



(12) 发明专利

(10) 授权公告号 CN 102662690 B

(45) 授权公告日 2014. 06. 11

(21) 申请号 201210067122. 3

审查员 刘佳

(22) 申请日 2012. 03. 14

(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

(72) 发明人 边超 朱文焕

(74) 专利代理机构 北京三高永信知识产权代理有限公司 11138

代理人 罗振安

(51) Int. Cl.

G06F 9/445(2006. 01)

(56) 对比文件

WO 2010/140403 A1, 2010. 12. 09,

CN 101126995 A, 2008. 02. 20,

权利要求书2页 说明书8页 附图3页

(54) 发明名称

应用程序启动方法和装置

(57) 摘要

本发明公开了一种应用程序启动方法和装置,属于计算机领域。所述方法包括:当接收到启动应用程序的指令时,加载所述应用程序对应的预取数据,所述预取数据包括所述应用程序的历史启动过程中发生的页面错误对应的文件块;加载所述应用程序的代码,以便启动所述应用程序。该装置包括:预取数据加载模块和代码加载模块。本发明通过在应用程序启动时,先加载所述应用程序的历史启动过程中发生的页面错误对应的文件块,再加载应用程序代码,大大减少了应用程序启动过程中页面错误发生的次数,减轻了程序载入过程中频繁交换内存页面与交换文件的现象,提高了应用程序的启动速度和启动效率。

当接收到启动应用程序的指令时,加载所述应用程序对应的预取数据,所述预取数据包括所述应用程序的历史启动过程中发生的页面错误对应的文件块 101

加载该应用程序的代码,以便启动该应用程序 102

1. 一种应用程序启动方法,其特征在于,所述方法包括:

获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址;

根据所述映像文件列表和所述虚拟地址计算每个所述虚拟地址对应的文件偏移量;

根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并,得到所述应用程序对应的预取数据;

当接收到启动应用程序的指令时,加载所述应用程序对应的预取数据,所述预取数据包括所述应用程序的历史启动过程中发生的页面错误对应的文件块;

加载所述应用程序的代码,以便启动所述应用程序。

2. 根据权利要求 1 所述的方法,其特征在于,获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址,具体包括:

获取所述应用程序的日志文件;

将所述日志文件中从用户界面进程开始时间到显示前景窗口时间之间的时间段确定为预设启动时间段;

获取所述预设启动时间段内进程加载的映像文件列表;

获取所述预设启动时间段内进程发生的页面错误的虚拟地址。

3. 根据权利要求 1 所述的方法,其特征在于,根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并操作,得到所述应用程序对应的预取数据,具体包括:

根据所述每个所述虚拟地址对应的文件偏移量,将距离小于预设比特数的文件块进行合并操作,得到所述应用程序对应的预取数据。

4. 一种应用程序启动装置,其特征在于,所述装置包括:

获取模块,用于获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址;

偏移量计算模块,用于根据所述映像文件列表和所述虚拟地址计算每个所述虚拟地址对应的文件偏移量;

预取数据获取模块,用于根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并,得到所述应用程序对应的预取数据;

预取数据加载模块,用于当接收到启动应用程序的指令时,加载所述应用程序对应的预取数据,所述预取数据包括所述应用程序的历史启动过程中发生的页面错误对应的文件块;

代码加载模块,用于加载所述应用程序的代码,以便启动所述应用程序。

5. 根据权利要求 4 所述的装置,其特征在于,所述获取模块包括:

日志获取单元,用于获取所述应用程序的日志文件;

时间段确定单元,用于将所述日志文件中从用户界面进程开始时间到显示前景窗口时间之间的时间段确定为预设启动时间段;

映像文件列表获取单元,用于获取所述预设启动时间段内进程加载的映像文件列表;

