



(12)发明专利

(10)授权公告号 CN 103635876 B

(45)授权公告日 2018.01.02

(21)申请号 201280030252.8

(22)申请日 2012.06.18

(65)同一申请的已公布的文献号

申请公布号 CN 103635876 A

(43)申请公布日 2014.03.12

(30)优先权数据

13/163,745 2011.06.20 US

(85)PCT国际申请进入国家阶段日

2013.12.19

(86)PCT国际申请的申请数据

PCT/US2012/043033 2012.06.18

(87)PCT国际申请的公布数据

W02012/177577 EN 2012.12.27

(73)专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72)发明人 J·C·斯帕德林 G·亨特

A·F·托玛斯 S·马耶

(74)专利代理机构 上海专利商标事务所有限公
司 31100

代理人 顾嘉运

(51)Int.Cl.

G06F 9/06(2006.01)

G06F 9/30(2006.01)

G06F 12/02(2006.01)

(56)对比文件

US 2005081107 A1,2005.04.14,说明书第
0011,0169-0176,0233-0236段.US 2005081107 A1,2005.04.14,说明书第
0011,0169-0176,0233-0236段.US 2007011658 A1,2007.01.11,第0031段,
图3.

CN 101473301 A,2009.07.01,

US 2008163176 A1,2008.07.03,

审查员 张苗

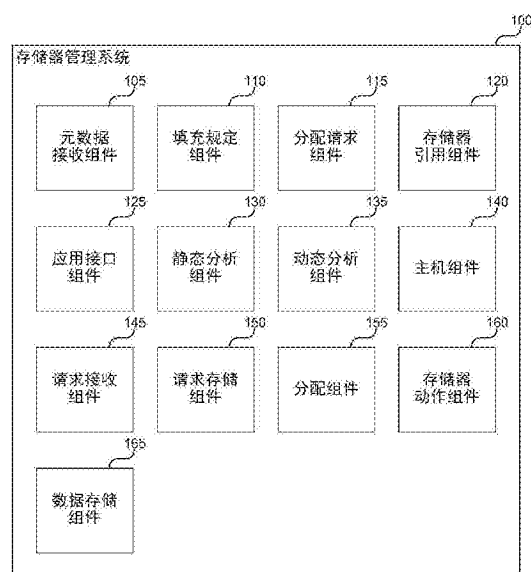
权利要求书2页 说明书15页 附图8页

(54)发明名称

用于未经修改的应用的存储器管理模型和
接口

(57)摘要

本文描述了一种存储器管理系统,它从各应用接收描述存储器正被如何使用的信息并且允许应用主机对使用存储器的应用请求施加更多控制。该系统提供应用存储器管理应用编程接口(API),该API允许应用指定与存储器分配有关的有助于稍后管理存储器的更多信息。该系统还提供以下能力:静态地和/或动态地分析传统应用以便向未被修改成与该系统一起工作的各应用给予参与更有效的存储器管理的一些能力。该系统提供应用主机改变以利用各应用所提供的信息并使用该信息和该应用的存储器使用中的挂钩来更有效地管理存储器。由此,该系统提供用于管理存储器的新模型,它改进了应用主机行为并允许各应用更有效地使用计算资源。



1. 一种用于分析未被专门设计成提供存储器分配信息的应用并提取存储器分配信息的计算机实现的方法,所述方法包括:

对所述应用执行分析以确定所述应用做出的存储器分配,其中所述分析包括静态地分析应用二进制码或其它应用代码以确定所述应用如何使用存储器,以及动态地分析正在运行的所述应用以收集与所述应用对存储器的使用相关的附加信息;

将所述应用挂钩以便在该应用内的该应用分配存储器的各位置处提供分配信息,其中所述挂钩包括重定向所述应用二进制码以便在所述应用二进制码中的一点处截取或扩充应用的标准存储器相关行为;以及

从所述应用接收分配存储器的请求以及经挂钩的应用代码所提供的相关联的分配信息,

其中,前述步骤由至少一个处理器来执行。

2. 如权利要求1所述的方法,其特征在于,检测所述应用包括在所述应用的二进制模块上检测指定所述应用是否参与增强的存储器模型的信息。

3. 如权利要求1所述的方法,其特征在于,检测所述应用包括确定先前高速缓存的应用分配信息是否可用。

4. 如权利要求1所述的方法,其特征在于,执行分析包括对所述应用应用静态二进制分析以标识应用二进制模块中的存储器相关行为。

5. 如权利要求1所述的方法,其特征在于,执行分析包括对所述应用应用动态分析以标识在所述应用的运行时的存储器相关行为。

6. 如权利要求1所述的方法,其特征在于,执行分析包括标识对用于分配、访问或释放存储器的一个或多个特定的主机应用编程接口 (API) 的调用。

7. 如权利要求1所述的方法,其特征在于,执行分析包括标识所述应用代码的存储器相关区域附近的分配元数据。

8. 如权利要求1所述的方法,其特征在于,将所述应用挂钩包括提供调用用于提供分配元数据的分配函数而不是调用所述应用最初调用的不带有元数据的标准分配函数的应用挂钩。

9. 如权利要求1所述的方法,其特征在于,接收所述分配请求包括由主机接收所述请求,所述主机能够确定所述分配信息来自未被专门设计成提供分配信息的应用。

10. 一种用于向应用主机提供对未被设计成提供分配信息的软件应用中的对存储器的分配和使用的更多控制的计算机系统,所述系统包括:

被配置成执行包含在以下组件内的软件指令的处理器和存储器;

元数据接收组件,所述元数据接收组件接收与存储器分配相关联的信息,所述信息描述在所述应用主机必须释放所述存储器分配的情况下所述软件应用是否能恢复所述存储器分配的内容;

应用接口组件,所述应用接口组件提供所述应用和主机之间的通信接口以协商对存储器分配的管理;

静态分析组件,所述静态分析组件静态地分析应用二进制码或其它应用代码以确定所述应用如何使用存储器;

动态分析组件,所述动态分析组件动态地分析正在运行的应用以收集与所述应用对存

储器的使用相关的附加信息;以及

主机组件,所述主机组件包括所述应用在其中运行的环境并提供对由所述系统提供的存储器管理器的访问,其中所述静态分析组件和动态分析组件向所述主机提供应用存储器使用信息。

11.如权利要求10所述的系统,其特征在于,所述应用接口组件包括被所述应用用来请求存储器分配并指定关于所述分配的信息的一个或多个函数或基类,以及被所述主机用来使用由所述应用提供的用户定义的函数来与所述应用的存储器交互的函数或基类。

12.如权利要求10所述的系统,其特征在于,所述应用接口组件提供通过静态和/或动态分析来确定的经插装的应用代码之间的交互,并与所述主机交互以提供分配信息。

13.如权利要求10所述的系统,其特征在于,所述静态分析组件分析二进制代码、中间代码或应用的其它经编译或可运行版本。

14.如权利要求10所述的系统,其特征在于,所述静态分析组件对所述应用二进制码进行插装以接收信息或截取所述应用的特定的存储器相关动作。

用于未经修改的应用的存储器管理模型和接口

[0001] 背景

[0002] 计算机系统上的存储器管理指的是多个应用和操作系统对存储器的使用达成一致的方式。虽然每一计算机系统具有固定量的物理随机存取存储器(RAM)或其他存储器,但操作系统可向各应用和各操作系统组件呈现表示与物理存储器不同的存储器大小的虚拟存储器。在一些情况下,虚拟存储器允许操作系统约束每一应用访问存储器的特定部分以阻止一个应用通过无意地或有意地修改另一应用的存储器来干扰该另一应用的操作。操作系统一般提供用于响应于应用和操作系统组件请求来分配并释放存储器的一个或多个功能。操作系统可向应用提供存储器池,该应用可从该池中分配存储器块。如果一应用使用或者一组应用一起使用比所安装的物理存储器的量更多的虚拟存储器,则操作系统可以在被称为分页或盘交换(即,向盘存储和从中取回存储器的各页)的过程中使用较慢的基于盘的存储来通过交换文件扩展存储器的外观大小。

[0003] 除了所提供的分配和释放功能之外,操作系统具有对各应用如何使用存储器的非常少的洞察。许多计算设备包含围绕存储器的特定限制。例如,移动计算设备可包括与在台式计算机系统上通常可用的存储器量相比小得多的存储器量(或系统可能想要解除某存储器的动力以降低能耗),从而对该设备造成与多少应用可同时运行、每一应用可请求/消费多少存储器等等有关的限制。在特定计算系统内主存应用代码的其他计算环境也可对该环境的存储器使用施加限制或上限。主机,如VMware和MICROSOFT™虚拟PC、系统管理程序、操作系统以及其他,可被分配有限的资源。在所有这些情况中,有效的存储器管理变得更加引人注目。

[0004] 新计算平台引入新技术或重新要求旧技术来解决各应用之间共享的有限存储器的问题。例如,移动电话操作系统可以创建每一应用的存储器快照,使得在该应用不在前台(例如,被主动使用)时,操作系统将它关闭并将该应用的存储器的镜像储存在较慢存储上(例如,闪存或其他存储)。在该应用被选择时,操作系统将所存储的镜像重新加载到存储器并开始该应用。该应用可能甚至不知道它被关闭过。虽然这些技术是有用的,但操作系统然而将遭受该应用对使用存储器的不透明请求的损害。当前做出的与动态存储器使用有关的大多数决策是基于在运行时期间找到的信息来做出的。这样的信息的示例包括对所分配的存储器段的引用的大小和数量。这一信息随后可被用来确定哪些分配将被分页到盘,被高速缓存到高性能存储器,或被某种自动存储器管理系统来释放。不幸的是,任何平台都受到可能许多年的传统应用的限制,所以在与存储器管理一样宽广的区域中采用新模型是困难的。

