

(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
2016 年 10 月 20 日 (20.10.2016)



(10) 国际公布号
WO 2016/165542 A1

- (51) 国际专利分类号:
G06F 12/02 (2006.01)
- (21) 国际申请号: PCT/CN2016/077301
- (22) 国际申请日: 2016 年 3 月 25 日 (25.03.2016)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201510178548.X 2015 年 4 月 15 日 (15.04.2015) CN
- (71) 申请人: 阿里巴巴集团控股有限公司 (ALIBABA GROUP HOLDING LIMITED) [—/CN]; 英属开曼群岛大开曼乔治城资本大厦一座四层 847 号邮箱, Grand Cayman (KY)。
- (72) 发明人: 及
- (71) 申请人 (仅对美国): 杨成虎 (YANG, Chenghu) [CN/CN]; 中国浙江省杭州市余杭区文一西路 969 号 3 号楼 5 楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。
- (74) 代理人: 北京国昊天诚知识产权代理有限公司 (CO-HORIZON INTELLECTUAL PROPERTY INC.); 中国北京市朝阳区小关北里甲 2 号渔阳置业大厦 B 座 605, Beijing 100029 (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, 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, ST, 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)。

本国际公布:

- 包括国际检索报告(条约第 21 条(3))。

(54) Title: METHOD FOR ANALYZING CACHE HIT RATE, AND DEVICE

(54) 发明名称: 缓存命中率分析的方法及设备

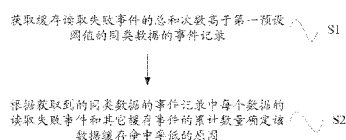


图 1

(57) Abstract: The present application provides a method for analyzing a cache hit rate, and a device. In the present application, data of a low cache hit rate is obtained according to the total number of times of data cache read failure events, and a cause of the low cache hit rate can be effectively determined according the total quantity of the data read failure events for the low cache hit rate and the other cache events, thereby improving the quality of cache codes according to the cause, and improving the performance of the whole system.

(57) 摘要: 本申请提供一种缓存命中率分析的方法及设备, 本申请根据某类数据缓存读取失败事件的总和次数得到存在缓存命中率低的数据, 根据存在缓存命中率低的数据的读取失败事件和其它缓存事件的累计数量, 可以有效确定缓存命中率低的原因, 有助于后续根据该原因改善缓存代码的质量, 提升整体系统性能。

WO 2016/165542 A1

缓存命中率分析的方法及设备

技术领域

5 本申请涉及通信及计算机领域，尤其涉及一种缓存命中率分析的方法及设备。

背景技术

缓存 (cache) 是一种临时存储数据的系统，缓存与数据库 (DB)、文件
10 系统等存储系统的区别在于，容量有限，价格贵，掉电即丢失，但访问速度快，写入速度快。缓存常常作为系统的数据缓冲，将频繁访问的数据放入缓存中，避免多次到数据库，或者文件系统的施行慢访问。在缓存系统中的一个核心指标为命中率 (hit rate)，命中率等于读取请求返回存在的次数/总共的读取次数。其中，命中 (hit) 是指读取某一数据时，假若该数据存在，
15 则是命中；非命中 (miss) 是指读取某一数据时，假若该数据不存在，则是非命中。

缓存命中率的高低与用户的使用逻辑关系密切。当命中率低时，是不能提升系统的性能的，所以命中率的高低直接影响系统的响应时间和整体吞吐量。而关于如何提升命中率，目前没有有效的量化方式，现有的判断方式，
20 基本是基于开发人员对代码的理解，业务结构的熟悉成度，人为判断命中率低的原因，再进行试验验证，只能通过人为经验判断，没有可依赖的工具。

发明内容

本申请的目的是提供一种缓存命中率分析的方法及设备，能够得到精确
25 的缓存命中率低的原因。

有鉴于此，本申请提供一种缓存命中率分析的方法，其中，包括：

获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录；

根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因。

5 进一步的，所述其它缓存事件包括删除事件、写入事件和淘汰事件中的一种或任意组合。

进一步的，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的删除事件和读取失败事件的累计数量均大于第二预设
10 阈值，且删除事件的累计数量与读取失败事件的累计数量的差值小于第三预设阈值，则确定该数据缓存命中率低的原因因为数据被主动删除。

进一步的，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的读取失败事件的累计数量大于第四预设阈值，且读取失
15 败事件的累计数量与写入事件的累计数量的差值大于第五预设阈值，则确定该数据缓存命中率低的原因因为访问的数据不存在。

进一步的，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的淘汰事件和读取失败事件的累计数量均大于第六预设
20 阈值，且淘汰事件的累计数量和读取失败事件的累计数量的差值小于第七预设阈值，则确定该数据缓存命中率低的原因因为缓存容量不足。

进一步的，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的写入事件和读取失败事件的累计数量均小于第八预设阈
25 值，且写入事件的累计数量与读取失败事件的累计数量差值小于第九预设阈值，则确定该数据缓存命中率低的原因因为访问的数据不具有热点性。

进一步的，获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录，包括：

获取每个数据的缓存读取失败事件和其它缓存事件的累计数量；

从所有数据中筛选出读取失败事件的累计数量大于第十预设阈值的

5 数据；

按预设分类标准将筛选出的每个数据分到对应的类；

计算每一同类数据的缓存读取失败事件的总和次数；

获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录，其中，所述同类数据的事件记录包括该类数据中每个数据的读
10 取失败事件和其它缓存事件的累计数量。

进一步的，获取每个数据的缓存读取失败事件和其它缓存事件的累计数量，包括：

通过回调接口获取每个数据的缓存读取失败事件或其它缓存事件，所述回调接口设置于缓存代码的事件路径上；

15 计算每个数据的缓存读取失败事件和其它缓存事件的累计数量。

进一步的，所述预设分类标准包括按数据的功能或按数据的唯一标识进行分类。

进一步的，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

20 将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表；

根据所述图表确定缓存命中率低的原因。

根据本申请的另一方面还提供一种用于缓存命中率分析的设备，包括：

第一装置，用于获取缓存读取失败事件的总和次数高于第一预设阈值的

25 的同类数据的事件记录；

第二装置，用于根据获取到的同类数据的事件记录中每个数据的读取

失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因。

进一步的，所述其它缓存事件包括删除事件、写入事件和淘汰事件中的一种或任意组合。

进一步的，所述第二装置，用于当某个数据的删除事件和读取失败事件的累计数量均大于第二预设阈值，且删除事件的累计数量与读取失败事件的累计数量的差值小于第三预设阈值，则确定该数据缓存命中率低的原因因为数据被主动删除。

进一步的，所述第二装置，用于当某个数据的读取失败事件的累计数量大于第四预设阈值，且读取失败事件的累计数量与写入事件的累计数量的差值大于第五预设阈值，则确定该数据缓存命中率低的原因因为访问的数据不存在。

进一步的，所述第二装置，用于当某个数据的淘汰事件和读取失败事件的累计数量均大于第六预设阈值，且淘汰事件的累计数量和读取失败事件的累计数量的差值小于第七预设阈值，则确定该数据缓存命中率低的原因因为缓存容量不足。

进一步的，所述第二装置，用于当某个数据的写入事件和读取失败事件的累计数量均小于第八预设阈值，且写入事件的累计数量与读取失败事件的累计数量差值小于第九预设阈值，则确定该数据缓存命中率低的原因因为访问的数据不具有热点性。

进一步的，所述第一装置包括：

第一一模块，用于获取每个数据的缓存读取失败事件和其它缓存事件的累计数量；

第一二模块，用于从所有数据中筛选出读取失败事件的累计数量大于第十预设阈值的数据；

第一三模块，用于按预设分类标准将筛选出的每个数据分到对应的类；

第一四模块，用于计算每一同类数据的缓存读取失败事件的总和次数；

第一五模块，用于获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录，其中，所述同类数据的事件记录包括该类数据中每个数据的读取失败事件和其它缓存事件的累计数量。