虚拟地址获取单元,用于获取所述预设启动时间段内进程发生的页面错误的虚拟地址。

6. 根据权利要求 4 所述的装置,其特征在于,所述预取数据获取模块具体用于根据所

述每个所述虚拟地址对应的文件偏移量,将距离小于预设比特数的文件块进行合并操作,得到所述应用程序对应的预取数据。

应用程序启动方法和装置

技术领域

[0001] 本发明涉及计算机领域,特别涉及一种应用程序启动方法和装置。

背景技术

[0002] 随着计算机技术的飞速发展,用户会安装各种各样的应用程序,应用程序启动过程中的磁盘 I/O 基本上是由页面错误(Page Fault)引入的,当执行到某处而其对应的虚拟地址并不在进程的工作集(Working Set)中时,就会发生 Page Fault, Page Fault 的顺序一般是这样的:从一个文件的某部分中读入一些页面,然后可能从同一文件的相距较远的另一部分中读入一些页面,然后从另一个不同的文件中读入一些页面,可能又要从一个目录中读入页面,然后再从第一个文件中读入一些页面,读取文件不连续,且读取位置也有可能不连续,这种跳跃的读取大大减慢了每一次启动应用程序的速度,如果对应的内容不在物理内存里,就需要从文件中读取,触发 Hard Fault,应用程序冷启动时间较长的一个主要原因就是 Hard Fault 过多。而 Hard Fault 是按需读取的(一次 16KB 或者 32KB),再考虑到系统其他的 I/O,就可能造成 I/O 偏移量在磁盘上的跳跃,从而大大减慢了每一次访问的速度,使得应用程序的启动效率降低,而启动速度是应用程序的最重要性能指标之一,直接影响着用户的对应用程序的体验印象。

发明内容

[0003] 为了加快应用程序的启动速度,本发明实施例提供了一种应用程序启动方法和装置。所述技术方案如下:

[0004] 本发明实施例提供了一种应用程序启动方法,所述方法包括:

[0005] 获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址;

[0006] 根据所述映像文件列表和所述虚拟地址计算每个所述虚拟地址对应的文件偏移量;

[0007] 根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并,得到所述应用程序对应的预取数据;

[0008] 当接收到启动应用程序的指令时,加载所述应用程序对应的预取数据,所述预取数据包括所述应用程序的历史启动过程中发生的页面错误对应的文件块;

[0009] 加载所述应用程序的代码,以便启动所述应用程序。

[0010] 获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址,具体包括:

[0011] 获取所述应用程序的日志文件;

[0012] 将所述日志文件中从用户界面进程开始时间到显示前景窗口时间之间的时间段确定为预设启动时间段;

[0013] 获取所述预设启动时间段内进程加载的映像文件列表;

- [0014] 获取所述预设启动时间段内进程发生的页面错误的虚拟地址。
- [0015] 根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并操作,得到所述应用程序对应的预取数据,具体包括:
- [0016] 根据所述每个所述虚拟地址对应的文件偏移量,将距离小于预设比特数的文件块进行合并操作,得到所述应用程序对应的预取数据。
- [0017] 本发明实施例提供了一种应用程序启动装置,所述装置包括:
- [0018] 获取模块,用于获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址;
- [0019] 偏移量计算模块,用于根据所述映像文件列表和所述虚拟地址计算每个所述虚拟地址对应的文件偏移量;
- [0020] 预取数据获取模块,用于根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并,得到所述应用程序对应的预取数据;
- [0021] 预取数据加载模块,用于当接收到启动应用程序的指令时,加载所述应用程序对应的预取数据,所述预取数据包括所述应用程序的历史启动过程中发生的页面错误对应的文件块;
- [0022] 代码加载模块,用于加载所述应用程序的代码,以便启动所述应用程序。
- [0023] 所述获取单元包括:
- [0024] 日志获取子单元,用于获取所述应用程序的日志文件;
- [0025] 时间段确定子单元,用于将所述日志文件中从用户界面进程开始时间到显示前景窗口时间之间的时间段确定为预设启动时间段;
- [0026] 映像文件列表获取子单元,用于获取所述预设启动时间段内进程加载的映像文件列表;
- [0027] 虚拟地址获取子单元,用于获取所述预设启动时间段内进程发生的页面错误的虚拟地址。
- [0028] 所述预取数据获取单元具体用于根据所述每个所述虚拟地址对应的文件偏移量,将距离小于预设比特数的文件块进行合并操作,得到所述应用程序对应的预取数据。
- [0029] 本发明实施例提供的技术方案带来的有益效果是:
- [0030] 通过在应用程序启动时,先加载所述应用程序的历史启动过程中发生的页面错误对应的文件块,再加载应用程序代码,大大减少了应用程序启动过程中页面错误发生的次数,减轻了程序载入过程中频繁交换内存页面与交换文件的现象,提高了应用程序的启动速度和启动效率。