[0005] 概述

[0006] 本文描述了一种存储器管理系统,它从各应用接收描述存储器正被如何使用的信息并且允许应用主机对使用存储器的应用请求施加更多控制。当今,除了应用已做出了多少存储器请求和每一请求所请求的存储器大小之外,应用主机对该应用的存储器使用知之甚少。然而,应用主机不知道每一存储器分配的目的,哪些存储器分配将被很快使用,在应用主机需要更多存储器的情况下哪些存储器分配将被容易地重新创建,哪些存储器分配暂

时不会被使用并且因此可被分页到盘而不影响该应用的性能,等等。不幸的是,虽然应用主机的任务是做出这些类型的决策,但应用拥有与有效做出这些决策有关的最多信息。

[0007] 存储器管理系统用若干方式克服了这些问题。首先,该系统提供应用存储器管理应用编程接口(API),该API允许该应用指定与存储器分配有关的有助于稍后管理存储器的更多信息。该API还可向应用主机提供通知该应用何时需要存储器以及在需要时在没有应用交互的情况下抢先释放并重新创建存储器分配的能力。其次,该系统提供以下能力:静态地和/或动态地分析传统应用以便向未被修改成与该系统一起工作的各应用给予参与更有效的存储器管理的某种能力。第三,该系统提供应用主机改变以利用各应用所提供的信息并使用该信息和该应用的存储器使用中的挂钩(hook)来更有效地管理存储器。由此,本存储器管理系统提供用于管理存储器的新模型,它改进了应用主机行为并潜在地允许各应用更高效地使用计算资源。

[0008] 提供本概述是为了以简化的形式介绍将在以下详细描述中进一步描述的选择的概念。本概述并不旨在标识所要求保护主题的关键特征或必要特征,也不旨在用于限制所要求保护主题的范围。

[0009] 附图简述

[0010] 图1是示出在一个实施例中的存储器管理系统的各组件的框图。

[0011] 图2是示出在一个实施例中的存储器管理系统的操作环境的框图。

[0012] 图3是示出在一个实施例中的软件应用内的存储器管理系统请求分配和使用存储器的处理的流程图。

[0013] 图4是示出在一个实施例中的主机内的存储器管理系统接收分配和使用存储器的应用请求的处理的流程图。

[0014] 图5是示出在一个实施例中的存储器管理系统分析未被专门设计成提供存储器分配信息的应用的处理的流程图。

[0015] 图6是示出在一个实施例中的存储器管理系统静态地分析应用并提供增强的存储器信息的清单的处理的流程图。

[0016] 图7是示出一个实施例中的存储器管理系统响应于检测到的存储器压力来采取与存储器有关的动作的处理的流程图。

[0017] 图8是示出一个实施例中的存储器管理系统激活其存储器先前已被主机修改的应用的处理的流程图。

[0018] 详细描述

[0019] 本文描述了一种存储器管理系统,它从各应用接收描述存储器正被如何使用的信息并且允许应用主机对使用存储器的应用请求施加更多控制。当今,除了应用已做出了多少存储器请求和每一请求所请求的存储器大小之外,应用主机对该应用的存储器使用知之甚少。然而,应用主机不知道每一存储器分配的目的,哪些存储器分配将被很快使用,在应用主机需要更多存储器的情况下哪些存储器分配将被容易地重新创建,哪些存储器分配暂时不会被使用并且因此可被分页到盘而不影响该应用的性能,等等。不幸的是,虽然应用主机的任务是做出这些类型的决策,但应用拥有与有效做出这些决策有关的最多信息。这一冲突当今是通过应用主机提供基本水平的功能并猜测要采取哪些动作来解决的。在许多情况下,应用主机可能恰好在应用需要存储器之前将该存储器分页到盘,或者应用主机可能

花费大量努力来管理对应用不重要的存储器。

[0020] 本存储器管理系统用本文进一步讨论的若干方式克服了这些问题。首先,该系统提供应用存储器管理应用编程接口(API),该API允许该应用指定与存储器分配有关的有助于稍后管理存储器的更多信息。该API还可向应用主机提供通知该应用何时需要存储器以及在需要时在没有应用的交互的情况下抢先释放并重新创建存储器分配的能力。其次,该系统提供以下能力:静态地和/或动态地分析传统应用以便向未被修改成与该系统一起工作的各应用给予参与更有效的存储器管理的某种能力。第三,该系统提供内核级操作系统(或主机)改变以利用各应用所提供的信息并使用该信息和该应用的存储器使用中的挂钩来更有效地管理存储器。因而,本存储器管理系统提供用于管理存储器的新模型,它改进了应用主机行为并潜在地允许各应用更高效地使用计算资源。如本文所描述的,应用主机可指的是执行应用的操作系统或另一类型的主机(例如,本身在操作系统上或虚拟化子系统上运行的应用),如SILVERLIGHT™、.NET所提供的运行时,本机Win32主机,或VMware和虚拟PC所提供的其他主机或虚拟机。以上介绍的这三个区域中的每一个在以下各节中进一步详细描述。

[0021] 经修改的应用

[0022] 在许多情况下,软件开发人员修改软件应用以与存储器管理系统进行交互是可能的。对于主动开发的应用,软件开发人员可选择采用存储器管理系统来获得它所提供的益处,或可由该应用在其上操作的特定平台来托管以采用存储器管理系统。在许多情况下,应用可能保持该应用不太可能使用的所分配的存储器。例如,在用户从该应用的用户界面的一部分转移到另一部分时,该应用可常常保持来自先前界面的信息以防用户重新访问该界面。当今,该存储器只在需要时被主机视作其它主动使用的存储器。存储器管理系统提供了供应用将诸如此类情形通知给主机以使得这类存储器可被降低优先级的方式。作为响应,主机可将这类存储器选为分页的良好候选或做出由于来自应用的添加的信息而更高效的其它存储器管理决策。

[0023] 在一些实施例中,存储器管理系统提供应用编程模型或框架,该应用编程模型或框架使存储器管理器能够做出关于优化运行时期间的存储器使用的智能决策。这可通过利用接收用于分配和填充被指定给应用请求的任何给定存储器对象的存储器分配的元数据和动作两者的应用编程模型/框架来完成。元数据提供应用想要传递给主机的描述存储器分配的特性或目的的信息,诸如存储器的优先级、正被分配的存储器的数量、从头重新创建存储器内容的容易度(例如,这些内容可以从文件加载或者可由算法来计算)、访问频率、应用可能多久以后使用存储器,等等。用于分配和填充存储器的动作可以向主机提供足够的信息以使得能够释放并且随后应应用的请求重新创建被释放的存储器。通过允许开发者指定用于填充存储器的元数据和动作,存储器管理系统能够优化与应用所指定的所需使用一致的存储器使用。

[0024] 存储器管理系统提供给应用的API为应用开发者提供了一种通过应用框架或编程模型来指定描述对存储器分配的使用的元数据的手段。另外,该API允许应用框架规定开发者利用标准手段来通过公知的功能填充或修改存储器。这允许存储器管理系统出于性能原因或出于其它目的不失时机地填充存储器,以便在低存储器可用性时段期间(即,在释放存储器的机会成本大于稍后重新分配和重新填充存储器的成本时)释放存储器。

[0025] 优化存储器使用可涵盖本领域内已知的许多技术,但将一般意味着为了性能或大小而优化。在性能的情况下,优化可能意味着在填充存储器的手段可用的情况下允许进行尚不需要的存储器分配。这在当前CPU使用率较低且应用空闲的情况下可以是合乎需要的。在某些情况下,应用分配存储器的请求可变成笔记,主机存储该笔记以供将来参考,而不在此时进行任何分配。随后,当应用通过API请求使用存储器时或者在主机确定满足该请求的合适时间时,存储器管理系统实际分配所请求的存储器。在大小的情况下,优化可能意味着减少存储器占用或基于当前分配的存储器和未来存储器需求来做出决策。

[0026] 应用和主机之间的实际接口可采取本领域内的普通技术人员将会认识到的各种形式。例如,应用可以为每一种类型的分配提供分配函数并在分配请求中向主机传递针对该分配的指针或引用。当主机准备好执行分配时,主机调用所提供的分配函数并且应用使用常规存储器分配函数来创建存储器。类似地,应用可传递对其它函数的引用以使得主机能够请求释放存储器、移动存储器、将存储器内容切换至不同类型的存储,等等。相同的概念还可用于分配,当应用向操作系统请求存储器时,操作系统可基于多个因素来延迟分配。当操作系统准备好分配时,(用所分配的存储器)回调函数引用或者引发一事件(或类似机制)。该接口还可接收元数据,诸如存储其大小(可以不同于所请求的大小)、优先级、高速缓存偏好、可分页性、如何填充存储器、对存储器的依赖或引用、是否更新存储器,等等。在一些实施例中,系统提供分配从中导出以定义每一种类型的存储器分配的存储器接口类。该类可包括用于检索应用专用分配函数的GetPointer函数或用于检索执行其它存储器处理任务的函数的其它GetX函数。另选地或另外地,应用可以按传统方式进行分配并且然后调用RegisterPointer函数,该函数向主机注册所分配的存储器并指定要与所分配的存储器相关联的本文描述的附加信息。以下伪代码提供了应用可使用的一个存储器类的示例。

```
CMemChunk // 用于所有存储器分配的鸡肋
```

```
{  
    <分配的全局列表>  
    ...  
};
```

```
[0027] CMyMemory : public CMemChunk
```

```
{  
    分配(大小);  
    填充() { <如何填充存储器> } // 覆盖  
    属性 <例如, 优先级、可分页性等>  
};
```