进一步的，所述第一一模块包括：

5 第一一一单元，用于通过回调接口获取每个数据的缓存读取失败事件或其它缓存事件，所述回调接口设置于缓存代码的事件路径上；

第一一二单元，用于计算每个数据的缓存读取失败事件和其它缓存事件的累计数量。

进一步的，所述预设分类标准包括按数据的功能或按数据的唯一标识
10 进行分类。

进一步的，所述第二装置，用于将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表，根据所述图表确定缓存命中率低的原因。

与现有技术相比，本申请根据某类数据缓存读取失败事件的总和次数
15 得到存在缓存命中率低的数据，根据存在缓存命中率低的数据的读取失败事件和其它缓存事件的累计数量，可以有效确定缓存命中率低的原因，有助于后续根据该原因改善缓存代码的质量，提升整体系统性能。

进一步的，本申请根据可每个数据的读取失败事件（GetMiss）的累计数量与删除事件（DeleteExist）、写入事件（SetNotExist）和淘汰事件（Evict）
20 的累计数量中至少一项的组合即可有效地确定该数据缓存命中率低的原因。

进一步的，本申请通过回调接口能够精确地监听和记录所有数据的事件，后续可根据记录的数据的事件汇总得到每个数据的缓存读取失败事件和其它缓存事件的累计数量。

25 进一步的，本申请将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表，可根据生成的图表

更直观地得到命中率低的原因。

附图说明

通过阅读参照以下附图所作的对非限制性实施例所作的详细描述，本
5 申请的其它特征、目的和优点将会变得更明显：

图 1 示出根据本申请一个方面的缓存命中率分析的方法的流程图；

图 2 示出本申请一优选实施例的缓存命中率分析的方法的流程图；

图 3 示出本申请另一优选实施例的缓存命中率分析的方法的流程图；

图 4 示出本申请再一优选实施例的缓存命中率分析的方法的流程图；

10 图 5 示出本申请一实施例的所有数据读取失败事件和其它缓存事件的
累计数量的有色图；

图 6 示出根据本申请另一个方面的用于缓存命中率分析的设备结构
图；

图 7 示出根据本申请一优选实施例的用于缓存命中率分析的设备结构
15 图；

图 8 示出根据本申请另一优选实施例的用于缓存命中率分析的设备结
构图；

附图中相同或相似的附图标记代表相同或相似的部件。

20 具体实施方式

在本申请一个典型的配置中，终端、服务网络的设备和可信方均包括
一个或多个处理器（CPU）、输入/输出接口、网络接口和内存。

内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器
（RAM）和/或非易失性内存等形式，如只读存储器（ROM）或闪存（flash
25 RAM）。内存是计算机可读介质的示例。

计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以

由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存 (PRAM)、静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、其他类型的随机存取存储器 (RAM)、只读存储器 (ROM)、电可擦除可编程只读存储器 (EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器 (CD-ROM)、数字多功能光盘 (DVD) 或其他光学存储、磁盒式磁带，磁带磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括非暂存电脑可读媒体 (transitory media)，如调制的数据信号和载波。

如图 1 所示，本申请提供一种缓存命中率分析的方法，包括：

步骤 S1，获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录；其中，读取失败事件 (GetMiss) 由用户主动发起的 Get 请求产生，并且读取结果为不存在，在此，如果缓存读取失败事件的总和次数高于第一预设阈值，则可确定该类数据的缓存命中率低，进而可以在后续步骤中判断该类数据的缓存命中率低的原因；

步骤 S2，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因。在此，某类数据的缓存读取失败事件的总和次数等于该类数据中所有数据的缓存读取失败事件的累计数量的相加之和，影响缓存命中率低的原因基本有如下几个方面的因素：

1. 缓存容量不足：当容量不足时缓存系统会产生淘汰，导致数据在缓存中的概率降低。虽然缓存系统具有 LRU 算法尽量保证淘汰的数据为非热点数据，但无法从本质上确保命中率的稳定。在此，淘汰 (evict) 是指一般缓存系统容量有限，当存储容量不够时，缓存系统会发生淘汰，也就是删除一些数据，以释放空间，删除数据的策略一般采用 LRU (Least Recently Used) 算

法。其中，LRU 算法既近期最少使用算法，其是一种数据结构算法，可以找出最近最少使用的元素，一般用于 Cache 淘汰策略里，也就是说，将最近最少使用的元素删除，达到释放空间的目的。

2.数据被主动删除：按照缓存系统的典型使用场景，数据发生修改或者删除时，需要将缓存内对应的数据删除，下一次访问时必然发生不命中。

3.访问的数据不存在：整个业务系统中不具有该数据存在，必然不命中。其中每个数据可以根据对应的 Key 查找，Key 是缓存中数据的唯一索引，同一缓存系统中不会重复。

4.访问的数据不具有热点性：根据缓存系统的使用方式，第一次访问发现数据不命中时，需要从后端数据源读取数据写入缓存系统中，第二次访问则出现命中，假设极端情况下，每个数据只会被访问一次，那命中率理论数值则为 0。本申请根据某类数据缓存读取失败事件的总和次数得到存在缓存命中率低的数据，根据存在缓存命中率低的数据的读取失败事件和其它缓存事件的累计数量，可以有效确定缓存命中率低的原因，有助于后续根据该原因改善缓存代码的质量，提升整体系统性能。

本申请的缓存命中率分析的方法一优选的实施例，所述其它缓存事件包括删除事件 (DeleteExist)、写入事件 (SetNotExist) 和淘汰事件 (Evict) 中的一种或任意组合。其中，删除事件 (DeleteExist) 由用户主动发起的 Delete 请求产生，删除事件中所指定的数据在删除之前是存在的；写入事件 (SetNotExist) 由用户主动发起的 Set 请求产生，写入事件中所指定的数据在删除之前不存在；淘汰事件 (Evict) 是指当缓存容量满时，发生淘汰动作。根据每个数据的读取失败事件 (GetMiss) 的累计数量与删除事件 (DeleteExist)、写入事件 (SetNotExist) 和淘汰事件 (Evict) 的累计数量中至少一项的组合即可有效地确定该数据缓存命中率低的原因。本领域技术人员应能理解上述其它缓存事件的描述仅为举例，其他现有的或今后可能出现的其它缓存事件的描述如可适用于本申请，也应包含在本申请保护

范围以内，并在此以引用方式包含于此。

本申请的缓存命中率分析的方法一优选的实施例中，步骤 S2，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

- 5 当某个数据的删除事件的累计数量（deleteExistCount）和读取失败事件（getMissCount）的累计数量均大于第二预设阈值，且删除事件的累计数量的累计数量（deleteExistCount）与读取失败事件的累计数量（getMissCount）的差值小于第三预设阈值，则确定该数据缓存命中率低的原因
- 10 为数据被主动删除。例如，如表 1 所示，对于 Key 为 MainPage 的数据记录，根据 MainPage 的名称可以确定该 Key 内容是首页数据，该数据的同类数据只有一个，即其自身，但其单个的读取失败事件的累计数量（getMissCount）比较高为 30，说明其是命中率低的数据，再根据其删除事件的累计数量（deleteExistCount）也比较高为 30，说明此 Key 对应数据的修改过于频繁，造成了命中率低。本领域技术人员应能理解上述确定数据缓存
- 15 命中率低的原因的描述仅为举例，其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

-{ Key: MainPage, deleteExistCount: 30, evictCount: 0, getMissCount: 30, setNotExistCount: 30 }
-{ Key: item_09cdec2d, deleteExistCount: 0, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_122cf833, deleteExistCount: 1, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_14a3a03a, deleteExistCount: 0, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_15ecb7bd, deleteExistCount: 0, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_17085737, deleteExistCount: 0, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_1c902fdd, deleteExistCount: 2, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_1ef6c4bc, deleteExistCount: 2, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_1fd78837, deleteExistCount: 1, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }

