



(12) 发明专利

(10) 授权公告号 CN 102662713 B

(45) 授权公告日 2014. 04. 16

(21) 申请号 201210106766. 9

CN 102360345 A, 2012. 02. 22, 说明书  
[0011] 至 [0014].

(22) 申请日 2012. 04. 12

CN 101604283 A, 2009. 12. 16, 全文.

(73) 专利权人 腾讯科技(深圳)有限公司

地址 518000 广东省深圳市福田区赛格科技园 2 栋东 403 室

审查员 康健

(72) 发明人 林道正 聂科峰 方斌 李世平

(74) 专利代理机构 北京三高永信知识产权代理  
有限责任公司 11138

代理人 罗振安

(51) Int. Cl.

G06F 9/445(2006. 01)

(56) 对比文件

CN 101008922 A, 2007. 08. 01, 说明书第 3 页  
最后 3 段.

US 20080156355 A1, 2008. 05. 30, 全文.

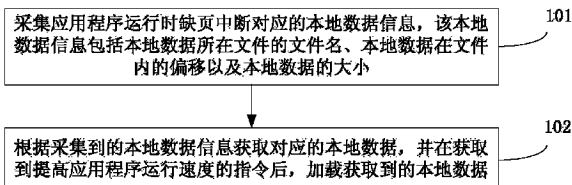
权利要求书2页 说明书11页 附图2页

(54) 发明名称

提高应用程序运行速度的方法、装置及终端

(57) 摘要

本发明公开了一种提高应用程序运行速度的方法、装置及终端,属于计算机领域。方法包括:采集应用程序运行时缺页中断对应的本地数据信息,本地数据信息包括本地数据所在文件的文件名、本地数据在文件内的偏移以及本地数据的大小;根据采集到的本地数据信息获取对应的本地数据,并在获取到提高应用程序运行速度的指令后,加载获取到的本地数据。本发明通过加载根据应用程序运行时缺页中断对应的本地数据信息获取到的本地数据,以减少应用程序访问 I/O 操作的耗时,提高应用程序运行速度;由于缺页中断对应的本地数据信息不局限于访问场景,可增强速度提高的效果;针对不同应用程序具有不同的速度优化程度,提高速度优化的针对性,扩大速度优化的空间。



1. 一种提高应用程序运行速度的方法,其特征在于,所述方法包括:

采集应用程序运行时缺页中断对应的本地数据信息,所述本地数据信息包括本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;

根据采集到的所述本地数据信息获取对应的本地数据,并在获取到提高所述应用程序运行速度的指令后,加载获取到的所述本地数据;

其中,所述采集应用程序运行时缺页中断对应的本地数据信息,具体包括:

通过事件跟踪机制采集应用程序运行时的硬缺页事件和软缺页事件,从记录了所述硬缺页事件和软缺页事件的事件跟踪日志中获取所述应用程序运行时缺页中断对应的本地数据信息;

或者,所述采集应用程序运行时缺页中断对应的本地数据信息,具体包括:

通过钩子 HOOK 技术对系统底层调用进行监控,并获取用于记录缺页中断对应的本地数据信息的监控代码,根据所述监控代码获取所述应用程序运行时缺页中断对应的本地数据信息。

2. 根据权利要求 1 所述的方法,其特征在于,所述根据采集到的所述本地数据信息获取对应的本地数据,具体包括:

解析采集到的所述本地数据信息,得到本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;

按照所述本地数据所在文件的文件名将采集到的本地数据信息进行聚合;

按照所述本地数据在所述文件内的偏移将聚合后的本地数据信息进行合并,并在合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息;

根据选取的所述预取数据信息获取对应的本地数据。

3. 根据权利要求 2 所述的方法,其特征在于,所述在合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息之后,还包括:

根据所述应用程序的信息存储选取的所述预取数据信息;

所述加载获取到的本地数据,具体包括:

根据所述应用程序的信息获取所述预取数据信息,并根据所述预取数据信息获取对应的本地数据,将获取到的本地数据加载至内存。

4. 根据权利要求 1 至 3 中任一权利要求所述的方法,其特征在于,所述加载获取到的本地数据之前,还包括:

输出应用程序列表,提示用户在所述应用程序列表中选择提高运行速度的应用程序;

根据用户在所述应用程序列表中选择的应用程序接收提高所述应用程序运行速度的指令以及所述应用程序的信息。

5. 一种提高应用程序运行速度的装置,其特征在于,所述装置包括:

采集模块,用于采集应用程序运行时缺页中断对应的本地数据信息,所述本地数据信息包括本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;

获取模块,用于根据所述采集模块采集到的所述本地数据信息获取对应的本地数据;

加载模块,用于获取到提高所述应用程序运行速度的指令后,加载所述获取模块获取到的所述本地数据;

其中,所述采集模块,具体用于通过事件跟踪机制采集应用程序运行时的硬缺页事件和软缺页事件,从记录了所述硬缺页事件和软缺页事件的事件跟踪日志中获取所述应用程序运行时缺页中断对应的本地数据信息;

或者,所述采集模块,具体用于通过钩子 HOOK 技术对系统底层调用进行监控,并获取用于记录缺页中断对应的本地数据信息的监控代码,根据所述监控代码获取所述应用程序运行时缺页中断对应的本地数据信息。

6. 根据权利要求 5 所述的装置,其特征在于,所述获取模块,具体包括:

解析单元,用于解析所述采集模块采集到的本地数据信息,得到本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;

聚合单元,用于按照所述解析单元解析出的本地数据所在文件的文件名将采集到的本地数据信息进行聚合;

合并单元,用于按照所述解析单元解析出的本地数据在所述文件内的偏移将所述聚合单元聚合后的本地数据信息进行合并;