[0028] 另选地或另外地,开发者可使用源注释语言(SAL)或其它标记来将本文描述的框架引入应用代码,以标识现有存储器分配并且指定与每一次分配、访问或其它存储器交互相关的附加参数和元数据。

[0029] 在一些实施例中,存储器管理系统可以在单个应用中操作并且不与内核或其它主

机共享。应用能够受益于改进的存储器管理,所述改进的存储器管理的自己的内置存储器管理器能够通过使用本文描述的框架所提供的附加信息来执行。在某些情况下,主机然后可提供注册函数,应用可调用该函数来获得跨应用益处并允许主机也使用明确定义的存储器分配和使用。作为示例,该系统可以在恶意软件扫描之前通知应用,以使得该应用能够卸载任何较不相关的存储器以加速扫描。作为另一示例,应用可以在CPU去往空闲状态之前预先分配存储器,以使得应用能够在用户做了某件事并且CPU恢复执行时快速响应事件(增强对功率状态改变的响应性)。

[0030] 未经修改的应用

[0031] 在软件开发者不可能将软件应用修改成与存储器管理系统交互或者该系统被实现为与未经修改的应用(即,未被专门设计成与该系统一起工作的应用)一起操作的情况下,该系统仍然可能提供存储器管理益处。为此,该系统收集描述应用如何使用存储器自身的的信息(例如,基于静态分析和/或剖析应用—使应用运行、截取存储器分配以及查看跨应用运行时的使用)。该信息在确定性能特性时是有用的,并且可由应用主机的存储器管理器来智能地使用。可使用该信息的方式的示例包括智能垃圾收集、到盘的智能分页、到更高性能的存储器高速缓存的智能高速缓存以及甚至警告用户应用可能遇到的潜在存储器限制。

[0032] 利用对二进制码的静态分析、对二进制码的行为的运行时分分析以及通过对二进制码进行插装(instrument),收集到关于二进制码的给定存储器分配中的任一个以及对这些分配的使用的附加信息是可能的。该信息然后可被用来导出围绕物理存储器中的分配的加载/卸载和定位的更智能的行为。存储器管理系统提供了用描述对分配本身的潜在或实际使用的元数据来自动注释应用存储器分配的手段。该分析可以自动地对二进制码静态地或在运行时期动态地执行,而不需要任何开发者交互或重新创作现有应用。一旦执行,该分析就可被系统高速缓存以使得主机操作系统知道将来如何处理应用。该分析还可被发布以供其它客户机发现,而不仅仅被本地地高速缓存。另外,信息可能被展示给用户以供可选的编辑,从而允许管理员或用户定制应用的元数据以及应用主机将如何处理应用的存储器分配。

[0033] 可能使用静态和动态分析来导出关于应用的存储器分配的附加信息,并且该信息然后可被用来帮助引导应用和系统存储器分配的总体管理。示例是在存储器正被触碰时;可能确定是否使用另一存储器段来帮助填充该存储器分配。如果该存储器分配然后取决于另一分配,则可指示该依赖关系,或者通过一位来标记该存储器分配即可,该位指示该存储器分配不是在没有输入的情况下生成的。静态分析可确定软件代码在何处触碰到存储器,如何使用存储器、如何填充存储器、代码路径有多常见(例如,它是否是单写/多读的(WORM))以及使用存储器的频率。动态分析可以对所有分配和/或访问进行插装(类似于剖析器)并且可捕捉系统环境对代码操作的影响、影响操作的用户设置以及不可用或难以静态地确定的其它数据。

[0034] 在一些实施例中,存储器管理系统在分析应用后输出应用的存储器使用的清单或其它描述。这允许系统高速缓存分析结果并重用结果以便将来执行应用。在应用二进制码第一次执行时(或者在如应用虚拟化的序列化的过程期间),操作系统可被设计成与系统一起执行该分析,并且然后存储该分析以便在每一次二进制码执行时使用。内核或其它主机可读取清单数据并通过描述应用如何使用存储器的附加信息来采取适当动作。应用使用可

随时间变化,以使得该清单或其他高速缓存有时可被动态更新。在某些情况下,该系统可以为每一分配分配整页存储器,以使得每一次存储器访问都触发页面错误,该页面错误允许内核控制如何使用存储器并提供此处进一步描述的对存储器的应用引用以及实际存储器分配之间的这种类型的间接性。

[0035] 在一些实施例中,该系统可提供关于应用存储器行为的报告。这是分析的另一种用途并且可帮助管理员做出关于应用的决策,诸如应用将在哪一个服务器或虚拟机上良好地运行。该系统还可提供对存储器使用的市场评级,以使得移动电话应用商店的用户例如能够在下载应用之前知道该应用将如何使用移动电话的可用存储器(例如,高度使用、低度使用等)。管理员还可使用该信息来基于收集到的关于存储器消耗的元数据进行跨各种系统的IT软件部署。

[0036] 存储器管理系统可使用从存储器使用分析中导出的信息来在系统空闲或未被充分利用时预先分配并且可能填充存储器。知道应用在将来对存储器的使用允许系统更高效地分配存储器并利用系统未被充分利用的时间来分配可以在系统处于更重负载时使用的存储器。这允许更重的负载具有对由于早先完成的工作而变得可用的更多处理或其它资源的访问权。

[0037] 主机修改

[0038] 存储器管理系统包括对内核或应用主机的修改以接收由应用提供的或通过对应用的分析来确定的关于存储器使用的附加信息。不像传统软件存储器管理,由于本文描述的元数据以及传递自应用的其它信息提供的对存储器使用的增加的洞察,内核可以做更多以便在没有应用知识或动作的情况下高效地管理存储器。内核可以稍后通知应用已采取什么动作或者管理存储器以使其在应用需要该存储器时就位,从而使得应用不知晓或不关心内核的存储器相关动作。内核然后可执行更好的分页(例如通过将较不重要的或不太可能使用的存储器卸载到较慢的盘存储),可以在出现存储器压力时释放存储器,并且可以采取其它动作来替使用该内核提供的操作环境的一个或多个应用管理存储器。例如,内核可以更好地分配存储器以实现较少的碎片。

[0039] 存储器是操作系统中的受限资源;因此,内核适当地跟踪在何处分配存储器以使其能够在需要时从应用恢复存储器是重要的。一个解决方案是在向应用分配存储器时使应用向存储器分配优先级。由此,当系统确定存储器较少时,内核可确定优先级最低的存储器在何处并解除对该存储器的分配或对该存储器进行分页,而不影响具有优先级较高的存储器的其它应用的执行。

[0040] 传统上,存储器的优先级及其到各应用的分发由内核确定。当系统以较少的存储器运行时,内核可以任意地释放被应用使用的具有较高优先级的存储器,由此使应用的性能中断或降级。而优先级较低的存储器本应首先被解除分配。存在与内核如何确定从哪一个应用释放存储器相关的有趣方面。一个解决方案是使应用在其本身中确定对于存储器资源的优先级次序是什么。存储器优先级方案要求应用分配在分配或解除分配时对应用分配或解除分配的存储器的优先级。由此,当存在存储器压力时,内核具有通过优先级排名的存储器地图,并且可以首先释放和定位优先级较低的存储器。或者,内核可以向应用发送具有需要被释放的分配的列表的通知。存储器管理系统可由操作系统来完全主存,或者可以在操作和应用之间协作。

[0041] 优先级模型可通过创建只要在做出存储器请求时就封装应用的存储器分配API来实现。这也可使未注册的应用受益。通过利用该API,应用和优先级由子系统来自动跟踪,而不使得应用主动管理存储器优先级。内核对象结合注册、计算、信令和存储器管理功能中的全部,以使得每一个应用都只需调用该对象(或者在未经修改的应用的情况下代表应用调用该对象)。

[0042] 一种替换解决方案是在内核之外具有主应用,该主应用被用来跟踪当前正在运行的各种应用。当应用请求存储器时,这些应用被注册到主应用。由此,当应用变成不响应时,主应用可确定是否清除该僵尸应用以回收存储器或者使该僵尸应用保持挂起以使得可以在需要存储器时将存储器返回给该应用。

[0043] 内核或主机可使用增强的存储器信息来更高效地做出各种存储器管理决策。例如,低功率设备的内核可选择关闭存储器排以节省电池寿命。该系统可清除或换出具有存储器排中的将被关闭的存储器的进程。系统还可整理跨若干排分布的存储器的碎片以取得可被关闭的一些空排。作为另一示例,内核可选择将一些存储器分配推送到其它设备(包括基于云的存储设施)。许多占用量较小的低功率操作系统不具有分页概念,因此在这种情况下可通过由另一设备/服务主存的代表性状态传输(REST)接口来支持存储器分配。以与滚动遍历数据库快照类似的方式,应用在存储器中不具有整个数据库,但将各数据库片段从数据库服务/服务器传递给该应用。该应用不具有它们正在查看快照而不是整个数据库的知识。能够确信地移走存储器或解除存储器分配允许内核/主机更好地管理存储器并且仍然维持应用对存储器可用性的期待。