-{ Key: item_1ff9b528, deleteExistCount: 0, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_31c3e0a6, deleteExistCount: 0, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_3752aa0a, deleteExistCount: 0, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_3c77cac2, deleteExistCount: 1, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_43eaf434, deleteExistCount: 0, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_4511044f, deleteExistCount: 0, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_4f8f3999, deleteExistCount: 0, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_52fa54d4, deleteExistCount: 2, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_55fd3295, deleteExistCount: 1, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_58ea547f, deleteExistCount: 1, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_5afffbfe, deleteExistCount: 2, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_67bf2f27, deleteExistCount: 0, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_69c42d6e, deleteExistCount: 0, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_6bb5a7da, deleteExistCount: 2, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_7466fb35, deleteExistCount: 2, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_7b59ef31, deleteExistCount: 2, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_7e26abc2, deleteExistCount: 1, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_82a5b1a8, deleteExistCount: 0, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_842c6a4a, deleteExistCount: 1, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_880242f6, deleteExistCount: 2, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_8d446f69, deleteExistCount: 0, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_8df56967, deleteExistCount: 0, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_8ebdbdfd, deleteExistCount: 0, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_91096a22, deleteExistCount: 2, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_992b919c, deleteExistCount: 0, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_a4f1bb1e, deleteExistCount: 1, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_aa8892f5, deleteExistCount: 0, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_ac4985e3, deleteExistCount: 0, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_b57baa59, deleteExistCount: 0, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_be813651, deleteExistCount: 2, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_c1edea98, deleteExistCount: 1, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_c254be5b, deleteExistCount: 0, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_c4385493, deleteExistCount: 2, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_c86d52f4, deleteExistCount: 0, evictCount: 12, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_cf726c32, deleteExistCount: 0, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_e8b98eab, deleteExistCount: 1, evictCount: 13, getMissCount: 8, setNotExistCount: 8 }

-{ Key: item_e92c7bfc, deleteExistCount: 1, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: item_eda48d55, deleteExistCount: 0, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_f05b915a, deleteExistCount: 0, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_f220ecd9, deleteExistCount: 1, evictCount: 11, getMissCount: 8, setNotExistCount: 8 }
-{ Key: item_f6087d9d, deleteExistCount: 2, evictCount: 14, getMissCount: 9, setNotExistCount: 9 }
-{ Key: misc_foot, setdeleteExistCount: 0, evictCount: 0, getMissCount: 20, NotExistCount: 0 }
-{ Key: nick_0282d81c, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_149a3e16, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_1c8768b5, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_280f3dd2, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_5bbecebd, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_640aaa5e, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_66608a9f, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_684b984a, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_6da33be3, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_71b6db52, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_72341f21, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_9a121e06, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_9aee2b6f, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_a592aedc, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_a8608f23, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_a91e57aa, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_a9e261db, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_b0b76c12, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_b19ae545, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_b44dede3, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_bd335833, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_d3273f07, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_dd25603e, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_eadeaead, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: nick_f21f0ce4, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: session_0197f1f9, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: session_0f10bac7, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: session_104ba3c7, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: session_146af489, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: session_1fb8dbd7, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }

-{ Key:session_2410cd60, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key:session_284dfa9a, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: session_3781cdf6, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key:session_50e0ba1e, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key:session_8ed5c998, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key:session_a3dc28b2, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key:session_b9a9e2d2, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key:session_c7936ceb, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key:session_ff83d040, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 1 }
-{ Key: token_3a492fbc, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_4f58ca30, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_58f50b6f, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_5e1cb9a6, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_6824fe45, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_84fd1778, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_ae211626, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_be5e931f, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_d4529ce6, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }
-{ Key: token_f6db4f04, deleteExistCount: 0, evictCount: 0, getMissCount: 1, setNotExistCount: 0 }

表 1

本申请的缓存命中率分析的方法一优选的实施例中，步骤 S2，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

- 5 当某个数据的读取失败事件的累计数量 (getMissCount) 大于第四预设阈值，且读取失败事件的累计数量 (getMissCount) 与写入事件的累计数量 (setNotExistCount) 的差值大于第五预设阈值，则确定该数据缓存命中率低的原因，例如，对于表 1 中 Key 为 misc_foot 的数据，或者一组以 Key 为 token_前缀的同类数据的事件记录，在此，Key 为 misc_foot
- 10 的数据的同类数据只有一个，即其自身，但其读取失败事件 (getMissCount) 的累计数量为 20，说明该数据的缓存命中率低，而 Key 为 token_前缀的同类数据中的每个数据的读取失败事件 (getMissCount) 的累计数量为 1，但以

Key 为 token_前缀的同类数据的读取失败事件 (getMissCount) 的累计数量的总和次数较高, 即该同类数据中的数据个数较多, 说明以 Key 为 token_前缀的同类数据缓存命中率低, 后续根据每条数据记录的详细信息发现写入事件 (SetNotExistCount) 数值为 0, 说明业务系统没有对 Key 为 misc_foot
5 的数据或 Key 为 token_前缀的数据做缓存操作, 造成命中率低。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅为举例, 其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请, 也应包含在本申请保护范围以内, 并在此以引用方式包含于此。

本申请的缓存命中率分析的方法一优选的实施例中, 步骤 S2, 根据获
10 取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因, 包括:

当某个数据的淘汰事件的累计数量 (evictCount) 和读取失败事件的累计数量 (getMissCount) 均大于第六预设阈值, 且淘汰事件的累计数量 (evictCount) 和读取失败事件的累计数量 (getMissCount) 的差值小于第七预设阈值, 则确定该数据缓存命中率低的原因因为缓存容量不足。例如,
15 表 1 中, 以 item 为前缀的同类数据的读取失败事件的累计数量 (getMissCount) 的总和次数比较高, 即该同类数据中的数据个数较多, 且根据其详细信息发现, 淘汰事件的累计数量 (evictCount) 的数值也偏高, 说明其缓存容量不足导致的淘汰, 从而引起命中率下降。本领域技术人员应
20 能理解上述确定数据缓存命中率低的原因的描述仅为举例, 其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请, 也应包含在本申请保护范围以内, 并在此以引用方式包含于此。

本申请的缓存命中率分析的方法一可选的实施例中, 所述第二阈值、第四阈值和第六阈值可以设置为相同的值。

25 本申请的缓存命中率分析的方法一优选的实施例中, 步骤 S2, 根据获

取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的写入事件的累计数量（setNotExistCount）和读取失败事件的累计数量（getMissCount）均小于第八预设阈值，且写入事件的累计数量（setNotExistCount）与读取失败事件的累计数量（getMissCount）差值小于第九预设阈值，则确定该数据缓存命中率低的原因因为访问的数据不具有热点性。例如，表 1 中以 session 为前缀的同类数据的读取失败事件（GetMissCount）的累计数量基本为 1 或者 2，但读取失败事件（GetMissCount）的累计数量的总和次数较大，即该同类数据中的数据个数较多，而每个数据记录的写入事件（SetNotExistCount）基本为 1，说明此类型的数据的热点性不高。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅为举例，其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

如图 2 所示，本申请的缓存命中率分析的方法一优选的实施例中，步骤 S1，获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录，包括：

步骤 S11，获取每个数据的缓存读取失败事件和其它缓存事件的累计数量；如图 3 所示，本申请的缓存命中率分析的方法一较佳的实施例中，步骤 S11，包括步骤 S111 和步骤 S112，其中，步骤 S111，通过回调接口获取每个数据的缓存读取失败事件或其它缓存事件，所述回调接口设置于缓存代码的事件路径上；在此，可以通过回调接口每个数据的缓存读取失败事件或其它缓存事件进行监听，具体可按照事件发生的先后顺序逐条记录下监听到的事件的信息，包括事件的种类是什么如读取失败事件或其它缓存事件，以及事件对应数据的 Key 是什么，记录方式可以采用文件追加写方式实现，可以在指定的一段时间后会生成一个事件记录文件，由于回调接口