附图说明

[0031] 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。

[0032] 图 1 是本发明实施例提供了一种应用程序启动方法的流程图;

[0033] 图 2 是本发明实施例提供了一种应用程序启动方法的流程图;

[0034] 图 3 是现有技术应用程序启动时间与采用本发明实施例的应用程序启动方法的启动时间的对比示意图；

[0035] 图 4 是本发明实施例提供的一种应用程序启动装置的结构示意图。

具体实施方式

[0036] 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。

[0037] 在介绍本发明提供的应用程序启动方法和装置之前，对本发明涉及的关键用语进行以下说明：

[0038] 页面错误：当前进程访问的虚拟页面没有被映射到有效的物理页面时，应用程序要使用一个保存在硬盘上的虚拟内存页面，则操作系统会在硬盘上找到该页面，并将其调入物理内存中，然后程序才能运行。如果虚拟页面被映射到内存中其他状态的物理页面，发生的是软页面错误(Soft Fault)，如果虚拟页面被映射到磁盘上的文件中，发生的是硬页面错误(Hard Fault)。

[0039] 映像文件：将磁盘上的文件映射到虚拟地址空间，通过访问内存一样访问这些文件，这些被映射的文件称为映像文件，进程启动过程中使用映像文件载入 EXE 和 DLL 文件。

[0040] 磁盘 I/O：磁盘的读写。

[0041] 图 1 是本发明实施例提供的一种应用程序启动方法的流程图。参见图 1，该实施例包括：

[0042] 101、当接收到启动应用程序的指令时，加载该应用程序对应的预取数据，该预取数据包括该应用程序的历史启动过程中发生的页面错误对应的文件块；

[0043] 102、加载该应用程序的代码，以便启动该应用程序。

[0044] 可选地，当接收到启动应用程序的指令时，加载所述应用程序对应的预取数据，之前包括：

[0045] 分析所述应用程序的历史启动过程，获取所述历史启动过程中应用程序对应的预取数据。

[0046] 可选地，分析所述应用程序的历史启动过程，获取所述历史启动过程中应用程序对应的预取数据，具体包括：

[0047] 获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址；

[0048] 根据所述映像文件列表和所述虚拟地址计算每个所述虚拟地址对应的文件偏移量；

[0049] 根据所述每个所述虚拟地址对应的文件偏移量，对文件块进行合并，得到所述应用程序对应的预取数据。

[0050] 可选地，获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址，具体包括：

[0051] 获取所述应用程序的日志文件；

[0052] 将所述日志文件中从用户界面进程开始时间到显示前景窗口时间之间的时间段确定为预设启动时间段；

[0053] 获取所述预设启动时间段内进程加载的映像文件列表；

[0054] 获取所述预设启动时间段内进程发生的页面错误的虚拟地址。

[0055] 可选地,根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并操作,得到所述应用程序对应的预取数据,具体包括:

[0056] 根据所述每个所述虚拟地址对应的文件偏移量,将距离小于预设比特数的文件块进行合并操作,得到所述应用程序对应的预取数据。

[0057] 本实施例提供的方法,通过在应用程序启动时,先加载所述应用程序的历史启动过程中发生的页面错误对应的文件块,再加载应用程序代码,大大减少了应用程序启动过程中页面错误发生的次数,减轻了程序载入过程中频繁交换内存页面与交换文件的现象,提高了应用程序的启动速度和启动效率。

[0058] 图2是本发明实施例提供的一种应用程序启动方法的流程图。该实施例的执行主体为终端,该终端可以为PC(Personal Computer,个人电脑)或移动终端,该移动终端可以为智能手机、平板电脑、MP3(Moving Picture Experts Group Audio Layer III,动态影像专家压缩标准音频层面3)、PDA(Personal Digital Assistant,个人数字助理)等。参见图2,该实施例具体包括:

[0059] 201、获取该应用程序的日志文件;