[0044] 在一些实施例中,该系统与除了存储器之外的其它主机级对象或组件一起使用。例如,该系统可以在显示器关闭时关闭图形处理单元(GPU),或者在与各驱动程序相关联的硬件未被使用或将在某一时间内不使用时关闭各驱动程序。该系统还可用于系统休眠,以便首先整理存储器碎片并且然后将线性的存储器分配流传输到可以在系统被唤醒时容易被重新加载的盘。不像存储大小等于整个物理存储器的文件的传统休眠,该存储器管理系统可鼓励应用在休眠之前释放不需要的或容易恢复的存储器(或者可以为这些应用做这些事),并由此可存储较少的休眠数据量。除了存储器之外的其它被管理的项目可包括文件句柄、计时器、内核对象等。该系统可以就像存储器使用那样接收关于这些项目的使用信息(或者从静态和动态分析中确定该信息)。

[0045] 在一些实施例中,存储器管理系统接收与每一次存储器分配相关联的功率状态。例如,在功率级别X,系统可确定不再需要某些存储器分配。这可允许电池受限设备通过卸载在该功率级别不需要的任何存储器来切换至较低功率模式。例如,移动电话可以在存储器中保存足够的应用数据来响应电话呼叫或向用户警告新电子邮件,但可以卸载其它优先级较低的功能或应用。在某些情况下,在某些较低功率状态中应用可能只需在使用存储器之前验证指针有效性,但在较高功率状态中可具有对存储器的正常访问权。这可被提供为应用和用于任何给定平台的主机之间的合同保证。

[0046] 系统组件和操作环境

[0047] 图1是示出在一个实施例中的存储器管理系统的各组件的框图。系统100包括元数据接收组件105、填充规定组件110、分配请求组件115、存储器引用组件120、应用接口组件125、静态分析组件130、动态分析组件135、主机组件140、请求接收组件145、请求存储组件

150、分配组件155、存储器动作组件160和数据存储组件165。这些组件中的每一个都在此处进一步详细讨论。

[0048] 元数据接收组件105、填充规定组件110、分配请求组件115和存储器引用组件120一起构成系统100展示给应用的存储器框架。希望修改他们的应用代码以实现更高效的存储器管理的开发者可利用来自他们的应用的这些组件来创建允许主机或内核代表应用来更有效地管理存储器的应用。

[0049] 元数据接收组件105接收与每一次存储器分配相关联的信息,该信息向应用主机提供描述应用如何使用存储器的信息。例如,该元数据可指示对于应用而言应该可以多容易地访问分配或者应用计划多频繁地访问与该分配相关联的存储器。该元数据还可指示存储器内容有多难以生成,并由此对于应用或主机而言在内容被释放或分页到盘的情况下替换存储器内容有多难。元数据接收组件105可以在对存储器分配API的调用中接收元数据或者在存储器已经在用于提供元数据的后续API中被分配之后接收元数据。

[0050] 填充规定组件110接收描述如何填充特定存储器分配的内容的信息。存储器内容可来自各种源,诸如来自读取存储在盘上的文件内容、来自对输入数据执行一个或多个计算、来自用户输入、来自网络访问的信息(例如,数据库或因特网)等等。在一些实施例中,应用向主机传递存储器填充函数以使得主机能够调用该函数来在该主机确定的时刻填充存储器内容。为了高效地使用处理和其它资源,主机可选择延迟填充存储器直到资源未被充分利用或空闲。另外,主机还可自由地释放或解除先前填充的存储器以作他用,并且然后稍后在应用期望使用存储器之前重新分配并重新填充该存储器。接收到的元数据可提供主机用来知晓应用将何时使用存储器的信息,或者应用可以在每一次试图使用存储器之前通知主机以使得主机能够检查存储器的当前状态。

[0051] 分配请求组件115将来自应用的请求提交给主机以便基于接收到的元数据和填充规定来分配存储器。注意,尽管主机接收请求,但是是立即服务该请求以作为响应还是等待直到另一适当时刻是主机的自由。在极端情况下,主机可能未分配任何存储器直到存储器准备好被访问,以允许主机节省有限的资源直到应用需要存储器且必须为该应用分配存储器以使得能够执行其工作的最后可能时刻。分配请求组件115将所提交的请求存储在数据存储中,该数据存储由主机管理并且包括接收到的元数据和填充规定以便在稍后存储器管理动作中使用。

[0052] 存储器引用组件120在应用访问作为先前提交的分配请求的主体的存储器之前从应用接收指示。因为主机可使存储器不可用或延迟实际分配存储器直到它选择,所以应用需要确保该存储器在应用准备好使用它时可用的方法。存储器引用组件120服务于该目的,并且允许应用指示它准备好访问特定存储器分配。作为响应,主机可以向应用传递对实际存储器位置的指针(如果该存储器已经可用的话),或者可基于接收到的填充规定和元数据来创建和填充存储器(如果该存储器当前未被分配的话)。应用可发送释放存储器的指示以使得主机在存储器管理决策中自由地再次包括该存储器。

[0053] 应用接口组件125提供应用和主机之间的通信接口以协商对存储器分配的使用。该接口可包括被应用用来请求存储器分配并指定关于分配的信息的一个或多个函数或基类,以及被主机用来使用应用所提供的用户定义的函数来与应用的存储器交互的函数或基类。应用和主机之间的增强的通信允许主机具有对应用的存储器使用的比正常水平高得多

的可见性,并允许主机代表各种应用管理存储器以便更有效地共享有限资源。

[0054] 对于未被专门构建成使用存储器管理系统100的应用,应用接口组件125提供通过静态和/或动态分析来确定的任何插装的应用代码之间的交互,并以与以上讨论相似的方式与主机交互。在未被构建成使用系统100的应用的情况下,系统100可具有关于存储器分配的目的或其它规定的较少信息,并且可被限于系统100可通过对应用的静态和动态分析来发现的信息。然而,这样的分析仍然可发现可用于更有效地管理传统应用分配的存储器的元数据。在某些情况下,系统100能够自动检测应用正如何填充存储器并且能够在没有应用的显式协作的情况下执行上述相同种类的存储器内容按需填充。

[0055] 静态分析组件130静态地分析应用二进制码或其它应用代码以确定应用如何使用存储器。组件130可分析二进制代码、中间代码(例如,MICROSOFT™中间语言(IL)代码)或应用的其它经编译或可运行版本。静态分析在过去几年中进步显著,并且用于确定应用二进制码做了什么以及怎么做的许多技术在本领域内是公知的。存储器管理系统100使用这些技术来专门聚焦于应用分配和使用存储器的区域。组件130可以对应用二进制码进行插装以接收信息或截取应用的特定动作并且可以用新或另外的动作来替换所截取的动作。例如,如果组件130发现对存储器分配函数的调用,则组件130可收集可以从静态分析中获得的任何元数据并且调用可接收到元数据作为参数的分配函数版本。以此方式,主机就像应用被其开发者修改为提供元数据信息那样从应用接收这样的信息。同样,系统可确定应用如何填充或访问特定存储器分配,以使得可以向主机提供描述存储器引用的填充规定和信息。

[0056] 动态分析组件135动态地分析正在运行的应用以收集难以用静态分析来确定的关于该应用对存储器的使用的附加信息。应用经常包括阻碍静态分析的编程步骤(有意地或者只是因为这些步骤以阻碍静态分析的方式出现)。动态分析具有可用信息,诸如从外部组件接收到的响应的内容以及由应用使用的存储器的实际内容,对于这些内容在静态分析期间只能进行猜测或逼近。由此,动态分析可以找到在静态分析期间找不到的存储器使用和其它应用活动。组件135还可使用动态分析来确认静态分析的结果。动态分析组件135向主机提供所确定的信息以便进一步允许传统应用利用系统100提供的至少一些功能。

[0057] 主机组件140包括应用在其中运行的环境并提供对系统100提供的存储器管理的访问。取决于该平台,主机可包括各种软件实体。对于传统的桌面计算系统,主机是操作系统,该操作系统可包括内核或其它内核模式软件代码,诸如驱动程序。对于托管软件环境,主机可包括运行时环境,诸如MICROSOFT™.NET™运行时环境或MICROSOFT™SILVERLIGHT™运行时环境。web应用也可包括应用主机并且可以在沙箱环境中运行,该沙箱环境在其它服务器或桌面计算环境中运行。服务器计算系统可包括虚拟化硬件或软件并且可包括在比一个或多个操作系统内核更高的层面操作的管理程序。存储器管理系统100可被实现为各种层面的软件代码并由此不限于任何特定类型的主机。主机组件140表示负责与应用交互并管理用于系统100在其上运行的特定平台的资源的组件。

[0058] 请求接收组件145、请求存储组件150、分配组件155和存储器动作组件160一起构成被修改成实现本文描述的系统100的存储器管理器的组件。

[0059] 请求接收组件145从应用接收分配存储器的请求。每一个请求包括向主机提供描述应用如何使用所分配的存储器的信息的元数据。请求接收组件145可以从在特定主机上