设置于缓存代码的事件路径上，只要有事件发生，回调接口都能监听和记录该事件；步骤 S112，计算每个数据的缓存读取失败事件和其它缓存事件的累计数量。在此，可根据上述步骤 S111 生成的事件记录文件的内容汇总计算每个数据的缓存读取失败事件（GetMissCount）的累计数量和其它缓存事件的累计数量（EvictCount，DeleteExistCount，SetNotExistCount）。例如，可以按照顺序读取事件记录文件，并在内存（或者其他存储介质）中记录对应的 Key，并且累计此 Key 的事件记录次数，记录扫描完成后，会生成一个[Key: EvictCount，DeleteExistCount，GetMissCount，SetNotExistCount]的 Key 列表，本领域技术人员应能理解上述获取每个数据的缓存读取失败事件和其它缓存事件的累计数量的描述仅为举例，其他现有的或今后可能出现的获取每个数据的缓存读取失败事件和其它缓存事件的累计数量的描述如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此；

步骤 S12，从所有数据中筛选出读取失败事件的累计数量大于第十预设阈值的数据；由于所有数据生成的如上述的 Key 列表规模会比较庞大，不利于后续分析的效率，所以可按照每条数据的事件记录中读取失败事件的累计数量（GetMissCount）的大小，从大到小筛选前 N 条大于第十预设阈值的数据，其中，N 太小会影响数据准确度，N 太大会影响视图展现，所以可以根据数据的大小设置 N 为默认的值，如 100，也就是说选取最多 100 条数据的事件记录；

步骤 S13，按预设分类标准将筛选出的每个数据分到对应的类；

步骤 S14，计算每一同类数据的缓存读取失败事件的总和次数；

步骤 S15，获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录，其中，所述同类数据的事件记录包括该类数据中每个数据的读取失败事件和其它缓存事件的累计数量。在此，如果缓存读取失

败事件的总和次数高于第一预设阈值，则可确定该类数据的缓存命中率
低，进而可以在后续步骤中判断该类数据的缓存命中率低的原因。缓存读
取失败事件的总和次数高于第一预设阈值有三种情况，一种情况是，某类
数据中的数据的事件记录的条数很少，但每个数据的缓存读取失败事件的
5 累计数量很高，导致这些累计数量相加后的缓存读取失败事件的总和次数
高于第一预设阈值；第二种情况是，虽然某类数据中每个数据的缓存读取
失败事件的累计数量很小，但该类数据中的数据的事件记录的条数很多，
导致这些累计数量相加后的缓存读取失败事件的总和次数仍高于第一预
设阈值；第三种情况是，综合第一和第二种情况，既有单条数据的缓存读
10 取失败事件的累计数量很高的情形，又有多条数据的缓存读取失败事件的
累计数量相加这和很高的情形，最终也导致这些累计数量相加后的缓存读
取失败事件的总和次数仍高于第一预设阈值。本领域技术人员应能理解上
述获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事
件记录的描述仅为举例，其他现有的或今后可能出现的获取缓存读取失败事
15 件的总和次数高于第一预设阈值的同类数据的事件记录的描述如可适用于
本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

本申请的缓存命中率分析的方法一优选的实施例中，步骤 S13 中所述
预设分类标准包括按数据的功能或按数据的唯一标识（Key）进行分类。缓
存系统中的 Key 一般具有前缀一致性，就是说同一类型的功能会采用同一个
20 前缀表达，以便区分：例如，表 1 的 Key 中，带同一前缀的为同一类业务，
如 item、session、nick 等前缀就表示不同的类。本领域技术人员应能理解上
述预设分类标准的描述仅为举例，其他现有的或今后可能出现的预设分类标
准的描述如可适用于本申请，也应包含在本申请保护范围以内，并在此以引
用方式包含于此。

25 如图 4 所示，本申请的缓存命中率分析的方法一优选的实施例中，步

步骤 S2，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

步骤 S21，将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表，如有色图、曲线图或柱状图 5 图中的任一种；例如，如图 5 所示，当生成有色图时，可根据数据的 Key 的字典顺序将 Key 排列，每行 10 个，共 10 行，每个 Key 对应绘出一个色块，色块的背景颜色深度由该 Key 对应的数据的读取失败事件的累计数量（GetMissCount）的数值决定；GetMissCount 最大的数值定义为正红色（R255,G0,B0），GetMissCount 为 0 的颜色定义为正绿色（R0, G255, B0），10 正红色和正绿色之间的中间值为黄色，GetMissCount 越大颜色越偏红，公式为：

假设整个数列中最大的 GetMissCount 为 MAX

如果 $\text{GetMissCount} \geq \text{MAX} / 2$ ，那么

$\text{R}=255; \text{G}=255 * 2 * (1 - \text{GetMissCount} / \text{MAX}); \text{B}=0;$

15 如果 $\text{GetMissCount} < \text{MAX} / 2$ ，那么

$\text{R}=255 * (2 * \text{GetMissCount} / \text{MAX}); \text{G}=255; \text{B} = 0;$

另外，需要在每个色块上标注该色块对应的数据其它缓存事件的累计数量，标注方式可以是先在每个色块上隐藏其它缓存事件的累计数量，然后在选中该色块时显示隐藏的其它缓存事件的累计数量；

20 步骤 S22，根据所述图表确定缓存命中率低的原因。在此，可根据生成的图表更直观地得到命中率低的原因。具体的，缓存系统中的 Key 一般具有前缀一致性，就是说同一类型的功能会采用同一个前缀表达，以便区分：例如，session_312312， session_888544， item_box111， item_box222，因此，如图 5 所示，可按照 Key 的字典顺序将数据进行排列，将同一业务功能即同 25 一类的数据色块集中排放在一起。可根据 GetMissCount 越大，颜色越偏红的原则，或者同一类数据的色块的排列面积较大的原则，判断为命中率低的色

块，然后针对命中率低的色块，根据每个色块上标注该色块对应的数据其它缓存事件的累计数量，便可以轻松的找到命中率低的原因。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅为举例，其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

如图 6 所示，根据本申请的另一面本申请还提供一种用于缓存命中率分析的设备 100，其中，包括：

第一装置 1，用于获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录；其中，读取失败事件（GetMiss）由用户主动发起的 Get 请求产生，并且读取结果为不存在，在此，如果缓存读取失败事件的总和次数高于第一预设阈值，则可确定该类数据的缓存命中率低，进而可以在后续步骤中判断该类数据的缓存命中率低的原因；

第二装置 2，用于根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因。在此，某类数据的缓存读取失败事件的总和次数等于该类数据中所有数据的缓存读取失败事件的累计数量的相加之和，影响缓存命中率低的原因基本有如下几个方面的因素：

1.缓存容量不足：当容量不足时缓存系统会产生淘汰，导致数据在缓存中的概率降低。虽然缓存系统具有 LRU 算法尽量保证淘汰的数据为非热点数据，但无法从本质上确保命中率的稳定。在此，淘汰（evict）是指一般缓存系统容量有限，当存储容量不够时，缓存系统会发生淘汰，也就是删除一些数据，以释放空间，删除数据的策略一般采用 LRU（Least Recently Used）算法。其中，LRU 算法既近期最少使用算法，其是一种数据结构算法，可以找出最近最少使用的元素，一般用于 Cache 淘汰策略里，也就是说，将最近最少使用的元素删除，达到释放空间的目的。

2.数据被主动删除：按照缓存系统的典型使用场景，数据发生修改或者删

除时，需要将缓存内对应的数据删除，下一次访问时必然发生不命中。

3.访问的数据不存在：整个业务系统中不具有该数据存在，必然不命中。其中每个数据可以根据对应的 Key 查找，Key 是缓存中数据的唯一索引，同一缓存系统中不会重复。

- 5 4.访问的数据不具有热点性：根据缓存系统的使用方式，第一次访问发现数据不命中时，需要从后端数据源读取数据写入缓存系统中，第二次访问则出现命中，假设极端情况下，每个数据只会被访问一次，那命中率理论数值则为 0。本申请根据某类数据缓存读取失败事件的总和次数得到存在缓存命中率低的数据，根据存在缓存命中率低的数据的读取失败事件和其它缓存事件的累计数量，可以有效确定缓存命中率低的原因，有助于后续根据该原因改善缓存代码的质量，提升整体系统性能。
- 10