[0060] 其中,日志文件是包含关于系统消息的文件,不同的日志文件记载不同的信息,例如,有的用于加载默认的系统日志文件,有的仅用于记载安全消息,对于该步骤201来说,获取的该应用程序的日志文件用于记载该应用程序的启动和运行相关信息,该日志文件至少包括该应用程序的启动信息,该启动信息包括应用程序的历史启动过程中发生的Page Fault信息。

[0061] 该步骤201中获取的日志文件可以基于ETW(Event Tracing for Windows),该ETW是Windows提供的统一的跟踪和记录事件的机制,用户模式应用程序和内核模式驱动程序都可以使用ETW来记录事件。

[0062] 202、将该日志文件中从用户界面进程开始到显示前景窗口的时间段确定为预设启动时间段;

[0063] 其中,用户界面进程的开始时间可以理解为应用程序的启动时间,而显示前景窗口的时间可以理解为应用程序完成代码加载的时间。在本实施例中,仅以将从用户界面进程开始到显示前景窗口的时间段确定为预设启动时间段为例进行说明,该预设启动时间段还可以缩短到更小的范围内,具体可以由技术人员设置,本发明实施例不做具体限定。

[0064] 203、获取该预设启动时间段内进程加载的映像文件列表;

[0065] 其中,映像文件列表至少包括所加载文件的起始地址和路径,对于该应用程序来说,该映像文件列表即是由于应用程序启动过程中发生的Page Fault而加载的,因此,获取该映像文件列表,可以获知在应用程序启动过程中发生Page Fault时加载了哪些文件以及这些文件的路径。

[0066] 204、获取该预设启动时间段内进程发生的页面错误的虚拟地址;

[0067] 该虚拟地址即是应用程序启动过程中进程所访问的虚拟页面的地址,对于该步骤204来说,获取预设启动时间段内进程发生的页面错误的虚拟地址可以获知在应用程序启动过程中哪些虚拟地址未被映射到有效的物理地址中。

[0068] 需要说明的是,由于每次应用程序启动过程中发生的页面错误并不是完全相同的,因此,需要对多次应用程序的启动过程进行分析,以获取进程加载的映像文件列表和 Page Fault 发生时的虚拟地址。本实施例的步骤 201-204 仅以对单次启动的日志文件进行分析,并获取映像文件列表和 Page Fault 发生时的虚拟地址的过程为例进行说明,在实际应用中,可以同时多个日志文件进行分析,并获取映像文件列表和 Page Fault 发生时的虚拟地址,或,对多个日志文件顺序进行分析,并获取映像文件列表和 Page Fault 发生时的虚拟地址,对多个日志文件的分析顺序可以有多种,具体可以由技术人员设置,本实施例不做具体限定。

[0069] 205、根据该映像文件列表和该虚拟地址计算每个该虚拟地址对应的文件偏移量;

[0070] 本领域技术人员可以获知,对于映像文件列表中的单个映像文件来说,映像文件的起始地址减去 Page Fault 发生时的虚拟地址,得到相对虚拟地址,并通过相对虚拟地址获得文件偏移量,其中,该相对虚拟地址与文件偏移量是分段线性关系,该分段线性关系由映像文件(PE 文件格式)进行设置,本实施例不做赘述。

[0071] 该实施例与现有技术的区别在于,并不直接加载映像文件列表中的整个映像文件,而是将加载的的粒度更加细化,仅加载映像文件中的在应用程序启动过程中将会发生页面错误的文件块,以极少量的冗余数据换取 I/O 效率的大幅提升。

[0072] 206、根据该每个该虚拟地址对应的文件偏移量,对文件块进行合并,得到该应用程序对应的预取数据;

[0073] 具体地,根据该每个该虚拟地址对应的文件偏移量,将距离小于预设比特数的文件块进行合并操作,得到该应用程序对应的预取数据。优选地,该预设比特数可以为 64KB,如果两个文件块的距离小于 64KB,则将该两个文件块进行合并操作,通过对多个文件块的合并操作,得到该应用程序对应的预取数据。该预设比特数还可以有其他的数值,其具体数值可以由技术人员进行设置。

[0074] 本领域技术人员可以获知,一次磁盘 I/O 的大小在 WIN7 上可达到 2M,在 XP 上可达到 1M,而一次 Hard Fault 引入的 I/O 大小只有 16KB 或者 32KB,因此,在本实施例中,通过将相邻的文件块进行合并操作,可以大幅度的提高磁盘 I/O 的效率。