运行的多个或所有应用接收请求,以使得主机具有描述如何使用存储器的大范围的信息。主机然后可排定在决定采取特定动作(例如,分页或释放存储器以减小存储器压力)时影响哪些分配的优先级。

[0060] 请求存储组件150存储接收到的请求及相关联的元数据信息以供在存储器管理决策制定期间后续使用。主机可以在接收到请求时满足或不满足分配请求。然而,无论主机是立即还是稍后满足请求,主机都与分配相关联地存储应用提供的信息,以使得主机稍后在存储器动作是适当的时候具有可用信息。在一些实施例中,主机在应用已经分配存储器后在与应用分开的通信中接收到元数据和其它信息。在这些情况下,主机可以在多个时刻或者在有用信息可用时存储分配信息。

[0061] 分配组件155执行分配存储器的实际动作,以满足接收到的一个或多个应用分配请求。分配组件155可分配来自应用堆或直接来自计算机系统中安装的可由主机访问和管理的物理存储器的存储器。在分配后,存储器被托付给特定应用并且无法由其它应用使用(除非这些应用通过典型的共享存储器技术来请求这样做)。分配组件155可以与页表和虚拟存储器一起工作以便将窗口提供到供分配的物理存储器中,并且所提供的存储器可由盘或其它存储通过交换文件来备份。

[0062] 存储器动作组件160执行管理设备上的存储器的动作并访问先前存储的接收到的请求信息以确定将受到所执行的动作的影响的一个或多个合适的存储器分配。动作可包括将存储器内容交换到交换文件、释放先前分配的存储器、请求应用减少其存储器占用、或者影响一个或多个应用使用的存储器的任何其它动作。通常,主机执行存储器管理动作的目的是处理诸如存储器不足(例如,存储器压力)等当前或即将发生的情况、收集足够的存储器来满足应用需求,等等。

[0063] 数据存储组件165存储被主机用来代表一个或多个应用管理存储器的信息。数据存储组件165可包括留给系统100使用的存储器、单独的闪存或盘存储或用于存储数据的其它设施。数据存储组件165还可包括被存储在每一个应用的存储器空间中的跟踪涉及该应用的分配的数据以及被主机用来代表该应用管理存储器的其它信息。

[0064] 其上实现存储器管理系统的计算设备可包括中央处理单元、存储器、输入设备(例如,键盘和定点设备)、输出设备(例如,显示设备)和存储设备(例如,盘驱动器或其他非易失性存储介质)。存储器和存储设备是可以实现或启用该系统的计算机可执行指令(如软件)来编码的计算机可读存储介质。此外,数据结构和消息结构可被存储或经由诸如通信链路上的信号等数据传输介质传送。可以使用各种通信链路,诸如因特网、局域网、广域网、点对点拨号连接、蜂窝电话网络等。

[0065] 该系统的实施例可以在各种操作环境中实现,这些操作环境包括个人计算机、服务器计算机、手持式或膝上型设备、多处理器系统、基于微处理器的系统、可编程消费电子产品、数码照相机、网络PC、小型计算机、大型计算机、包括任何上述系统或设备、机顶盒、片上系统(SOC)等中任一种的分布式计算环境等。计算机系统可以是蜂窝电话、个人数字助理、智能电话、个人计算机、可编程消费电子设备、数码相机等。

[0066] 该系统可以在由一个或多个计算机或其他设备执行的诸如程序模块等计算机可执行指令的通用上下文中描述。一般而言,程序模块包括执行特定任务或实现特定抽象数据类型类型的例程、程序、对象、组件、数据结构等等。通常,程序模块的功能可在各个实施例中

按需进行组合或分布。

[0067] 图2是示出在一个实施例中的存储器管理系统的操作环境的框图。该环境包括内核/主机210、一个或多个应用220、存储器处理框架230、所存储的分配元数据240、以及一个或多个工具250。内核/主机210管理一个或多个应用220所共享的存储器。存储器处理框架230提供应用220可调用以分配存储器并提供存储器分配元数据的一个或多个API。存储器处理框架230存储接收到的元数据240。工具250操作未被设计成在本机提供分配信息的各应用以通过静态和动态分析来提取这一信息。任何所提取的信息随后被存储作为分配元数据240。

[0068] 系统操作

[0069] 图3是示出在一个实施例中的软件应用内的存储器管理系统请求分配和使用存储器的处理的流程图。

[0070] 在框310开始,应用设置描述应用将如何使用存储器分配的分配元数据。根据本文描述的存储器管理系统,该应用可构建用于传递给分配函数的参数结构,调用用于提供分配元数据的API,或设置分配存储器的派生类的参数。分配元数据可包括存储器对于该应用的优先级、该应用计划有多频繁地使用所分配的存储器、该应用替换该分配的内容有多困难、以及与该应用对存储器的使用以及主机可如何操纵该存储器来有效地使用多个应用所共享的物理存储器资源有关的其它信息。

[0071] 在框320继续,该应用设置填充该存储器分配的内容的存储器填充函数。存储器填充函数可以访问来自文件的信息,执行一个或多个计算以生成储存在该存储器分配中的结果,或执行填充该存储器分配的其他动作。通过向主机提供存储器填充函数而非正好在分配之后调用填充函数,该应用向主机提供按需释放和重新填充存储器所需的信息或推迟存储器的分配和初始填充直至对该主机更适当的时间所需的信息。

[0072] 在框330继续,该应用调用主机所提供的管理跨多个应用上共享的物理存储器的分配接口,其中该应用经由该分配接口向主机提供设置分配元数据和存储器填充函数。该应用接口可包括操作系统API、运行时函数、或在应用和负责存储器管理的主机之间传递信息其他方式。该分配接口提供比当今存储器分配中通常所提供的信息更多的信息,并且向主机给予有助于更高效地管理计算设备上的存储器的信息。

[0073] 在框340继续,该应用响应于调用分配接口而接收引用,其中该引用担当该应用对所分配的存储器的后续使用的间接标识符。在接收到请求之后,主机可以或不立即分配存储器。另外,主机有时可按在应用使用存储器之前主机需要再次接收控制的方式修改该存储器(例如,释放它、将它分页到盘,等等)。因而,主机向该应用提供对存储器的引用,并且在该应用想要使用该存储器时,该应用提供该引用以得到对该存储器的直接访问(例如,指针)。

[0074] 在决策框350继续,如果该应用检测到该应用对所分配的存储器已使用完毕,则该应用完成,否则该应用在框360继续。在完成该应用之前可调用分配接口以解除分配(即,释放)先前分配的存储器。如果该应用没有使用该存储器,并且主机实际上没有分配该存储器,则这一动作可简单地清理主机的所存储的与该分配有关的条目并将控制返回给该应用。

[0075] 在决策框360继续,如果该应用检测到该应用中对所分配的存储器的使用,则该应

用在框370继续,否则该应用循环到框350以等待存储器访问或该存储器分配的目的的完成。该应用可在应用软件代码中访问存储器的任何位置处调用特定函数,或可将存储器分配封装在用对主机的适当调用来包装存储器访问的类中以使得存储器可用。

[0076] 在框370继续,该应用向主机请求直接访问存储器分配。如果主机已经分配了存储器,则主机向该应用返回可访问该存储器的指针。如果主机尚未分配存储器或已经解除分配并重用了该存储器,则主机响应于应用请求来分配存储器,调用接收到的填充函数来填充存储器内容,并随后将访问该存储器的指针或其他装置返回给该应用。

[0077] 在框380继续,该应用通过从主机接收到的存储器地址来访问存储器分配。该应用可修改存储器,从存储器读取信息,或执行其他典型的存储器操作。在该应用完成访问该存储器时,该应用可向主机指示该主机可再次执行使得该存储器不可用的管理操作。该应用可更新填充函数所使用的数据以使得在主机再次重新填充该存储器的情况下,该存储器将包括最新改变(如果有的话)。在框380之后,这些步骤结束。

[0078] 图4是示出在一个实施例中的主机内的存储器管理系统接收分配和使用存储器的应用请求的处理的流程图。在框410中开始,主机接收来自应用的存储器分配请求。主机可对来自多个应用的请求进行服务,并且可包括在运行于主机平台上的多个应用之间划分或共享有限的物理存储器的存储器管理器。存储器分配请求可包括诸如所请求的存储器大小或与所请求的存储器相关的其他参数等信息。

[0079] 在框420继续,主机接收指定该应用计划使用所请求的存储器分配的方式的存储器分配元数据。该元数据可包括诸如存储器分配对该应用的重要性或优先级、该分配是否适于分页到盘、在主机必须释放该分配的情况下该应用是否可恢复该存储器的内容等信息。

[0080] 在框430继续,主机从该应用接收可被主机调用来填充所请求的存储器分配的内容的存储器填充函数。具有用于填充该存储器分配的手段允许主机在主机需要释放该存储器分配以满足另一应用请求或释放其他存储器压力的情况下推迟该存储器的分配以及恢复该存储器。

[0081] 在决策框440继续,如果主机确定主机可响应于该请求直接分配存储器,则在框450继续,否则主机在框460继续。主机可基于该应用指示存储器对该应用有多重要以及该应用计划多么频繁地访问该存储器来确定是否直接分配该存储器。主机还可考虑其他因素,如在请求的时刻主机在其上运行的计算设备有多繁忙。

