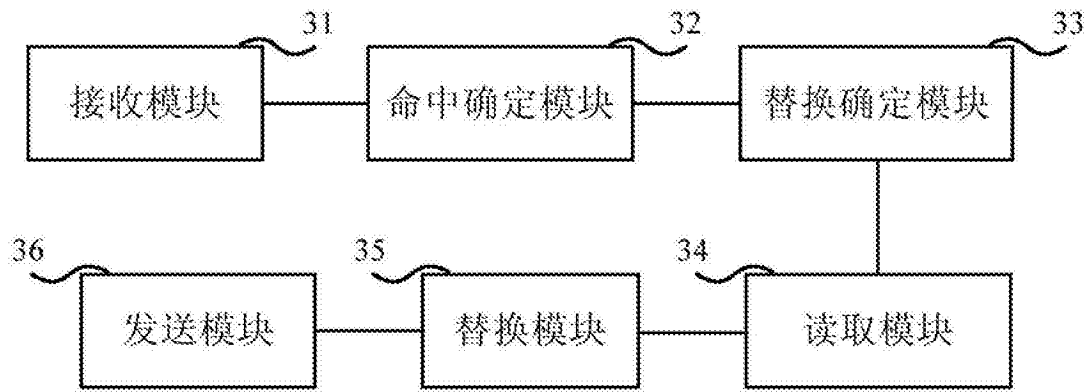


图 3