选取单元,用于在所述合并单元合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息;

获取单元,用于根据所述选取单元选取的所述预取数据信息获取对应的本地数据。

7. 根据权利要求 6 所述的装置,其特征在于,所述装置还包括:

存储模块,用于根据所述应用程序的信息存储所述选取单元选取的所述预取数据信息;

所述加载模块,具体包括:

第一获取单元,用于根据所述应用程序的信息获取所述存储模块存储的预取数据信息;

第二获取单元,用于根据所述第一获取单元获取到的预取数据信息获取对应的本地数据;

加载单元,用于将所述第二获取单元获取到的本地数据加载至内存。

8. 根据权利要求 5 至 7 中任一权利要求所述的装置,其特征在于,所述装置还包括:

输出模块,用于输出应用程序列表,提示用户在所述应用程序列表中选择提高运行速度的应用程序;

接收模块,用于根据用户在所述输出模块输出的应用程序列表中选择的应用程序接收提高所述应用程序运行速度的指令以及所述应用程序的信息。

9. 一种终端,其特征在于,所述终端包括提高应用程序运行速度的装置;

其中,所述提高应用程序运行速度的装置如所述权利要求 5 至 8 中任一权利要求所述的装置。

## 提高应用程序运行速度的方法、装置及终端

### 技术领域

[0001] 本发明涉及计算机领域,特别涉及一种提高应用程序运行速度的方法、装置及终端。

### 背景技术

[0002] 随着计算机技术的高速发展,应用程序的种类越来越多。为了满足用户对各类应用程序的运行需求,提高应用程序的运行速度,成为了众多研究者十分关注的课题。

[0003] 现有技术为提高应用程序的运行速度时,采取了预读和预加载(Prefetch)两种机制。其中,预读机制是指在读取本地数据文件时,除了读取当前所需数据外,还将当前所需数据后的内容也一并读入内存中,以在用到该部分内容时,可从内存中直接读取,从而减少应用程序在 IO (Input/Output, 输入输出)上的耗时;预加载机制是指在开机加载操作系统时,读取常用的应用程序的主要内容,以备在该类应用程序启动时耗费大量时间来读取数据。

[0004] 在实现本发明的过程中,发明人发现现有技术至少存在以下问题:

[0005] 预读机制仅适用于在加载应用程序时顺序访问本地数据的场景,针对应用程序运行时对本地数据进行随机访问的场景,该预读机制的作用并不显著;预加载机制针对所有应用程序运行速度进行优化,具有一定普遍性但缺乏针对性,对应用程序运行速度的优化空间有限。

### 发明内容

[0006] 为了在提高应用程序运行速度的同时,提高速度优化的针对性,并增强速度提高的效果,本发明实施例提供了一种提高应用程序运行速度的方法、装置及终端。所述技术方案如下:

[0007] 一方面,提供了一种提高应用程序运行速度的方法,所述方法包括:

[0008] 采集应用程序运行时缺页中断对应的本地数据信息,所述本地数据信息包括本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;

[0009] 根据采集到的所述本地数据信息获取对应的本地数据,并在获取到提高所述应用程序运行速度的指令后,加载获取到的所述本地数据;

[0010] 其中,所述采集应用程序运行时缺页中断对应的本地数据信息,具体包括:

[0011] 通过事件跟踪机制采集应用程序运行时的硬缺页事件和软缺页事件,从记录了所述硬缺页事件和软缺页事件的事件跟踪日志中获取所述应用程序运行时缺页中断对应的本地数据信息;

[0012] 或者,所述采集应用程序运行时缺页中断对应的本地数据信息,具体包括:

[0013] 通过钩子 HOOK 技术对系统底层调用进行监控,并获取用于记录缺页中断对应的本地数据信息的监控代码,根据所述监控代码获取所述应用程序运行时缺页中断对应的本地数据信息。

- [0014] 优选地,所述根据采集到的所述本地数据信息获取对应的本地数据,具体包括:
- [0015] 解析采集到的所述本地数据信息,得到本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;
- [0016] 按照所述本地数据所在文件的文件名将采集到的本地数据信息进行聚合;
- [0017] 按照所述本地数据在所述文件内的偏移将聚合后的本地数据信息进行合并,并在合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息;
- [0018] 根据选取的所述预取数据信息获取对应的本地数据。
- [0019] 进一步地,所述在合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息之后,还包括:
- [0020] 根据所述应用程序的信息存储选取的所述预取数据信息;
- [0021] 所述加载获取到的本地数据,具体包括:
- [0022] 根据所述应用程序的信息获取所述预取数据信息,并根据所述预取数据信息获取对应的本地数据,将获取到的本地数据加载至内存。
- [0023] 进一步地,所述加载获取到的本地数据之前,还包括:
- [0024] 输出应用程序列表,提示用户在所述应用程序列表中选择提高运行速度的应用程序;
- [0025] 根据用户在所述应用程序列表中选择的应用程序接收提高所述应用程序运行速度的指令以及所述应用程序的信息。
- [0026] 另一方面,还提供了一种提高应用程序运行速度的装置,所述装置包括:
- [0027] 采集模块,用于采集应用程序运行时缺页中断对应的本地数据信息,所述本地数据信息包括本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;
- [0028] 获取模块,用于根据所述采集模块采集到的所述本地数据信息获取对应的本地数据;
- [0029] 加载模块,用于获取到提高所述应用程序运行速度的指令后,加载所述获取模块获取到的所述本地数据;
- [0030] 其中,所述采集模块,具体用于通过事件跟踪机制采集应用程序运行时的硬缺页事件和软缺页事件,从记录了所述硬缺页事件和软缺页事件的事件跟踪日志中获取所述应用程序运行时缺页中断对应的本地数据信息;
- [0031] 或者,所述采集模块,具体用于通过钩子 HOOK 技术对系统底层调用进行监控,并获取用于记录缺页中断对应的本地数据信息的监控代码,根据所述监控代码获取所述应用程序运行时缺页中断对应的本地数据信息。
- [0032] 优选地,所述获取模块,具体包括:
- [0033] 解析单元,用于解析所述采集模块采集到的本地数据信息,得到本地数据所在文件的文件名、所述本地数据在所述文件内的偏移以及所述本地数据的大小;
- [0034] 聚合单元,用于按照所述解析单元解析出的本地数据所在文件的文件名将采集到的本地数据信息进行聚合;
- [0035] 合并单元,用于按照所述解析单元解析出的本地数据在所述文件内的偏移将所述聚合单元聚合后的本地数据信息进行合并;