[0082] 在框450继续,主机从主机可用的物理存储器中分配所请求的存储器。主机可设立被该应用通过其来访问物理存储器的页表或其他虚拟存储器。传统主机直接响应于应用请求来分配存储器,或者在主机不能满足该请求的情况下使该请求失败。然而,本存储器管理系统可以推迟请求或执行其他存储器管理动作以更高效地使用存储器和其他主机资源。

[0083] 在框460继续,主机存储接收到的请求以及接收到的元数据和填充函数以供稍后进行分配。主机可维护各应用所请求的存储器的数据存储,在需要存储器管理动作(如使得更多存储器可用)时从该数据存储中进行选择。该应用所提供的元数据帮助主机确定哪些应用和应用内的分配将受到主机的存储器管理动作的最少负面影响。在框460之后,这些步骤结束。

[0084] 在一些情况下,应用的开发人员不愿或不能做出改变以支持新操作系统特征。在

这样的情况下,本存储器管理系统仍然能够通过分析应用二进制信息以发现与存储器分配有关的信息来提供一些增强的存储器管理,如参考图5和6进一步描述的。

[0085] 图5是示出在一个实施例中的存储器管理系统分析未被专门设计成提供存储器分配信息的应用的处理的流程图。

[0086] 在框510开始,该系统检测描述应用做出的存储器分配的信息对其不可用的应用。该应用可在它的二进制模块上包括指定该应用是否参与本存储器管理系统所提供的存储器模型的标志或其他信息,或可以提供另一指示。该系统可将先前确定的应用分配信息高速缓存以使得对每一应用的分析只被执行一次。因而,在这一步骤中,该系统还确定该分析先前尚未被执行。

[0087] 在框520继续,该系统对该应用执行分析以确定该应用所做出的存储器分配。该分析可包括对该应用的各种形式的静态和/或动态分析。静态分析在不运行该应用的情况下发生,并调查该应用的二进制代码以确定该应用的行为。动态分析在该应用正在运行的情况下发生,并调查该应用的存储器内代码、数据结构、以及其他信息以确定该应用的行为。该系统可寻找对用于分配、访问、以及释放存储器的特定主机API的调用,并且可记录位置、周围补充信息、用于填充所分配的存储器的各步骤,等等。

[0088] 在框530继续,该系统将该应用挂钩以在该应用内的该应用分配存储器的各位置处提供分配信息。各种形式的应用挂钩技术是可用的,如MICROSOFT™ Detours以及其他,它们允许重定向应用二进制代码以在该程序的任何点处截取或扩充应用的标准行为。例如,该系统可以提供调用用于提供分配元数据的分配函数而非调用该应用最初调用的不带有元数据的标准分配函数的应用挂钩。

[0089] 在框540继续,该系统从该应用接收分配存储器的请求和经挂钩的应用代码所提供的相关联的分配信息。该请求由主机接收,主机能够确定该分配信息来自未被专门设计成提供分配信息的应用。对这样的应用而言,主机可与对被设计成与该系统一起工作的应用不同地操作。例如,主机可虑及未被设计成与该系统一起工作的应用所提供的可能更少的信息。在一些实施例中,为了鼓励采用新存储器模型,该系统可以通过在存储器压力发生时将不依附新存储器模型的应用的数据优先对换到盘或执行其他存储器管理动作来惩罚这些应用。在框540之后,这些步骤结束。

[0090] 图6是示出在一个实施例中的存储器管理系统静态地分析应用并提供增强的存储器信息的清单的处理的流程图。

[0091] 在框610开始,该系统接收不与对存储器分配函数的调用相关联地提供存储器分配信息的已编译应用代码。该系统可在运行应用的第一次请求时、在来自计算设备的硬盘驱动器或其他存储的扫描的批量处理期间、或响应于特定用户或管理员请求,来接收应用代码。已编译应用代码可包括针对特定处理器(例如,x86或x64二进制代码)的二进制代码、针对特定运行时的中间语言代码、或其他形式的非源应用代码。在源可用时,该应用可被直接修改以使用该系统,如本文其他部分所述的。

[0092] 在框620继续,该系统对接收到的应用代码执行静态分析以确定该代码内该应用分配存储器的位置。静态分析可以寻找对存储器分配函数的调用,如通过导入的模块或其他手段。在一些情况下,静态分析擅长找出函数调用以及所传递的参数。在一些实施例中,该系统用动态分析来扩充静态分析,并运行该应用以标识不能静态地获得或不能容易地静

态地检测的信息。

[0093] 在框630继续,该系统标识所分析的应用代码中的一个或多个存储器相关的代码动作。代码动作可包括存储器分配、存储器访问、存储器释放、和其他存储器操作。该系统可通过动作发生的地址或通过可稍后提供给主机或者被修改以改变默认应用行为的其他标识,来标识代码动作。

[0094] 在框640继续,该系统标识与每一所标识的代码动作有关的周围信息,所述周围信息提供了描述所分配的存储器如何被该应用使用的附加信息。周围信息可以告知该系统该分配被存储了多久(例如,在同一函数中被使用了一次或存储在全局变量中以供重复的后续使用)、该应用有多容易填充该存储器分配的内容,等等。周围信息可通过静态和/或动态分析来被标识。

[0095] 在框650继续,该系统将所标识的存储器相关的代码动作和任何所标识的周围信息储存在数据存储中以供在运行该应用时主机的后续检索。在一些实施例中,该系统将所标识的信息储存在与应用模块一起存储的清单中,使得加载该应用的主机可以执行对所标识的位置的任何截取或挂钩以向主机提供描述该应用所做出的存储器分配的更多信息。主机可以使用这一信息例如来在该系统空闲时或未被充分利用时预先分配该应用将在未来某一时刻需要的存储器,如该分析所确定的。这允许传统应用参与本存储器管理系统所提供的存储器模型。在框650之后,这些步骤结束。

[0096] 图7是示出一个实施例中的存储器管理系统响应于检测到的存储器压力来采取与存储器有关的动作的处理的流程图。存储器压力可在具有有限存储器的设备接近其可用存储器时发生。例如,系统操作系统可以将存储器压力定义为在已使用物理RAM的90%时发生。

[0097] 在框710开始,该系统接收描述多个存储器分配请求以及一个或多个应用如何使用存储器分配的信息。在一些情况下,该系统在从应用接收到各分配以及描述每一分配是如何使用的相关联的元数据时存储分配的列表或其他数据结构。分配使用信息可包括可帮助该系统确定哪些分配可被释放、分页或以其他方式处理来高效地管理可用存储器的优先级或其他信息。

[0098] 在框720继续,该系统检测指示需要采取动作来继续高效地运行各应用的存储器压力。检测到的存储器压力可包括内核或其他主机可采取的各种条件或动作。例如,内核可能想要关闭一排存储器以节省能量,并且可检测储存在该排上的分配并释放这些分配或将这些分配对换到盘以使得该排可被关闭。作为另一示例,该系统可以对各存储器排进行碎片整理以获得可被关闭的空排。在主机没有完全控制存储器的使用的情况下,在没有应用协作的情况下执行移动或释放存储器的操作通常是困难的。然而,本文描述的存储器管理系统给予主机更多信息和对如何管理存储器的更多控制。

[0099] 在框730继续,该系统枚举接收到的分配请求以确定要对其采取动作的分配。该系统可以遍历分配的列表或其他数据结构以确定在它们需要被释放的情况下能被容易地恢复的分配或应用不太可能再次使用的分配。

[0100] 在框740继续,该系统选择所枚举的分配中的要对其采取动作的一个或多个分配。该系统可基于接收到的描述请求分配的应用如何使用该分配的信息来选择分配。在一些情况下,取决于存储器管理器的特定目标(例如,对特定总大小的存储器进行操作并找出加起

来达到或超过该大小的分配), 该系统可以选择多个分配。

[0101] 在框750继续, 该系统对所选分配执行动作以释放存储器压力。该动作可包括释放先前分配的存储器、将存储器内容对换到盘或其他存储、将存储器从先前位置移动到新位置, 等等。在一些情况下, 该系统向该应用通知该动作, 使得该应用可以修改依赖于该分配的行为。例如, 该应用可通过在该应用下次使用存储器时重新分配该存储器来进行响应。在框750之后, 这些步骤结束。

[0102] 图8是示出一个实施例中的存储器管理系统激活其存储器先前已被主机修改的应用的处理的流程图。在许许多多任务系统中, 各应用被推到后台并且稍后被用户或操作系统重新激活。在移动设备上, 用户可一次与一个应用进行交互, 并且操作系统可在特定应用处于前台时挂起其他应用。操作系统可释放其他应用的存储器或者将它流传送到盘或其他存储。在激活时, 操作系统可采取动作来准备该应用再次运行。

[0103] 在框810中开始, 系统接收激活应用请求。该请求可来自应用自身、来自用户、或来自操作系统。激活请求可简单地指示用户再次与该应用交互, 使得该应用可请求先前所分配的存储器。

[0104] 在框820中继续, 该系统标识一个或多个先前接收到的应用存储器分配。该系统可维护一个或多个分配所请求的每一分配的列表或其他数据结构, 使得该系统可以通过遍历该列表来标识先前接收到的存储器分配。该系统可以检查每一分配的状态, 如与该分配相关的存储器是否仍然可用且仍然包含该应用最后置于其中的内容。