本申请的用于缓存命中率分析的设备一优选的实施例中，所述其它缓存事件包括删除事件（DeleteExist）、写入事件（SetNotExist）和淘汰事件（Evict）中的一种或任意组合。其中，删除事件（DeleteExist）由用户主动发起的 Delete 请求产生，删除事件中所指定的数据在删除之前是存在的；写入事件（SetNotExist）由用户主动发起的 Set 请求产生，写入事件中所指定的数据在删除之前不存在；淘汰事件（Evict）是指当缓存容量满时，发生淘汰动作。根据每个数据的读取失败事件（GetMiss）的累计数量与删除事件（DeleteExist）、写入事件（SetNotExist）和淘汰事件（Evict）的累计数量中至少一项的组合即可有效地确定该数据缓存命中率低的原因。本领域技术人员应能理解上述其它缓存事件的描述仅为举例，其他现有的或今后可能出现的其它缓存事件的描述如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

15

20

本申请的用于缓存命中率分析的设备一优选的实施例中，所述第二装置，用于当某个数据的删除事件的累计数量（deleteExistCount）和读取失败事件的累计数量（getMissCount）均大于第二预设阈值，且删除事件的累

25

计数量的累计数量 (deleteExistCount) 与读取失败事件的累计数量 (getMissCount) 的差值小于第三预设阈值, 则确定该数据缓存命中率低的原因数据被主动删除。例如, 如表 1 所示, 对于 Key 为 MainPage 的数据记录, 根据 MainPage 的名称可以确定该 Key 内容是首页数据, 该数据的同类数据只有一个, 即其自身, 但其单个的读取失败事件的累计数量 (getMissCount) 比较高为 30, 说明其是命中率低的数据, 再根据其删除事件的累计数量 (deleteExistCount) 也比较高为 30, 说明此 Key 对应数据的修改过于频繁, 造成了命中率低。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅为举例, 其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请, 也应包含在本申请保护范围以内, 并在此以引用方式包含于此。

本申请的用于缓存命中率分析的设备一优选的实施例中, 所述第二装置, 用于当某个数据的读取失败事件的累计数量 (getMissCount) 大于第四预设阈值, 且读取失败事件的累计数量 (getMissCount) 与写入事件的累计数量 (setNotExistCount) 的差值大于第五预设阈值, 则确定该数据缓存命中率低的原因访问的数据不存在。例如, 对于表 1 中 Key 为 misc_foot 的数据, 或者一组以 Key 为 token_前缀的同类数据的事件记录, 在此, Key 为 misc_foot 的数据的同类数据只有一个, 即其自身, 但其读取失败事件

(getMissCount) 的累计数量为 20, 说明该数据的缓存命中率低, 而 Key 为 token_前缀的同类数据中的每个数据的读取失败事件 (getMissCount) 的累计数量为 1, 但以 Key 为 token_前缀的同类数据的读取失败事件 (getMissCount) 的累计数量的总和次数较高, 即该同类数据中的数据个数较多, 说明以 Key 为 token_前缀的同类数据缓存命中率低, 后续根据每条数据记录的详细信息发现写入事件 (SetNotExistCount) 数值为 0, 说明业务系统没有对 Key 为 misc_foot 的数据或 Key 为 token_前缀的数据做缓存操作, 造成命中率低。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅为举

例,其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请,也应包含在本申请保护范围以内,并在此以引用方式包含于此。

本申请的用于缓存命中率分析的设备一优选的实施例中,所述第二装置,用于当某个数据的淘汰事件的累计数量 (evictCount) 和读取失败事件的累计数量 (getMissCount) 均大于第六预设阈值,且淘汰事件的累计数量 (evictCount) 和读取失败事件的累计数量 (getMissCount) 的差值小于第七预设阈值,则确定该数据缓存命中率低的原因因为缓存容量不足。例如,表 1 中,以 item 为前缀的同类数据的读取失败事件的累计数量

(getMissCount) 的总和次数比较高,即该同类数据中的数据个数较多,且根据其详细信息发现,淘汰事件的累计数量 (evictCount) 的数值也偏高,说明其缓存容量不足导致的淘汰,从而引起命中率下降。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅为举例,其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请,也应包含在本申请保护范围以内,并在此以引用方式包含于此。

本申请的用于缓存命中率分析的设备一可选的实施例中,所述第二阈值、第四阈值和第六阈值可以设置为相同的值。

本申请的用于缓存命中率分析的设备一优选的实施例中,所述第二装置,用于当某个数据的写入事件的累计数量 (setNotExistCount) 和读取失败事件的累计数量 (getMissCount) 均小于第八预设阈值,且写入事件的累计数量 (setNotExistCount) 与读取失败事件的累计数量 (getMissCount) 差值小于第九预设阈值,则确定该数据缓存命中率低的原因因为访问的数据不具有热点性。例如,表 1 中以 session 为前缀的同类数据的读取失败事件 (GetMissCount) 的累计数量基本为 1 或者 2,但读取失败事件 (GetMissCount) 的累计数量的总和次数较大,即该同类数据中的数据个数较多,而每个数据记录的写入事件 (SetNotExistCount) 基本为 1,说明

此类型的数据的热点性不高。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅为举例，其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

5 如图 7 所示，本申请的用于缓存命中率分析的设备一优选的实施例中，所述第一装置 1 包括：

 第一一模块 11，用于获取每个数据的缓存读取失败事件和其它缓存事件的累计数量；

 第一二模块 12，用于从所有数据中筛选出读取失败事件的累计数量大
10 于第十预设阈值的数据；由于所有数据生成的如上述的 Key 列表规模会比较庞大，不利于后续分析的效率，所以可按照每条数据的事件记录中读取失败事件的累计数量（GetMissCount）的大小，从大到小筛选前 N 条大于第十预设阈值的数据，其中，N 太小会影响数据准确度，N 太大会影响视图展现，所以可以根据数据的大小设置 N 为默认的值，如 100，也就是说选取最多 100
15 条数据的事件记录；

 第一三模块 13，用于按预设分类标准将筛选出的每个数据分到对应的类；

 第一四模块 14，用于计算每一同类数据的缓存读取失败事件的总和次数；

20 第一五模块 15，用于获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录，其中，所述同类数据的事件记录包括该类数据中每个数据的读取失败事件和其它缓存事件的累计数量。在此，如果缓存读取失败事件的总和次数高于第一预设阈值，则可确定该类数据的缓存命中率低，进而可以在后续步骤中判断该类数据的缓存命中率低的原因。
25 缓存读取失败事件的总和次数高于第一预设阈值有三种情况，一种情况是，某类数据中的数据的事件记录的条数很少，但每个数据的缓存读取失

败事件的累计数量很高，导致这些累计数量相加后的缓存读取失败事件的总和次数高于第一预设阈值；第二种情况是，虽然某类数据中每个数据的缓存读取失败事件的累计数量很小，但该类数据中的数据的事件记录的条数很多，导致这些累计数量相加后的缓存读取失败事件的总和次数仍高于第一预设阈值；第三种情况是，综合第一和第二种情况，既有单条数据的缓存读取失败事件的累计数量很高的情形，又有多条数据的缓存读取失败事件的累计数量相加这和很高的情形，最终也导致这些累计数量相加后的缓存读取失败事件的总和次数仍高于第一预设阈值。本领域技术人员应能理解上述获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录的描述仅为举例，其他现有的或今后可能出现的获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录的描述如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

如图 8 所示，本申请的用于缓存命中率分析的设备一优选的实施例中，所述第一一模块 11 包括：

第一一一单元 111，用于通过回调接口获取每个数据的缓存读取失败事件或其它缓存事件，所述回调接口设置于缓存代码的事件路径上；在此，可以通过回调接口每个数据的缓存读取失败事件或其它缓存事件进行监听，具体可按照事件发生的先后顺序逐条记录下监听到的事件的信息，包括事件的种类是什么如读取失败事件或其它缓存事件，以及事件对应数据的 Key 是什么，记录方式可以采用文件追加写方式实现，可以在指定的一段时间后会生成一个事件记录文件，由于回调接口设置于缓存代码的事件路径上，只要有事件发生，回调接口都能监听和记录该事件；