- [0036] 选取单元,用于在所述合并单元合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息;
- [0037] 获取单元,用于根据所述选取单元选取的所述预取数据信息获取对应的本地数据。
- [0038] 进一步地,所述装置还包括:
- [0039] 存储模块,用于根据所述应用程序的信息存储所述选取单元选取的所述预取数据信息;
- [0040] 所述加载模块,具体包括:
- [0041] 第一获取单元,用于根据所述应用程序的信息获取所述预取数据信息;
- [0042] 第二获取单元,用于根据所述第一获取单元获取到的预取数据信息获取对应的本地数据;
- [0043] 加载单元,用于将所述第二获取单元获取到的本地数据加载至内存。
- [0044] 进一步地,所述装置还包括:
- [0045] 输出模块,用于输出应用程序列表,提示用户在所述应用程序列表中选择提高运行速度的应用程序;
- [0046] 接收模块,用于根据用户在所述输出模块输出的应用程序列表中选择的应用程序接收提高所述应用程序运行速度的指令以及所述应用程序的信息。
- [0047] 再一方面,还提供了一种终端,所述终端包括上述提高应用程序运行速度的装置。
- [0048] 本发明实施例提供的技术方案带来的有益效果是:
- [0049] 通过采集应用程序运行时缺页中断对应的本地数据信息,并加载根据本地数据信息获取到的本地数据,从而减少应用程序访问 I/O 操作的耗时,达到提高应用程序运行速度的目的;另外,由于缺页中断对应的本地数据信息不局限于本地数据的访问场景,因而可增强速度提高的效果;且由于不同应用程序运行时缺页中断对应的本地数据信息不同,针对不同应用程序具有不同的速度优化程度,因而可提高速度优化的针对性,进而扩大速度优化的空间。

#### 附图说明

- [0050] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中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。
- [0051] 图 1 是本发明实施例一提供的提高应用程序运行速度的方法流程图;
- [0052] 图 2 是本发明实施例二提供的提高应用程序运行速度的方法流程图;
- [0053] 图 3 是本发明实施例三提供的提高应用程序运行速度的装置的结构示意图;
- [0054] 图 4 是本发明实施例三提供的获取模块的结构示意图;
- [0055] 图 5 是本发明实施例三提供的另一种提高应用程序运行速度的装置的结构示意图;
- [0056] 图 6 是本发明实施例三提供的加载模块的结构示意图;
- [0057] 图 7 是本发明实施例三提供的又一种提高应用程序运行速度的装置的结构示意图;

图。

### 具体实施方式

[0058] 为使本发明的目的、技术方案和优点更加清楚，下面将结合附图对本发明实施方式作进一步地详细描述。

[0059] 实施例一

[0060] 本实施例提供了一种提高应用程序运行速度的方法，参见图 1，本实施例提供的方法流程具体如下：

[0061] 101：采集应用程序运行时缺页中断对应的本地数据信息，该本地数据信息包括本地数据所在文件的文件名、本地数据在文件内的偏移以及本地数据的大小；

[0062] 其中，本实施例不对采集应用程序运行时缺页中断对应的本地数据信息的具体方式进行限定，包括但不限于如下方式：

[0063] 通过事件跟踪机制采集应用程序运行时的硬缺页事件和软缺页事件，从记录了硬缺页事件和软缺页事件的事件跟踪日志中获取应用程序运行时缺页中断对应的本地数据信息；

[0064] 或者，通过钩子 HOOK 技术对系统底层调用进行监控，并获取用于记录缺页中断对应的本地数据信息的监控代码，根据监控代码获取应用程序运行时缺页中断对应的本地数据信息。

[0065] 102：根据采集到的本地数据信息获取对应的本地数据，并在获取到提高应用程序运行速度的指令后，加载获取到的本地数据。

[0066] 优选地，根据采集到的本地数据信息获取对应的本地数据，具体包括：

[0067] 解析采集到的本地数据信息，得到本地数据所在文件的文件名、本地数据在文件内的偏移以及本地数据的大小；

[0068] 按照本地数据所在文件的文件名将采集到的本地数据信息进行聚合；

[0069] 按照本地数据在文件内的偏移将聚合后的本地数据信息进行合并，并在合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息；

[0070] 根据选取的预取数据信息获取对应的本地数据。

[0071] 进一步地，在合并后的本地数据信息中选取符合预取条件的本地数据信息作为预取数据信息之后，还包括：

[0072] 根据应用程序的信息存储选取的预取数据信息；

[0073] 加载获取到的本地数据，具体包括：

[0074] 根据应用程序的信息获取预取数据信息，并根据预取数据信息获取对应的本地数据，将获取到的本地数据加载至内存。

[0075] 进一步地，加载获取到的本地数据之前，还包括：

[0076] 输出应用程序列表，提示用户在应用程序列表中选择提高运行速度的应用程序；

[0077] 根据用户在应用程序列表中选择的应用程序接收提高应用程序运行速度的指令以及应用程序的信息。

[0078] 本实施例提供的方法，通过采集应用程序运行时缺页中断对应的本地数据信息，并加载根据本地数据信息获取到的本地数据，从而减少应用程序访问 IO 操作的耗时，达到