[0105] 在决策框830继续, 如果该系统确定该应用的分配全部就绪, 则该系统跳转到框860, 否则该系统在框840继续。该系统可以释放、移动、分页、或以其他方式对各应用分配采取动作以提供用于运行其他任务的更多存储器, 如运行其他应用。因而, 在再次运行该应用时, 该系统可恢复先前操作依据的分配或可向该应用通知这些动作, 使得该应用可以采取适当的动作。

[0106] 在框840继续, 该系统分配未就绪的分配以将这些分配置于该应用所预期的状态。该系统可以通过分配满足该分配的可用物理存储器来向这些分配指派物理存储器。在一些情况下, 该系统可以提供虚拟存储器, 如由物理存储器和对换文件所支持的那些。

[0107] 在框850继续, 该系统使用该应用所提供的填充函数来填充所分配的存储器内容。该应用向该系统提供足够信息, 使得该系统可以释放并重新创建各应用的存储器分配。这允许该系统做出在各应用之间共享资源的高效决策, 同时保持对应用的负面影响很低。理想地, 在该系统处于存储器或其他压力之下时, 该系统释放或移动将无论如何也不会很快使用的并且随后在它被再次需要之前有时间进行恢复的存储器。在传统系统中, 主机所能做的最好的是猜测, 但使用本文的存储器管理系统, 主机可以非常高效地选择要操作的一个或多个存储器分配。

[0108] 在框860继续, 该系统激活所请求的应用并向该应用提供该应用所预期的存储器分配。该系统可在为该应用准备存储器时挂起该应用, 并且在该应用的所有存储器分配就绪时恢复该应用。在框860之后, 这些步骤结束。

[0109] 从前面的描述中可以看出, 可以理解, 存储器管理系统的特定实施例只是为了说明, 但是, 在不偏离本发明的精神和范围的情况下, 可以进行各种修改。因此, 本发明只受所附权利要求限制。