[0075] 表 1 是应用程序启动过程中一个模块的文件块合并后的示例。

[0076] 表 1

[0077]

文件偏移范围	大小 (KB)	页面数量	所属的段
[400,1183FF]	6964	1741	.text
[873400,8F03FF]	500	125	.text
[915400,92C3FF]	92	23	.text
[954400,9F73FF]	652	163	.text
[A8AC00,BF6BFF]	1456	364	.rdata

[0078] 表 1 中,文件偏移范围是指合并前文件块的文件偏移范围,大小是指合并后的文

件块大小,页面数量是指合并的页面数量,所属的段是该文件所属的段,由表 1 可以看出,对于各个属于不同段的文件来说,合并后的文件块大小(即是一次磁盘 I/O 的大小)远大于合并前每次 Page Fault 时所发生的的磁盘 I/O 大小(16KB 或 32KB)。

[0079] 优选地,该步骤 201-206 的获取预取数据的过程可以在系统空闲或应用程序未启动时进行,在不影响系统其它应用程序运行的情况下,有效利用了系统效率。

[0080] 207、当接收到启动应用程序的指令时,加载该应用程序对应的预取数据,该预取数据包括该应用程序启动过程中发生的页面错误对应的文件块;

[0081] 当接收到启动应用程序的指令,在应用程序运行其他代码前加载该应用程序对应的预取数据,然后再执行步骤 208,等到应用程序真正访问这些页面时,将不再发生页面错误,尤其是耗费时间的硬页面错误(Hard Fault),从而加快了速度。

[0082] 优选地,预取数据可保存在一个 DB 文件中,当接收到启动应用程序的指令时,读取该 DB 文件,以加载该应用程序对应的预取数据。

[0083] 208、加载该应用程序的代码,以便启动该应用程序。

[0084] 本领域技术人员可以获知,该步骤 208 与现有技术中的加载方法同理,在此不再赘述。

[0085] 需要说明的是,本实施例提供的应用程序启动方法,适用于任何应用程序,并且几乎不需要对现有代码做修改。而本实施例提供的步骤 201-206 的预取数据的过程,也不限于应用程序的启动过程,还可以用于任何提高应用程序性能的场景。

[0086] 优选地,当进行了步骤 207-208 后,应用程序已经启动,可以通过对本次应用程序启动过程的检测,由于应用程序启动过程中最耗费时间的为发生 Hard Fault,则查看应用程序启动过程中发生的 Hard Fault 数量,如果 Hard Fault 数量与历史启动过程相比减少或者消失,则说明通过步骤 201-206 获取的预取数据包含了应用程序所要访问的大部分页面。下表 2 给出是应用程序启动过程中一个主要模块发生的 Hard Fault 数量对比。可以看出,采用本发明的技术方案后,发生 Hard Fault 次数大大减少或消失。

[0087] 表 2

[0088]

	WIN7 上 Hard Fault 次数	XP 上 Hard Fault 次数
未预取	423	330
预取后	35	0

[0089] 图 3 是现有技术应用程序启动时间与采用本发明实施例的应用程序启动方法的启动时间的对比示意图。由图 3 可以看出,现有技术应用程序启动时间 T2 中包括硬页面错误时间和加载应用程序代码时间,采用本发明实施例的应用程序启动方法的启动时间 T1 包括加载预取数据时间和加载应用程序代码时间, T1 远远小于 T2,采用本发明实施例的应用程序启动方法的启动时间中不包含硬页面错误时间,大大改善了启动的速度。

[0090] 本实施例提供的方法,通过在应用程序启动时,先加载所述应用程序的历史启动过程中发生的页面错误对应的文件块,再加载应用程序代码,大大减少了应用程序启动过程中页面错误发生的次数,减轻了程序载入过程中频繁交换内存页面与交换文件的现象,

提高了应用程序的启动速度和启动效率。进一步地,通过将相邻的文件块进行合并操作,可以大幅度的提高磁盘 I/O 的效率。

[0091] 为了进一步说明本发明的有益效果,下面从现有技术应用程序启动过程和采用本发明实施例的应用程序启动方法的启动过程之间的对比为例进行说明:

[0092] 表 3 是现有技术应用程序启动过程与采用本发明实施例的应用程序启动方法的启动过程中一个主要模块的磁盘 I/O 参数的对比。

[0093] 表 3

[0094]

	磁盘 I/O 大小 (MB)	磁盘 I/O 次数	磁盘 I/O 时间 (ms)
现有技术	8.46	387	2013
采用本发明方案	12.094	54	207

[0095] 通过表 3 中的参数比较可知,采用本发明方案后,该应用程序主要模块的磁盘 I/O 耗费时间从 2s 左右降低到 0.2s 左右,磁盘 I/O 的大小从 8M 增加到 12M 左右,次数从 387 降到 54 次,说明通过对文件块的合并和预取数据,虽然增加了磁盘 I/O 的大小,但是降低了磁盘 I/O 次数和时间,从而降低了应用程序启动过程中的总的 I/O 时间。

[0096] 表 4 是现有技术应用程序启动时间与采用本发明实施例的应用程序启动方法的启动时间的对比。

[0097] 表 4

[0098]

	应用程序在 XP 环境下的启动时间(s)	应用程序在 windows7 环境下的启动时间(s)
现有技术	5.28	6.43

[0099]

采用本发明方案	2.38	3.69
---------	------	------

[0100] 其中,为了能够减少干扰,本启动时间是指在不使用其他优化技术的情况下的启动时间。通过表 4 中的启动时间比较可知,采用本发明方案后,在不同的操作系统,如 XP 环境或 windows7 环境中,采用本发明方案后,应用程序启动所用时间比采用先用技术所用的启动时间减少了 50 ~ 60% 左右,

[0101] 图 4 是本发明实施例提供的一种应用程序启动装置的结构示意图。参见图 4,所述装置包括:

[0102] 预取数据加载模块 401,用于当接收到启动应用程序的指令时,加载所述应用程序对应的预取数据,所述预取数据包括所述应用程序的历史启动过程中发生的页面错误对应的文件块;

[0103] 代码加载模块 402,用于加载所述应用程序的代码,以便启动所述应用程序。

[0104] 所述装置还包括:

[0105] 预取数据获取模块 403,用于分析所述应用程序的历史启动过程,获取所述历史启动过程中应用程序对应的预取数据。

[0106] 所述预取数据获取模块 403 包括:

[0107] 获取单元,用于获取所述应用程序的历史启动过程中加载的映像文件列表和发生的页面错误 Page Fault 的虚拟地址;

[0108] 偏移量计算单元,用于根据所述映像文件列表和所述虚拟地址计算每个所述虚拟地址对应的文件偏移量;

[0109] 预取数据获取单元,用于根据所述每个所述虚拟地址对应的文件偏移量,对文件块进行合并,得到所述应用程序对应的预取数据。

[0110] 所述获取单元包括:

[0111] 日志获取子单元,用于获取所述应用程序的日志文件;

[0112] 时间段确定子单元,用于将所述日志文件中从用户界面进程开始时间到显示前景窗口时间之间的时间段确定为预设启动时间段;

[0113] 映像文件列表获取子单元,用于获取所述预设启动时间段内进程加载的映像文件列表;

[0114] 虚拟地址获取子单元,用于获取所述预设启动时间段内进程发生的页面错误的虚拟地址。

[0115] 所述预取数据获取单元具体用于根据所述每个所述虚拟地址对应的文件偏移量,将距离小于预设比特数的文件块进行合并操作,得到所述应用程序对应的预取数据。

[0116] 需要说明的是:上述实施例提供的应用程序启动装置在应用程序启动时,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,上述实施例提供的应用程序启动装置与应用程序启动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。