提高应用程序运行速度的目的；另外，由于缺页中断对应的本地数据信息不局限于本地数据的访问场景，因而可增强速度提高的效果；且由于不同应用程序运行时缺页中断对应的本地数据信息不同，针对不同应用程序具有不同的速度优化程度，因而可提高速度优化的针对性，进而扩大速度优化的空间。

#### [0079] 实施例二

[0080] 本实施例提供了一种提高应用程序运行速度的方法，该方法通过采集应用程序运行时缺页中断对应的本地数据信息，并在根据该本地数据信息训练得到预取数据信息后，根据预取数据信息加载对应的本地数据，从而减少应用程序访问 I/O 操作的耗时，达到提高应用程序运行的目的。结合上述实施例一的内容，本实施例以本地数据为磁盘数据为例，对提高应用程序运行速度的方法进行举例说明。参见图 2，本实施例提供的方法流程具体如下：

[0081] 201：采集应用程序运行时缺页中断对应的磁盘数据信息，该磁盘数据信息包括磁盘数据所在文件的文件名、磁盘数据在文件内的偏移以及磁盘数据的大小；

[0082] 其中，缺页中断是指要访问的页不在内存，需要操作系统将其调入内存后再进行访问。且缺页中断可分为硬缺页和软缺页，硬缺页事件是指发生缺页中断后，需要的数据不在缓存也不在分页内存中，需要通过 I/O 操作从磁盘上获取磁盘数据。软缺页是指发生缺页中断后，需要的数据已经被载入内存，但又被置换到磁盘中，需要从磁盘交换文件中将这块内存信息再载入内存中。由此可见，无论是硬缺页还是软缺页，缺页中断都将影响应用程序运行的速度，因此，为了提高应用程序运行速度，本实施例采取了采集应用程序运行时缺页中断对应的磁盘数据信息，后续步骤再通过提前加载根据磁盘数据信息获取到的磁盘数据的方式，避免发生缺页中断，进而提高应用程序运行速度。采集应用程序运行时缺页中断对应的磁盘数据信息时，包括但不限于如下方式：

[0083] 通过事件跟踪机制采集应用程序运行时的硬缺页事件和软缺页事件，从记录了硬缺页事件和软缺页事件的事件跟踪日志中获取应用程序运行时缺页中断对应的磁盘数据信息；

[0084] 或者，通过 HOOK 技术对系统底层调用进行监控，并获取用于记录缺页中断对应的磁盘数据信息的监控代码，根据监控代码获取应用程序运行时缺页中断对应的磁盘数据信息。

[0085] 具体地，事件跟踪机制可以采用 ETW (Event Tracing for Windows, Windows 事件跟踪)，ETW 是由操作系统提供的一种通用的、与性能日志和警报相比系统开销较低的事件追踪手段，用以监控具有负载的系统的性能。ETW 主要用于频繁记录事件、错误、警告或审核的服务器应用程序。ETW 提供用户模式的应用程序和内核模式的设备驱动程序所触发的事件追踪机制。此外，ETW 还能够动态地启用或者禁用日志记录，便于进行详细的追踪，而无需重新启动操作系统或者应用程序，因此，可通过 ETW 采集应用程序运行时的硬缺页事件和软缺页事件，并从记录了硬缺页事件和软缺页事件的事件跟踪日志中获取应用程序运行时缺页中断对应的磁盘数据信息。

[0086] 当然，除了采用上述两种采集方式外，还可以选择其他采集方式，本实施对此不做具体限定。无论采用哪种采集方式，在采集到应用程序运行时缺页中断对应的磁盘数据信息后，均可将其转化成磁盘内容描述格式 `<file, offset, length>`，其中，file 为磁盘数据



所在文件的文件名, offset 为磁盘数据在文件内的偏移, length 为磁盘数据的大小, 可以 byte (字节) 为单位。

[0087] 202: 根据采集到的磁盘数据信息得到预取数据信息;

[0088] 该步骤为优选步骤, 根据采集到的磁盘数据信息得到预取数据信息的方式, 包括但不限于如下步骤:

[0089] 解析采集到的磁盘数据信息, 得到磁盘数据所在文件的文件名、磁盘数据在文件内的偏移以及磁盘数据的大小;

[0090] 按照磁盘数据所在文件的文件名将采集到的磁盘数据信息进行聚合;

[0091] 按照磁盘数据在文件内的偏移将聚合后的磁盘数据信息进行合并, 并在合并后的磁盘数据信息中选取符合预取条件的磁盘数据信息作为预取数据信息。

[0092] 其中, 本实施例不对解析采集到的磁盘数据信息的方式进行限定, 为了便于说明, 本实施例以解析出的磁盘数据信息如下面表 1 所示为例, 对该步骤中得到预取数据信息的方式进行举例说明。

[0093] 表 1

[0094]

File	Offset	Length (byte)
Example1. dll	0	1024
Example2. dll	4096	512
Example3. data	0	10240
Example1. dll	1024	1024
Example2. dll	0	512