第一一二单元 112，用于计算每个数据的缓存读取失败事件和其它缓存事件的累计数量。在此，可根据上述第一一一单元 111 生成的事件记录文件的内容汇总计算每个数据的缓存读取失败事件（GetMissCount）的累计

数量和其它缓存事件的累计数量 (EvictCount, DeleteExistCount, SetNotExistCount)。例如, 可以按照顺序读取事件记录文件, 并在内存 (或者其他存储介质) 中记录对应的 Key, 并且累计此 Key 的事件记录次数, 记录扫描完成后, 会生成一个[Key: EvictCount, DeleteExistCount, GetMissCount, SetNotExistCount]的 Key 列表, 本领域技术人员应能理解上述获取每个数据的缓存读取失败事件和其它缓存事件的累计数量的描述仅为举例, 其他现有的或今后可能出现的获取每个数据的缓存读取失败事件和其它缓存事件的累计数量的描述如可适用于本申请, 也应包含在本申请保护范围以内, 并在此以引用方式包含于此

本申请的用于缓存命中率分析的设备一优选的实施例中, 所述预设分类标准包括按数据的功能或按数据的唯一标识 (Key) 进行分类。缓存系统中的 Key 一般具有前缀一致性, 就是说同一类型的功能会采用同一个前缀表达, 以便区分: 例如, 表 1 的 Key 中, 带同一前缀的为同一类业务, 如 item、session、nick 等前缀就表示不同的类。本领域技术人员应能理解上述预设分类标准的描述仅为举例, 其他现有的或今后可能出现的预设分类标准的描述如可适用于本申请, 也应包含在本申请保护范围以内, 并在此以引用方式包含于此。

本申请的用于缓存命中率分析的设备一优选的实施例中, 所述第二装置, 用于将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表, 根据所述图表确定缓存命中率低的原因。所图表如有色图、曲线图或柱状图中的任一种; 例如, 如图 5 所示, 当生成有色图时, 可根据数据的 Key 的字典顺序将 Key 排列, 每行 10 个, 共 10 行, 每个 Key 对应绘出一个色块, 色块的背景颜色深度由该 Key 对应的数据的读取失败事件的累计数量 (GetMissCount) 的数值决定; GetMissCount 最大的数值定义为正红色 (R255,G0,B0), GetMissCount 为 0

的颜色定义为正绿色（R0，G255，B0），正红色和正绿色之间的中间值为黄色，GetMissCount 越大颜色越偏红，公式为：

假设整个数列中最大的 GetMissCount 为 MAX

如果 $\text{GetMissCount} \geq \text{MAX} / 2$ ，那么

5 $R=255; G=255*2*(1-\text{GetMissCount}/\text{MAX}); B=0;$

如果 $\text{GetMissCount} < \text{MAX} / 2$ ，那么

$R=255 * (2*\text{GetMissCount}/\text{MAX}); G=255; B = 0;$

另外，需要在每个色块上标注该色块对应的数据其它缓存事件的累计数量，标注方式可以是先在每个色块上隐藏其它缓存事件的累计数量，然后
10 后在选中该色块时显示隐藏的其它缓存事件的累计数量；

后续可根据生成的图表更直观地得到命中率低的原因。具体的，缓存系统中的 Key 一般具有前缀一致性，就是说同一类型的功能会采用同一个前缀表达，以便区分：例如，session_312312， session_888544， item_box111， item_box222，因此，如图 5 所示，可按照 Key 的字典顺序将数据进行排列，
15 将同一业务功能即同一类的数据色块集中排放在一起。可根据 GetMissCount 越大，颜色越偏红的原则，或者同一类数据的色块的排列面积较大的原则，判断为命中率低的色块，然后针对命中率低的色块，根据每个色块上标注该色块对应的数据其它缓存事件的累计数量，便可以轻松的找到命中率低的原因。本领域技术人员应能理解上述确定数据缓存命中率低的原因的描述仅
20 为举例，其他现有的或今后可能出现的确定数据缓存命中率低的原因如可适用于本申请，也应包含在本申请保护范围以内，并在此以引用方式包含于此。

综上所述，本申请根据某类数据缓存读取失败事件的总和次数得到存在缓存命中率低的数据，根据存在缓存命中率低的数据的读取失败事件和其它缓存事件的累计数量，可以有效确定缓存命中率低的原因，有助于后续根据
25 该原因改善缓存代码的质量，提升整体系统性能。

进一步的，本申请根据可每个数据的读取失败事件（GetMiss）的累计

数量与删除事件(DeleteExist)、写入事件(SetNotExist)和淘汰事件(Evict)的累计数量中至少一项的组合即可有效地确定该数据缓存命中率低的原因。

进一步的,本申请通过回调接口能够精确地监听和记录所有数据的事件,后续可根据记录的数据的事件汇总得到每个数据的缓存读取失败事件和其它缓存事件的累计数量。

进一步的,本申请将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表,可根据生成的图表更直观地得到命中率低的原因。

显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围之内,则本申请也意图包含这些改动和变型在内。

需要注意的是,本申请可在软件和/或软件与硬件的组合体中被实施,例如,可采用专用集成电路(ASIC)、通用目的计算机或任何其他类似硬件设备来实现。在一个实施例中,本申请的软件程序可以通过处理器执行以实现上文所述步骤或功能。同样地,本申请的软件程序(包括相关的数据结构)可以被存储到计算机可读记录介质中,例如,RAM存储器,磁或光驱动器或软磁盘及类似设备。另外,本申请的一些步骤或功能可采用硬件来实现,例如,作为与处理器配合从而执行各个步骤或功能的电路。

另外,本申请的一部分可被应用为计算机程序产品,例如计算机程序指令,当其被计算机执行时,通过该计算机的操作,可以调用或提供根据本申请的方法和/或技术方案。而调用本申请的方法的程序指令,可能被存储在固定的或可移动的记录介质中,和/或通过广播或其他信号承载媒体中的数据流而被传输,和/或被存储在根据所述程序指令运行的计算机设备的工作存储器中。在此,根据本申请的一个实施例包括一个装置,该装置包

括用于存储计算机程序指令的存储器和用于执行程序指令的处理器，其中，当该计算机程序指令被该处理器执行时，触发该装置运行基于前述根据本申请的多个实施例的方法和/或技术方案。

5 对于本领域技术人员而言，显然本申请不限于上述示范性实施例的细节，而且在不背离本申请的精神或基本特征的情况下，能够以其他的具体形式实现本申请。因此，无论从哪一点来看，均应将实施例看作是示范性的，而且是非限制性的，本申请的范围由所附权利要求而不是上述说明限定，因此旨在将落在权利要求的等同要件的含义和范围内的所有变化涵括在本申请内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。
10 此外，显然“包括”一词不排除其他单元或步骤，单数不排除复数。装置权利要求中陈述的多个单元或装置也可以由一个单元或装置通过软件或者硬件来实现。第一，第二等词语用来表示名称，而并不表示任何特定的顺序。

权 利 要 求 书

1. 一种缓存命中率分析的方法，其中，包括：

获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录；

5 根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因。

2. 如权利要求 1 所述的方法，其中，所述其它缓存事件包括删除事件、写入事件和淘汰事件中的一种或任意组合。

3. 如权利要求 2 所述的方法，其中，根据获取到的同类数据的事件记
10 录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的删除事件和读取失败事件的累计数量均大于第二预设阈值，且删除事件的累计数量与读取失败事件的累计数量的差值小于第三预设阈值，则确定该数据缓存命中率低的原因因为数据被主动删除。

15 4. 如权利要求 2 或 3 所述的方法，其中，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的读取失败事件的累计数量大于第四预设阈值，且读取失败事件的累计数量与写入事件的累计数量的差值大于第五预设阈值，则确
20 定该数据缓存命中率低的原因因为访问的数据不存在。

5. 如权利要求 2 至 4 任一项所述的方法，其中，根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因，包括：