[0117] 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。

[0118] 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。

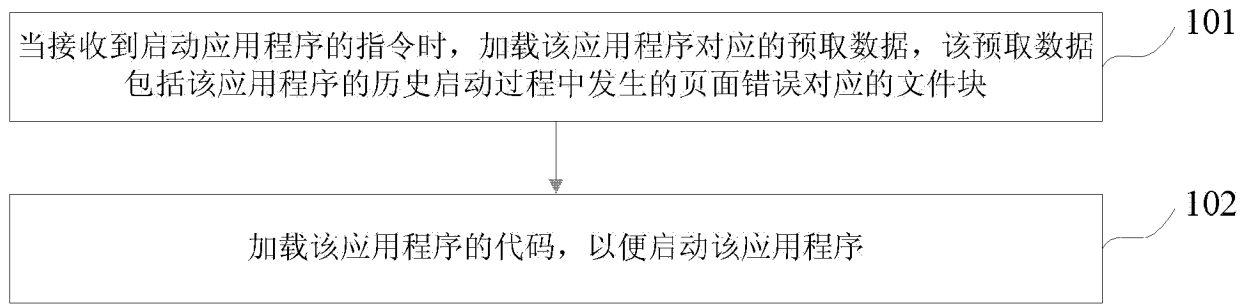