[0095] 从上面表 1 的内容中可以看出, 解析出了 5 段磁盘数据信息, 按照磁盘数据信息中磁盘数据所在文件的文件名对采集到的磁盘数据信息进行聚合时, 可将文件名相同的磁盘数据信息聚合在一起, 例如, 将文件名为 Example1. dll 的两段磁盘数据信息进行聚合, 将文件名为 Example2. dll 的两段磁盘数据信息进行聚合, 则得到的聚合结果可如下面表 2 所示:

[0096] 表 2

File	Offset	Length (byte)
Example1. dll	0	1024
	1024	1024
Example2. dll	0	512
	4096	512
Example3. data	0	10240

[0097]

[0098] 按照磁盘数据在文件内的偏移将聚合后的磁盘数据信息进行合并时,结合上述表 2 所示的聚合结果,可将对应同一文件的磁盘数据信息按照其中的偏移是否相距预设范围而进行合并,其中,预设范围可以设定为 1024,2056 等,本实施例不对预设范围的具体大小进行限定,此处仅以预设范围为 2056 为例。则对于文件名为 Example2. dll 的两段磁盘数据信息,由于一段磁盘数据信息的偏移为 0,另一段磁盘数据信息的偏移为 4096,由此可见,两段磁盘数据信息的偏移相距 4096,而该距离在预设范围 2056 之外,因而不对这两段磁盘数据信息进行合并。

[0099] 但对于文件名为 Example1. dll 的两段磁盘数据信息,由于一段磁盘数据信息的偏移为 0,另一段磁盘数据信息的偏移为 1024,由此可见,两段磁盘数据信息的偏移相距 1024,该距离在预设范围 2056 内,因此,可将其进行合并,合并之后的结果可如下面表 3 所示:

[0100] 表 3

File	Offset	Length (byte)
Example1. dll	0	2048
Example2. dll	0	512
	4096	512
Example3. data	0	10240

[0102] 经过上面的聚合及合并的过程,应用程序每次运行时都能得到如上表 3 所示结果,对于采集应用程序多次运行时缺页中断对应的磁盘数据信息,并对采集到的结果进行如上训练便得到预取数据信息,预取数据信息同磁盘数据信息一样,均用于记录应用程序很有可能需要通过 IO 访问获取的磁盘数据的信息。不同应用程序对应的预取数据信息也会不同,即使是同一应用程序,在不同场景下的预取数据信息也会有所不同,而该场景包括但不限于加载、运行等场景,如若需要确定应用程序在哪个场景下很有可能需要通过 IO 访问获取的磁盘数据,则在该场景下采集磁盘数据信息,并按照如上过程根据采集到的磁盘数据信息得到对应的预取数据信息,进而通过预取数据信息获取对应的磁盘数据即可。本实施例不对本实施例提供的方法所适用的场景进行限定。

[0103] 可选地,由于应用程序每次运行时都可得到如上面表 3 所示的结果,如果直接将应用程序每次运行时得到的如上面表 3 所示的结果进行合并,并将合并后的结果作为得到的预取数据信息,则该预取数据信息所覆盖的磁盘数据的范围会比较广,然而有些磁盘数据并不是应用程序运行时时常用到的,仅仅是在某次采集过程中采集到该磁盘数据对应的磁盘数据信息而已,因而加载该类磁盘数据不仅对提高应用程序运行速度没有过多的帮助,反而还占用了系统资源,对此,为了在提高应用程序运行速度的前提下,进一步缩小预取数据信息所覆盖的磁盘数据的范围,以减少加载不必要的磁盘数据,导致浪费资源,本实施例提供的方法在通过上述过程得到合并后的磁盘数据信息时,可并不直接将合并后的所有磁盘数据信息均作为预取数据信息,而是在合并后的磁盘数据信息中再挑选出符合预取条件的磁盘数据信息,之后再挑取出的磁盘数据信息作为预取数据信息。

[0104] 对于预取条件,可通过判断采集到的磁盘数据信息是否达到预设次数来得到,例

如,如果某段磁盘数据信息在 10 次采集过程中,仅被采集到了 1 次,则可得出该段磁盘数据信息被采集到的频率不高,应用程序运行时对该磁盘数据信息对应的磁盘数据的需求也不会很高,因而可将其从采集到的磁盘数据信息中进行删除;相反,如果某段磁盘数据信息在 10 次采集过程中,有 6 次被采集到,则可得出该段磁盘数据信息被采集到的频率较高,应用程序运行时对该磁盘数据信息对应的磁盘数据的需求也会相对较高,因而可将其保留在采集到的磁盘数据信息中。当然,除了以采集到的次数是否达到预设次数为标准判断合并后的磁盘数据信息是否符合预取条件外,还可以选择其他预取条件,本实施例不仅不对预设次数的多少进行限定,同样也不对具体的预取条件进行限定。

[0105] 203:根据应用程序的信息存储得到的预取数据信息;

[0106] 针对该步骤,在用户没有运行该应用程序的需求下,为了避免因加载该应用程序对应的磁盘数据而占用系统资源,本实施例提供的方法不会在得到预取数据信息后,直接加载根据预取数据信息获取到的磁盘数据,而是根据用户对该应用程序的运行需求来选择是否加载根据预取数据信息获取到的磁盘数据。之所以根据应用程序的信息存储得到的预取数据信息,是由于每个应用程序均可训练得到对应的预取数据信息,为了区分不同应用程序的预取数据信息,可根据各个应用程序的信息存储各自的预取数据信息。其中,应用程序的信息包括但不限于应用程序的名称、路径、版本号等,本实施例不对应用程序的具体信息进行限定。

[0107] 具体存储时,可将其存储在本地,也可以存储在网络中,当然,也可以同时存储在本地和网络中,本实施例不对存储位置及方式进行限定。

[0108] 204:输出应用程序列表,提示用户在应用程序列表中选择提高运行速度的应用程序;根据用户在应用程序列表中选择的应用程序接收提高该应用程序运行速度的指令以及该应用程序的信息;