当某个数据的淘汰事件和读取失败事件的累计数量均大于第六预设
25 阈值，且淘汰事件的累计数量和读取失败事件的累计数量的差值小于第七预设阈值，则确定该数据缓存命中率低的原因因为缓存容量不足。

6. 如权利要求 2 至 5 任一项所述的方法, 其中, 根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因, 包括:

当某个数据的写入事件和读取失败事件的累计数量均小于第八预设阈值, 且写入事件的累计数量与读取失败事件的累计数量差值小于第九预设阈值, 则确定该数据缓存命中率低的原因因为访问的数据不具有热点性。

7. 如权利要求 1 至 6 任一项所述的方法, 其中, 获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录, 包括:

获取每个数据的缓存读取失败事件和其它缓存事件的累计数量;

从所有数据中筛选出读取失败事件的累计数量大于第十预设阈值的数据;

按预设分类标准将筛选出的每个数据分到对应的类;

计算每一同类数据的缓存读取失败事件的总和次数;

获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录, 其中, 所述同类数据的事件记录包括该类数据中每个数据的读取失败事件和其它缓存事件的累计数量。

8. 如权利要求 7 所述的方法, 其中, 获取每个数据的缓存读取失败事件和其它缓存事件的累计数量, 包括:

通过回调接口获取每个数据的缓存读取失败事件或其它缓存事件, 所述回调接口设置于缓存代码的事件路径上;

计算每个数据的缓存读取失败事件和其它缓存事件的累计数量。

9. 如权利要求 7 或 8 所述的方法, 其中, 所述预设分类标准包括按数据的功能或按数据的唯一标识进行分类。

10. 如权利要求 1 或 2 任一项所述的方法, 其中, 根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因, 包括:

将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表；

根据所述图表确定缓存命中率低的原因。

11. 一种用于缓存命中率分析的设备，其中，包括：

5 第一装置，用于获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录；

第二装置，用于根据获取到的同类数据的事件记录中每个数据的读取失败事件和其它缓存事件的累计数量确定该数据缓存命中率低的原因。

12. 如权利要求 11 所述的设备，其中，所述其它缓存事件包括删除事件、写入事件和淘汰事件中的一种或任意组合。

13. 如权利要求 12 所述的设备，其中，所述第二装置，用于当某个数据的删除事件和读取失败事件的累计数量均大于第二预设阈值，且删除事件的累计数量与读取失败事件的累计数量的差值小于第三预设阈值，则确定该数据缓存命中率低的原因因为数据被主动删除。

15 14. 如权利要求 12 或 13 所述的设备，其中，所述第二装置，用于当某个数据的读取失败事件的累计数量大于第四预设阈值，且读取失败事件的累计数量与写入事件的累计数量的差值大于第五预设阈值，则确定该数据缓存命中率低的原因因为访问的数据不存在。

20 15. 如权利要求 12 至 14 任一项所述的设备，其中，所述第二装置，用于当某个数据的淘汰事件和读取失败事件的累计数量均大于第六预设阈值，且淘汰事件的累计数量和读取失败事件的累计数量的差值小于第七预设阈值，则确定该数据缓存命中率低的原因因为缓存容量不足。

25 16. 如权利要求 12 至 15 任一项所述的设备，其中，所述第二装置，用于当某个数据的写入事件和读取失败事件的累计数量均小于第八预设阈值，且写入事件的累计数量与读取失败事件的累计数量差值小于第九预设阈值，则确定该数据缓存命中率低的原因因为访问的数据不具有热点性。

17. 如权利要求 11 至 16 任一项所述的设备, 其中, 所述第一装置包括:

第一一模块, 用于获取每个数据的缓存读取失败事件和其它缓存事件的累计数量;

5 第一二模块, 用于从所有数据中筛选出读取失败事件的累计数量大于第十预设阈值的数据;

第一三模块, 用于按预设分类标准将筛选出的每个数据分到对应的类;

10 第一四模块, 用于计算每一同类数据的缓存读取失败事件的总和次数;

第一五模块, 用于获取缓存读取失败事件的总和次数高于第一预设阈值的同类数据的事件记录, 其中, 所述同类数据的事件记录包括该类数据中每个数据的读取失败事件和其它缓存事件的累计数量。

18. 如权利要求 17 所述的设备, 其中, 所述第一一模块包括:

15 第一一一单元, 用于通过回调接口获取每个数据的缓存读取失败事件或其它缓存事件, 所述回调接口设置于缓存代码的事件路径上;

第一一二单元, 用于计算每个数据的缓存读取失败事件和其它缓存事件的累计数量。

20 19. 如权利要求 17 或 18 所述的设备, 其中, 所述预设分类标准包括按数据的功能或按数据的唯一标识进行分类。

20. 如权利要求 11 或 12 任一项所述的设备, 其中, 所述第二装置, 用于将获取到的同类数据的事件记录中所有数据按读取失败事件和其它缓存事件的累计数量的大小生成图表, 根据所述图表确定缓存命中率低的原因。