图 1

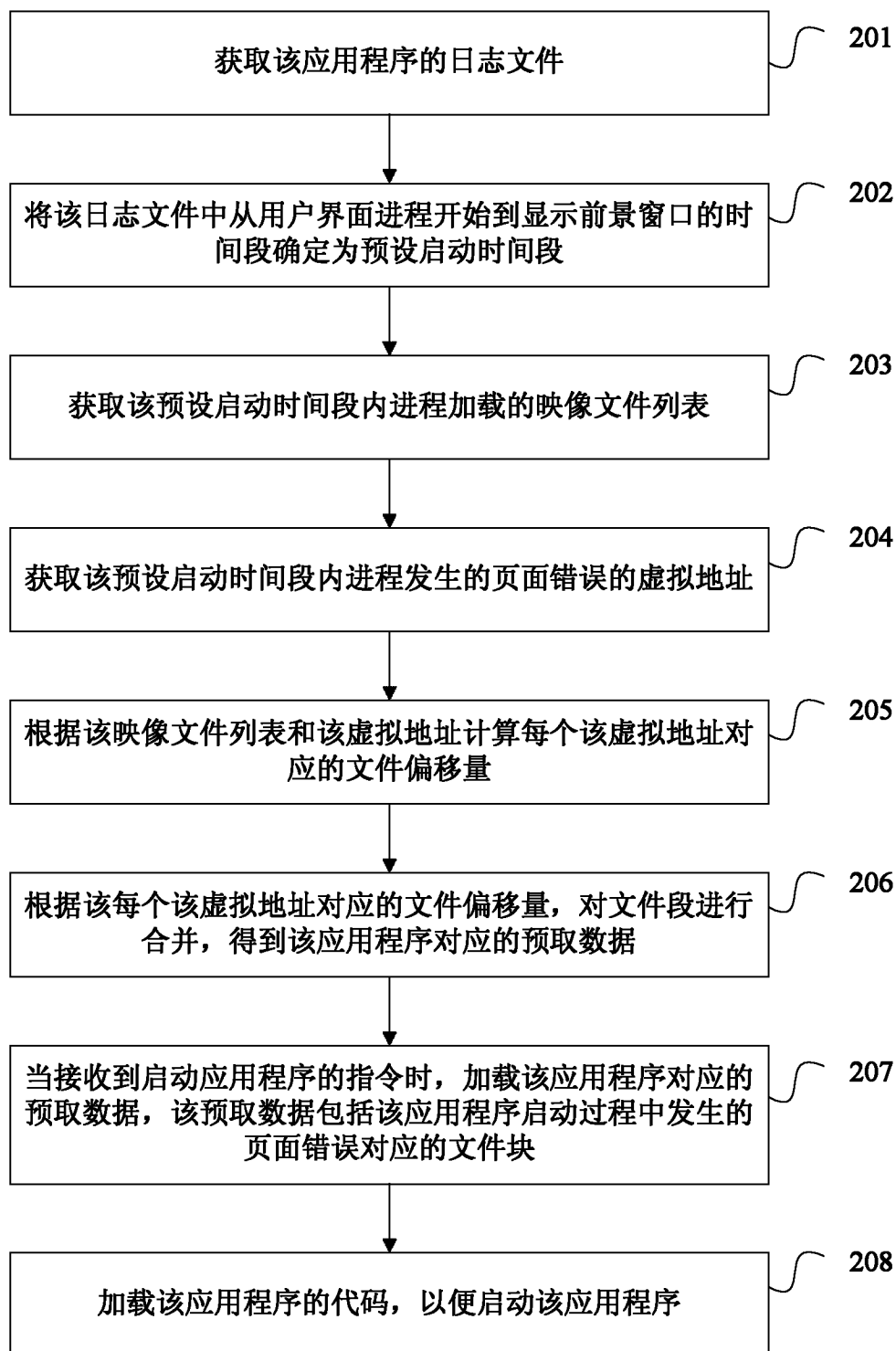


图 2

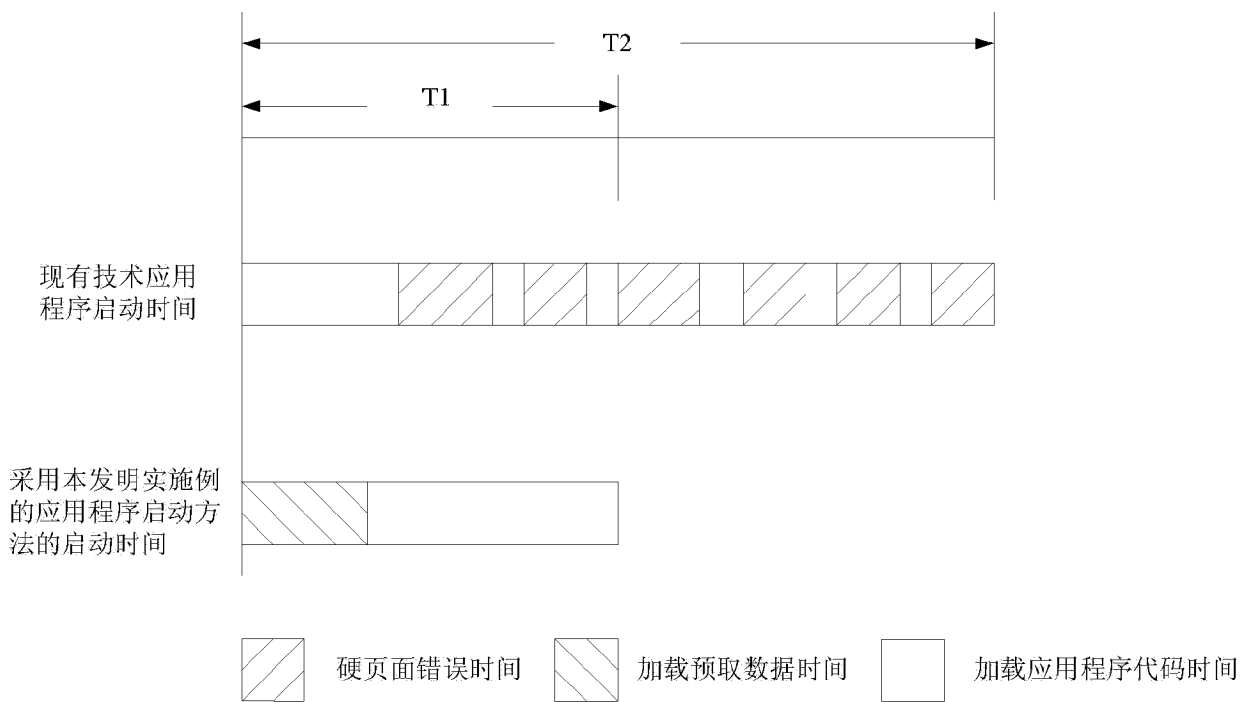


图 3

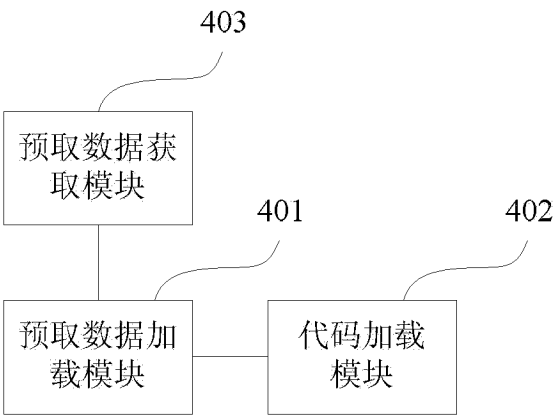


图 4