[0109] 其中,在输出应用程序列表时,既可以仅输出应用程序名称,也可以在输出应用程序名称的同时,输出该应用程序的图标,本实施例不对输出应用程序列表的方式及内容进行限定。之所以会输出应用程序列表,提示用户在应用程序列表中选择提高运行速度的应用程序,是为了实现针对每个应用程序分别提高运行速度的目的。例如,通过上述步骤 201 至步骤 203 得到并存储了 10 个应用程序的预取数据信息,在实际应用过程中,并不是这 10 个应用程序都需要运行,因此,为了避免同时加载这 10 个应用程序的磁盘数据而浪费系统资源,本实施例采取了由用户选取的方式,分别加载需要的磁盘数据。如果用户在应用程序列表中选取了哪个应用程序,则意味着用户希望提高该应用程序的运行速度,则在用户从应用程序列表中选应用程序的同时,触发发送提高应用程序运行速度的指令及该应用程序的信息,以触发后续步骤根据该指令及信息加载对应的磁盘数据。其中,该应用程序的信息即为用户选择的应用程序的信息。

[0110] 205:根据该应用程序的信息获取预取数据信息,根据预取数据信息获取对应的磁盘数据,将获取到的磁盘数据加载至内存。

[0111] 具体地,无论上述步骤 203 将预取数据信息是存储在本地还是在网络,由于是根据应用程序的信息进行存储的,因而每个预取数据信息均与应用程序的信息之间存在对应关系,则在上述步骤 204 获取到应用程序的信息之后,即可根据该应用程序的信息获取对应的预取数据信息。又由于预取数据信息中记录了磁盘数据的信息,因而可根据预取数据

信息获取到对应的磁盘数据。

[0112] 将获取到的磁盘数据加载至内存时,即可以在系统空闲时加载也可以在应用程序运行时加载,本实施例不对加载磁盘数据的时间进行限定。在系统空闲时加载获取到的磁盘数据时,可先判断内存中是否存在待加载的磁盘数据,如果已经存在该磁盘数据,则无需再将其加载至缓存,如果不存在该磁盘数据,则执行将其加载的步骤。相应的,在应用程序启动时加载获取到的磁盘数据时,可先判断该应用程序是否启动,如果启动,则再加载获取到的磁盘数据。无论是在系统空闲时加载获取到的磁盘数据,还是在应用程序启动时加载获取到的磁盘数据,具体加载过程均可参照现有的加载流程实现,本实施例不对具体的加载方式进行限定。

[0113] 本实施例提供的方法,通过采集应用程序运行时缺页中断对应的本地数据信息,并加载根据本地数据信息获取到的本地数据,从而减少应用程序访问 IO 操作的耗时,达到提高应用程序运行速度的目的;另外,由于缺页中断对应的本地数据信息不局限于本地数据的访问场景,因而可增强速度提高的效果;且由于不同应用程序运行时缺页中断对应的本地数据信息不同,针对不同应用程序具有不同的速度优化程度,因而可提高速度优化的针对性,进而扩大速度优化的空间。

[0114] 实施例三

[0115] 参见图 3,本发明实施例提供了一种提高应用程序运行速度的装置,该装置包括:

[0116] 采集模块 301,用于采集应用程序运行时缺页中断对应的本地数据信息,本地数据信息包括本地数据所在文件的文件名、本地数据在文件内的偏移以及本地数据的大小;

[0117] 获取模块 302,用于根据采集模块 301 采集到的本地数据信息获取对应的本地数据;

[0118] 加载模块 303,用于获取到提高应用程序运行速度的指令后,加载获取模块 302 获取到的本地数据。

[0119] 其中,采集模块 301,具体用于通过事件跟踪机制采集应用程序运行时的硬缺页事件和软缺页事件,从记录了硬缺页事件和软缺页事件的事件跟踪日志中获取应用程序运行时缺页中断对应的本地数据信息;

[0120] 或者,采集模块 301,具体用于通过钩子 HOOK 技术对系统底层调用进行监控,并获取用于记录缺页中断对应的本地数据信息的监控代码,根据监控代码获取应用程序运行时缺页中断对应的本地数据信息。

[0121] 该采集模块 301 采集应用程序运行时缺页中断对应的本地数据信息的方式,详见上述实施例二中步骤 201 的相关描述,此处不再赘述。

[0122] 优选地,参见图 4,该获取模块 302,具体包括:

[0123] 解析单元 302a,用于解析采集模块 301 采集到的本地数据信息,得到本地数据所在文件的文件名、本地数据在文件内的偏移以及本地数据的大小;

[0124] 聚合单元 302b,用于按照解析单元 302a 解析出的本地数据所在文件的文件名将采集到的本地数据信息进行聚合;

[0125] 合并单元 302c,用于按照解析单元 302a 解析出的本地数据在文件内的偏移将聚合单元 302b 聚合后的本地数据信息进行合并;

[0126] 选取单元 302d,用于在合并单元 302c 合并后的本地数据信息中选取符合预取条

件的本地数据信息作为预取数据信息；

[0127] 获取单元 302e,用于根据选取单元 302d 选取的预取数据信息获取对应的本地数据。

[0128] 其中,解析单元 302a 解析本地数据信息的方式、聚合单元 302b 将采集到的本地数据信息进行聚合的方式、合并单元 302c 将聚合单元 302b 聚合后的本地数据信息进行合并的方式,以及选取单元 302d 在合并单元 302c 合并后的本地数据信息中选取符合预取条件的本地数据信息的方式,均可详见上述实施例二中步骤 202 的相关描述,此处不再赘述。