25

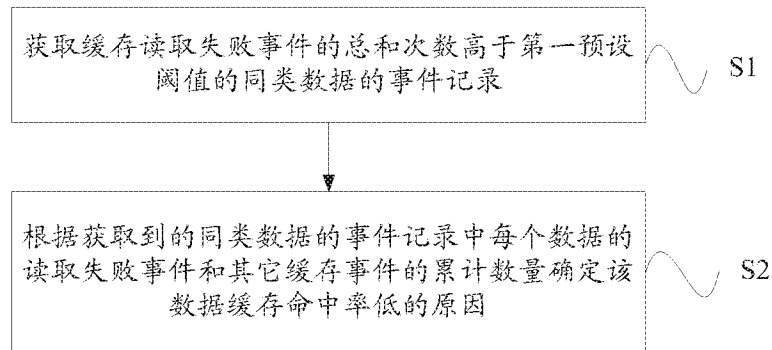


图 1

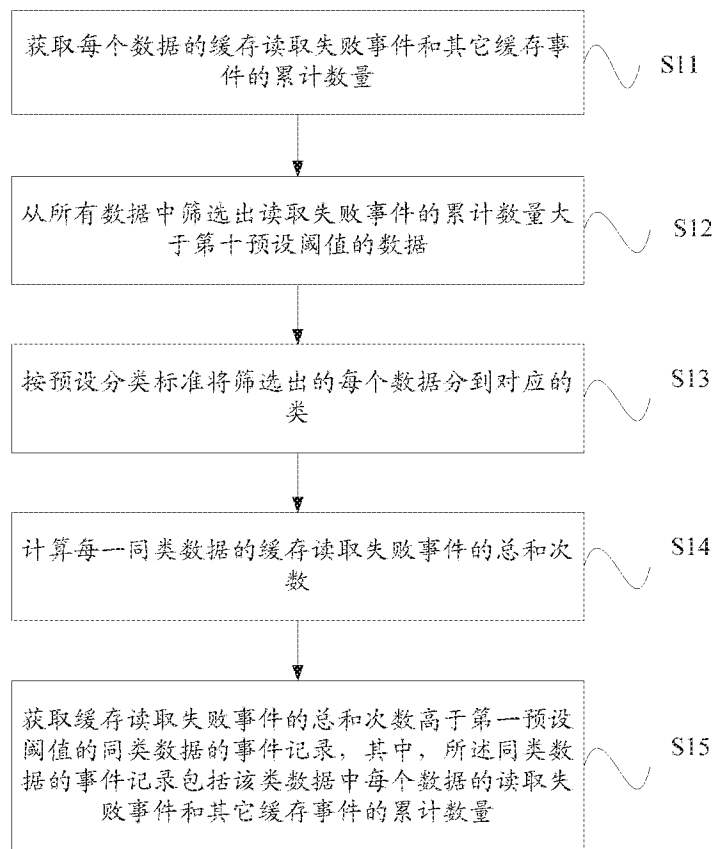


图 2

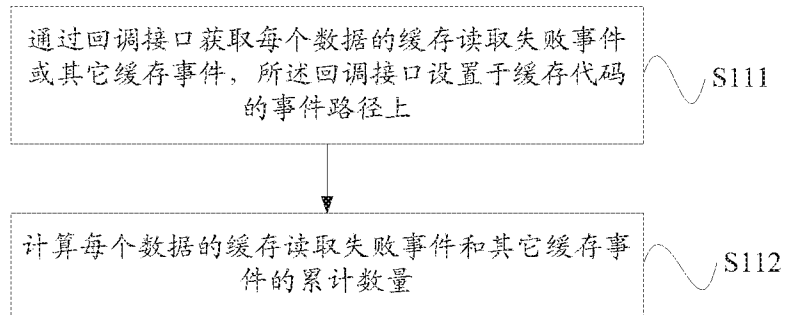


图 3

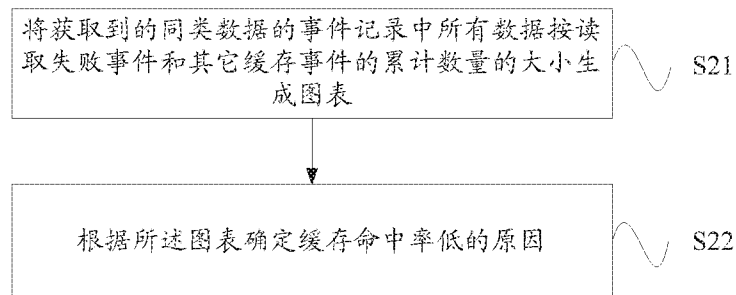


图 4

[illegible]

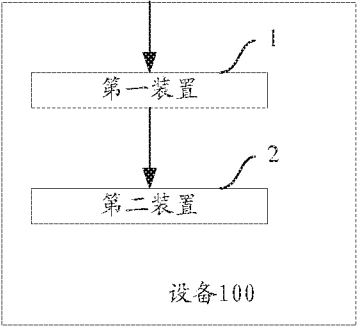


图 6

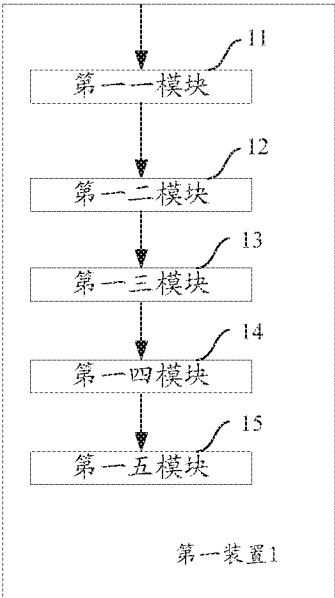


图 7

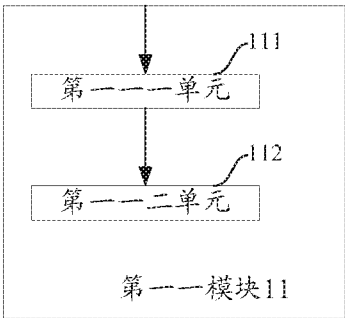


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2016/077301**A. CLASSIFICATION OF SUBJECT MATTER**

G06F 12/02 (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)

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)

WPI; EPODOC; CPRS; CNKI: cache, failure, miss, missing, error, reason, cause, threshold, event, statisti +, count, causality, total

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 1928841 A (PANASONIC CORPORATION), 14 March 2007 (14.03.2007), description, page 7, line 16 to page 18, line 18, and figures 1-12B	1, 2, 10, 11, 12, 20
A	CN 101753344 A (HUAWEI TECHNOLOGIES CO., LTD.), 23 June 2010 (23.06.2010), see the whole document	1-20
A	CN 101604287 A (INSPUR ELECTRONIC INFORMATION INDUSTRY CO., LTD.), 16 December 2009 (16.12.2009), see the whole document	1-20
A	US 5930507 A (FUJITSU LTD.), 27 July 1999 (27.07.1999), see the whole document	1-20

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

* Special categories of cited documents:	“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
“A” document defining the general state of the art which is not considered to be of particular relevance	“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
“E” earlier application or patent but published on or after the international filing date	“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
“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)	“&” document member of the same patent family
“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	

Date of the actual completion of the international search
27 May 2016 (27.05.2016)Date of mailing of the international search report
06 June 2016 (06.06.2016)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

WEN, RuiTelephone No.: (86-10) **62089524**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2016/077301

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 1928841 A	14 March 2007	US 7590792 B2	15 September 2009
		CN 1928841 B	13 October 2010
		JP 2007072883 A	22 March 2007
		US 2007055809 A1	08 March 2007
		JP 4827469 B2	30 November 2011
CN 101753344 A	23 June 2010	CN 101753344 B	05 September 2012
CN 101604287 A	16 December 2009	None	
US 5930507 A	27 July 1999	JP H08328870 A	13 December 1996

A. 主题的分类 G06F 12/02 (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)) WPI; EPDOC; CPRS; CNKI: cache, failure, miss, missing, error, reason, cause, threshold, event, statistical, count, 缓存, 高速缓冲存储, 失败, 未命中, 未中, 原因, 理由, 因由, 原由, 因果, 事件, 统计, 总计																	
C. 相关文件 <table border="1"> <thead> <tr> <th>类 型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>X</td> <td>CN 1928841 A (松下电器产业株式会社) 2007年 3月 14日 (2007 - 03 - 14) 说明书第7页第16行--第18页第18行, 图1-12B</td> <td>1, 2, 10, 11, 12, 20</td> </tr> <tr> <td>A</td> <td>CN 101753344 A (华为技术有限公司) 2010年 6月 23日 (2010 - 06 - 23) 参见全文</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>CN 101604287 A (浪潮电子信息产业股份有限公司) 2009年 12月 16日 (2009 - 12 - 16) 参见全文</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>US 5930507 A (FUJITSU LTD) 1999年 7月 27日 (1999 - 07 - 27) 参见全文</td> <td>1-20</td> </tr> </tbody> </table>			类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求	X	CN 1928841 A (松下电器产业株式会社) 2007年 3月 14日 (2007 - 03 - 14) 说明书第7页第16行--第18页第18行, 图1-12B	1, 2, 10, 11, 12, 20	A	CN 101753344 A (华为技术有限公司) 2010年 6月 23日 (2010 - 06 - 23) 参见全文	1-20	A	CN 101604287 A (浪潮电子信息产业股份有限公司) 2009年 12月 16日 (2009 - 12 - 16) 参见全文	1-20	A	US 5930507 A (FUJITSU LTD) 1999年 7月 27日 (1999 - 07 - 27) 参见全文	1-20
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
X	CN 1928841 A (松下电器产业株式会社) 2007年 3月 14日 (2007 - 03 - 14) 说明书第7页第16行--第18页第18行, 图1-12B	1, 2, 10, 11, 12, 20															
A	CN 101753344 A (华为技术有限公司) 2010年 6月 23日 (2010 - 06 - 23) 参见全文	1-20															
A	CN 101604287 A (浪潮电子信息产业股份有限公司) 2009年 12月 16日 (2009 - 12 - 16) 参见全文	1-20															
A	US 5930507 A (FUJITSU LTD) 1999年 7月 27日 (1999 - 07 - 27) 参见全文	1-20															
☐ 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 2016年 5月 27日		国际检索报告邮寄日期 2016年 6月 6日															
ISA/CN的名称和邮寄地址 中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10) 62019451		授权官员 温睿 电话号码 (86-10) 62089524															

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2016/077301

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)
CN	1928841	A	2007年 3月 14日	US	7590792	B2	2009年 9月 15日
				CN	1928841	B	2010年 10月 13日
				JP	2007072883	A	2007年 3月 22日
				US	2007055809	A1	2007年 3月 8日
				JP	4827469	B2	2011年 11月 30日
CN	101753344	A	2010年 6月 23日	CN	101753344	B	2012年 9月 5日
CN	101604287	A	2009年 12月 16日	无			
US	5930507	A	1999年 7月 27日	JP	H08328870	A	1996年 12月 13日