[0129] 进一步地,参见图 5,该装置还包括：

[0130] 存储模块 304,用于根据应用程序的信息存储选取单元 302d 选取的预取数据信息；

[0131] 参见图 6,该加载模块 303,具体包括：

[0132] 第一获取单元 3031,用于根据应用程序的信息获取存储模块 304 存储的预取数据信息；

[0133] 第二获取单元 3032,用于根据第一获取单元 3031 获取到的预取数据信息获取对应的本地数据；

[0134] 加载单元 3033,用于将第二获取单元 3032 获取到的本地数据加载至内存。

[0135] 其中,存储模块 304 存储预取数据信息的方式详见上述实施例二中步骤 203 的相关描述,此处不再赘述。第一获取单元 3031 获取预取数据信息的方式、第二获取单元 3032 获取本地数据的方式,以及加载单元 3033 加载本地数据的方式均详见上述实施例二中步骤 205 的相关描述,此处不再赘述。

[0136] 进一步地,参见图 7,该装置还包括：

[0137] 输出模块 305,用于输出应用程序列表,提示用户在应用程序列表中选择提高运行速度的应用程序；

[0138] 接收模块 306,用于根据用户在输出模块 305 输出的应用程序列表中选择的应用程序接收提高应用程序运行速度的指令以及应用程序的信息。

[0139] 该输出模块 305 输出应用程序列表的方式、接收模块 306 接收提高应用程序运行速度的指令以及应用程序的信息的方式均可详见上述实施例二中步骤 204 的相关描述,此处不再赘述。

[0140] 本实施例提供的装置,通过采集应用程序运行时缺页中断对应的本地数据信息,并加载根据本地数据信息获取到的本地数据,从而减少应用程序访问 I/O 操作的耗时,达到提高应用程序运行速度的目的;另外,由于缺页中断对应的本地数据信息不局限于本地数据的访问场景,因而可增强速度提高的效果;且由于不同应用程序运行时缺页中断对应的本地数据信息不同,针对不同应用程序具有不同的速度优化程度,因而可提高速度优化的针对性,进而扩大速度优化的空间。

[0141] 实施例四

[0142] 本实施例提供了一种终端,该终端包括提高应用程序运行速度的装置；

[0143] 其中,该提高应用程序运行速度的装置可如上述实施例三提供的装置,具体详见上述实施例三的描述,此处不再赘述。

[0144] 本实施例提供的终端包括但不限于手机、平板电脑、PDA (Personal Digital

Assistant,个人数字助理)等。

[0145] 本实施例提供的终端,通过提高应用程序运行速度的装置采集应用程序运行时缺页中断对应的本地数据信息,并加载根据本地数据信息获取到的本地数据,从而减少应用程序访问 IO 操作的耗时,达到提高应用程序运行速度的目的;另外,由于缺页中断对应的本地数据信息不局限于本地数据的访问场景,因而可增强速度提高的效果;且由于不同应用程序运行时缺页中断对应的本地数据信息不同,针对不同应用程序具有不同的速度优化程度,因而可提高速度优化的针对性,进而扩大速度优化的空间。

[0146] 需要说明的是:上述实施例提供的提高应用程序运行速度的装置在提高应用程序运行速度时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将装置的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的提高应用程序运行速度的装置、终端与提高应用程序运行速度的方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0147] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0148] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来指令相关的硬件完成,所述的程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0149] 以上所述仅为本发明的较佳实施例,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

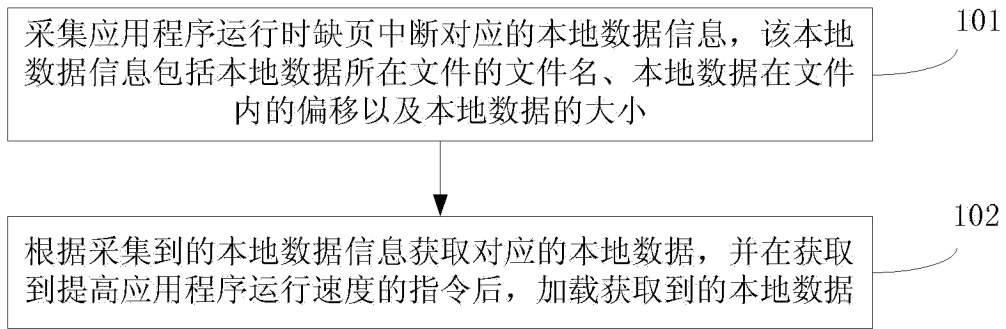


图 1

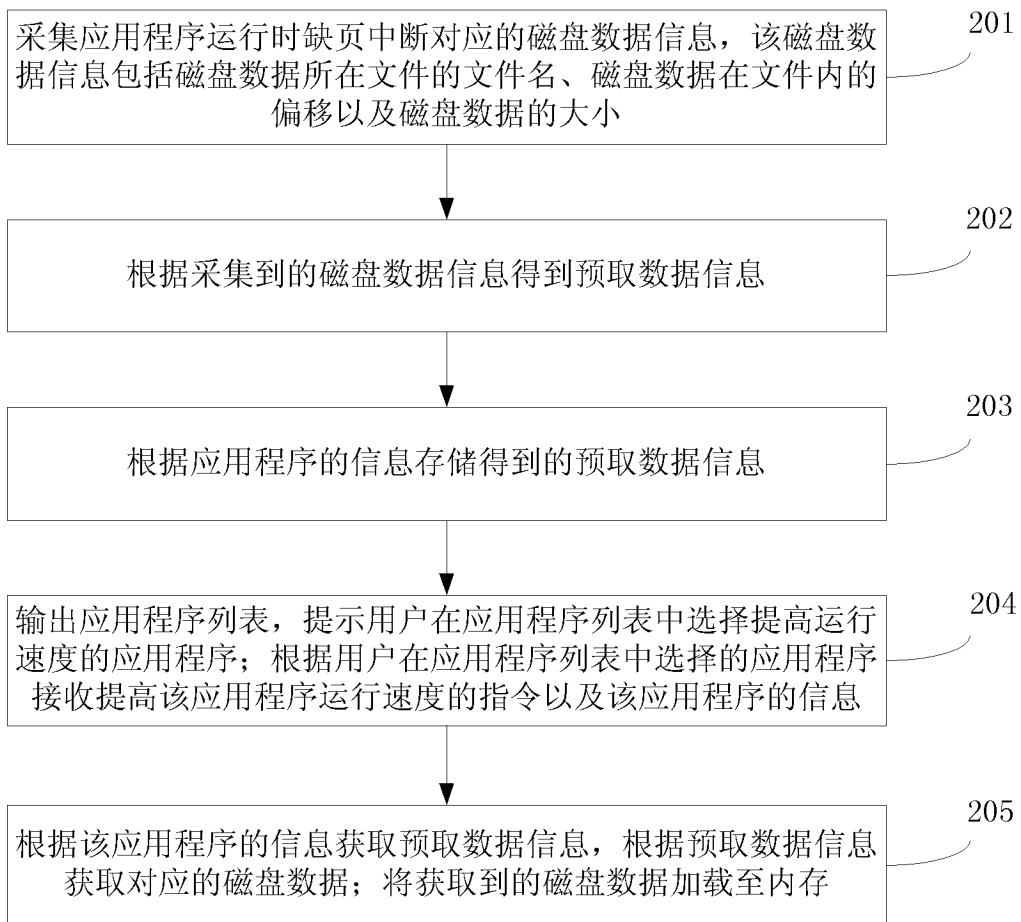


图 2

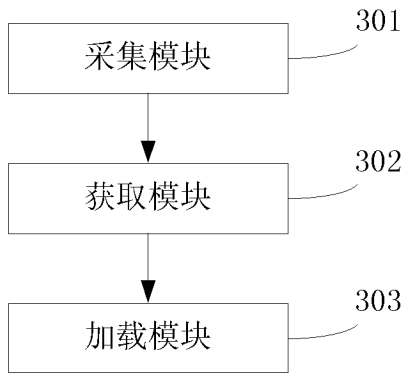


图 3

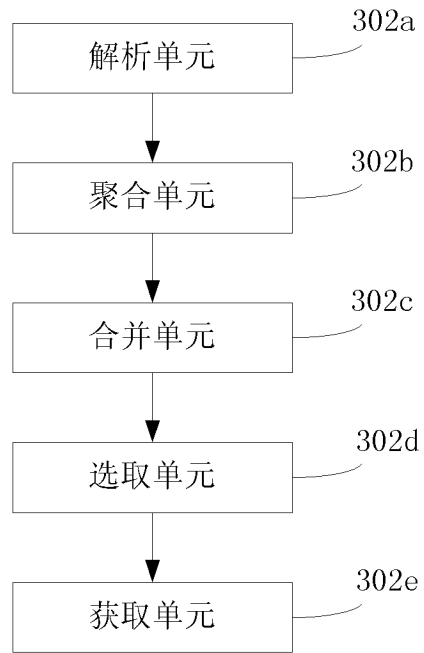


图 4

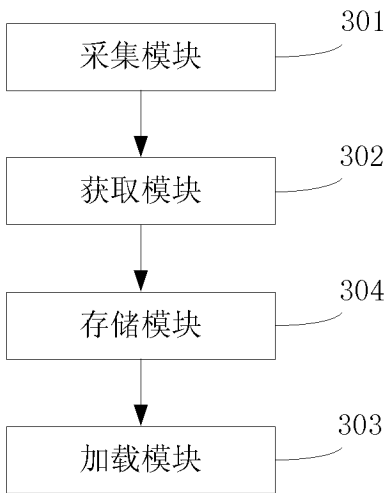


图 5

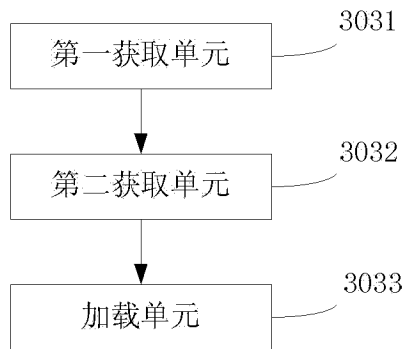


图 6

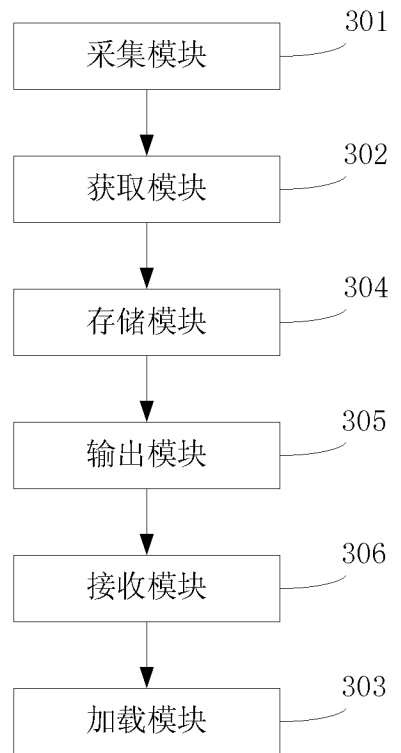


图 7