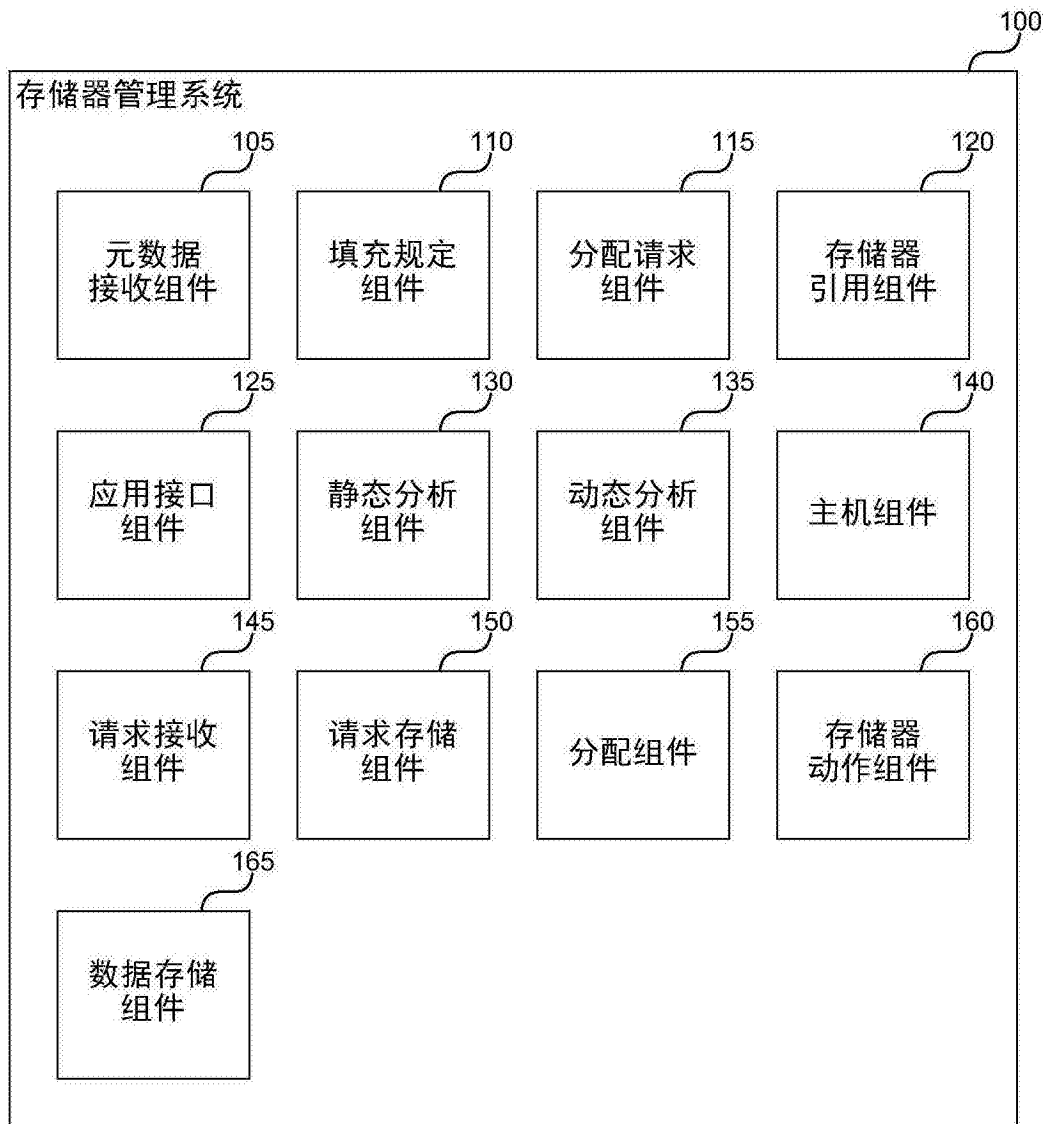


图1

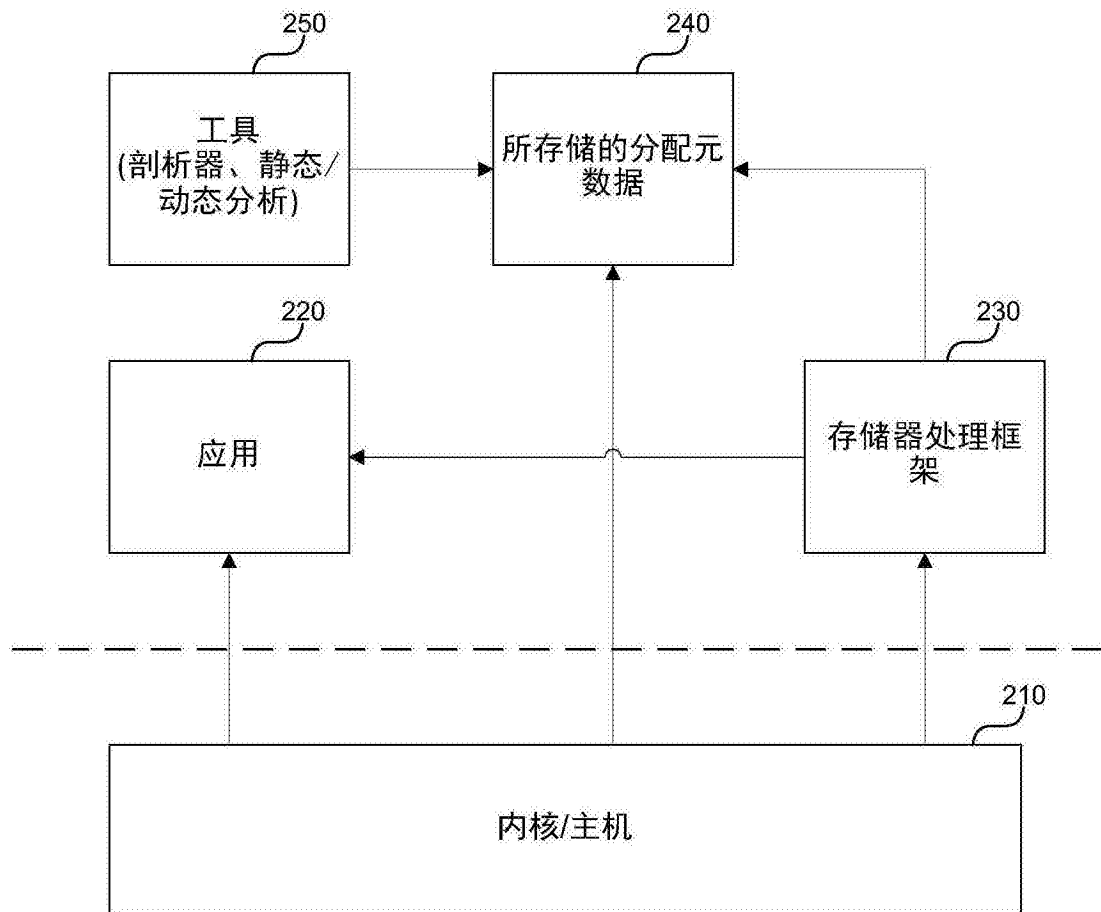


图2

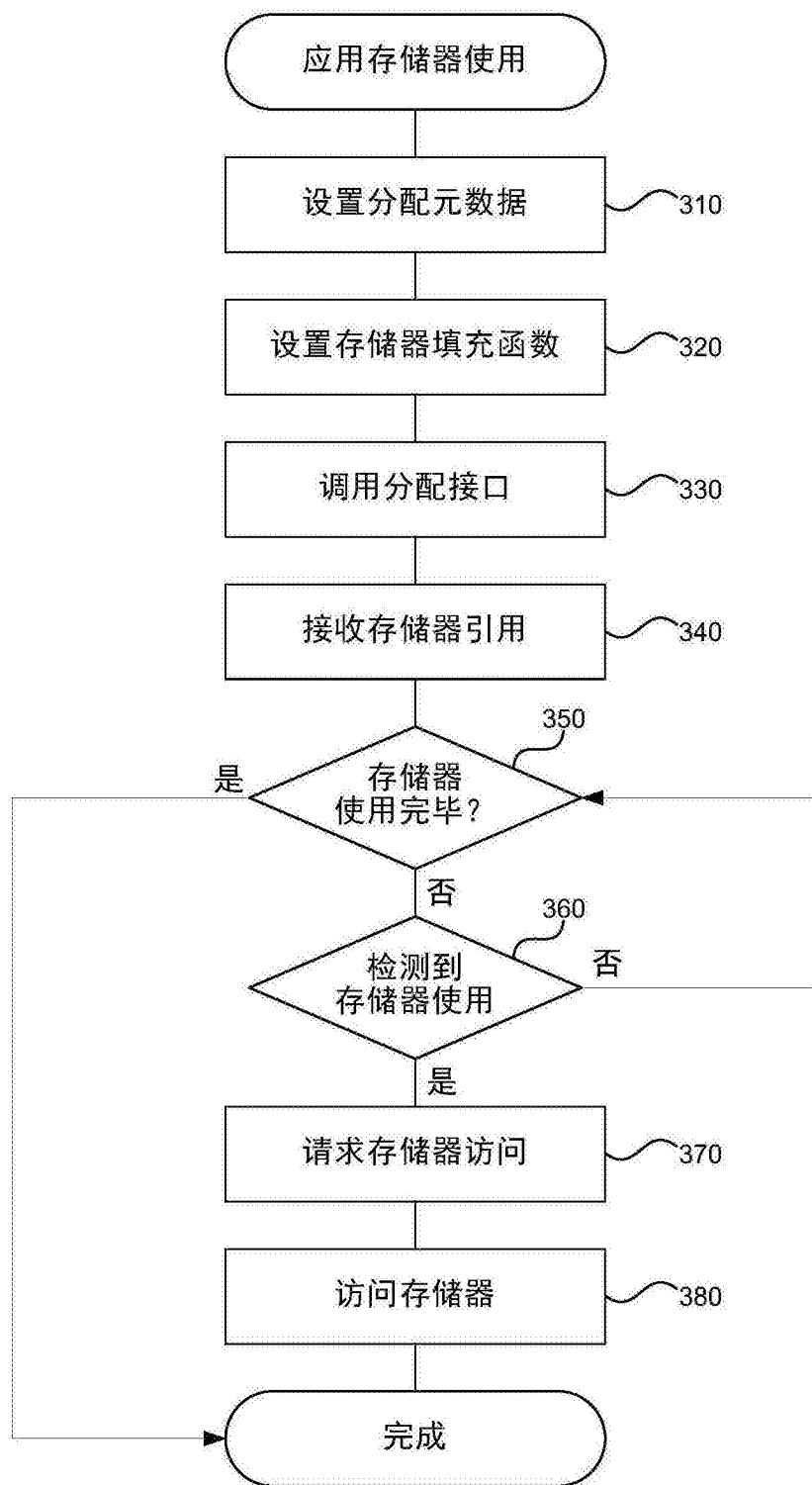


图3

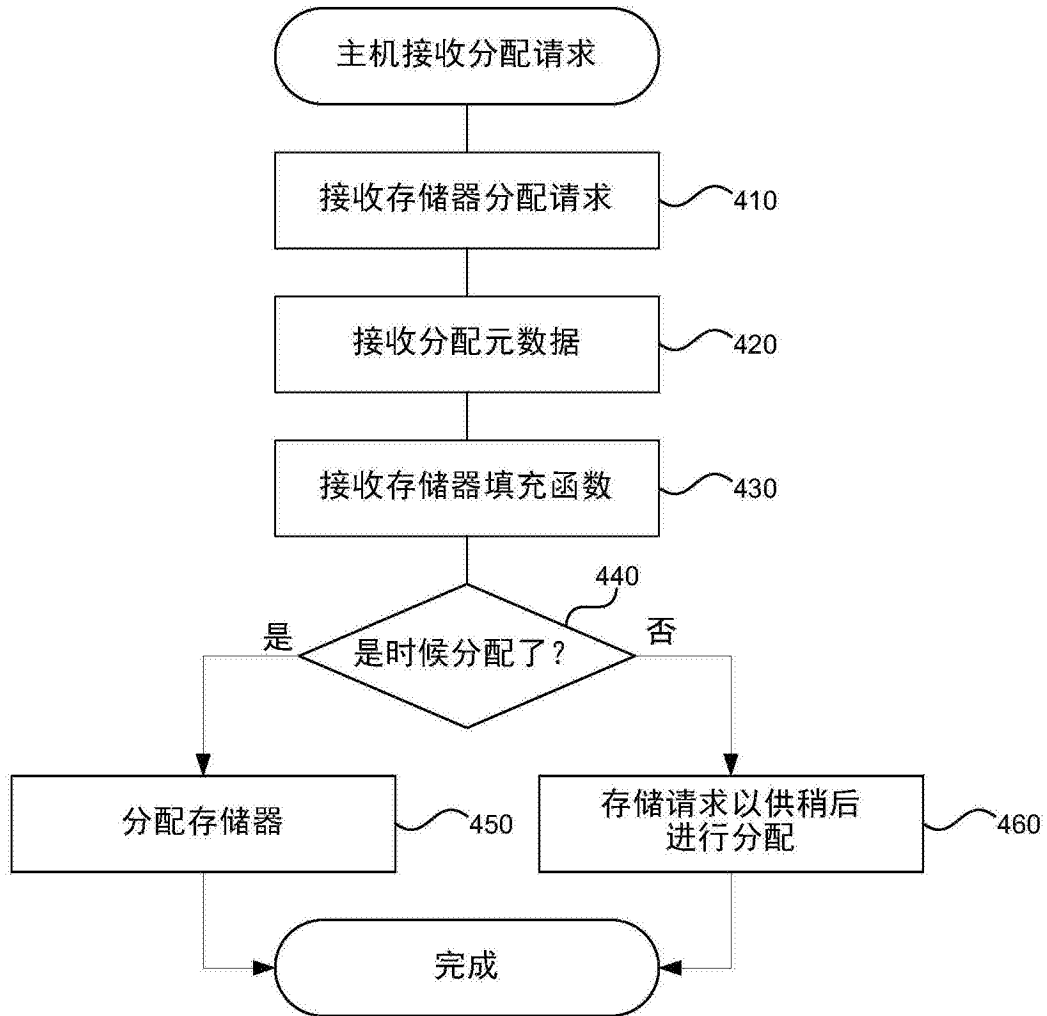


图4

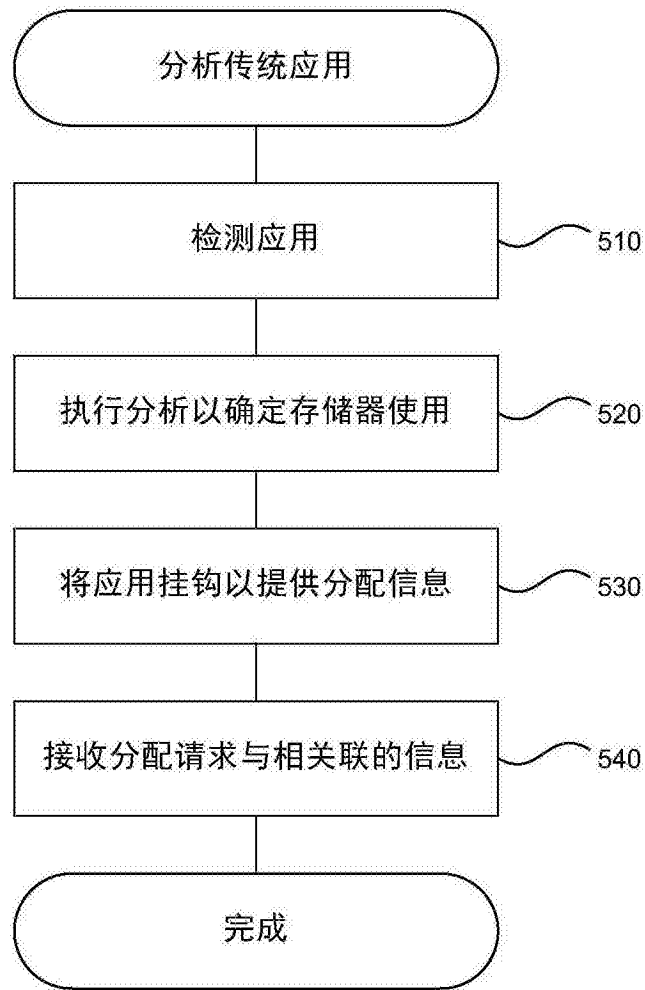


图5

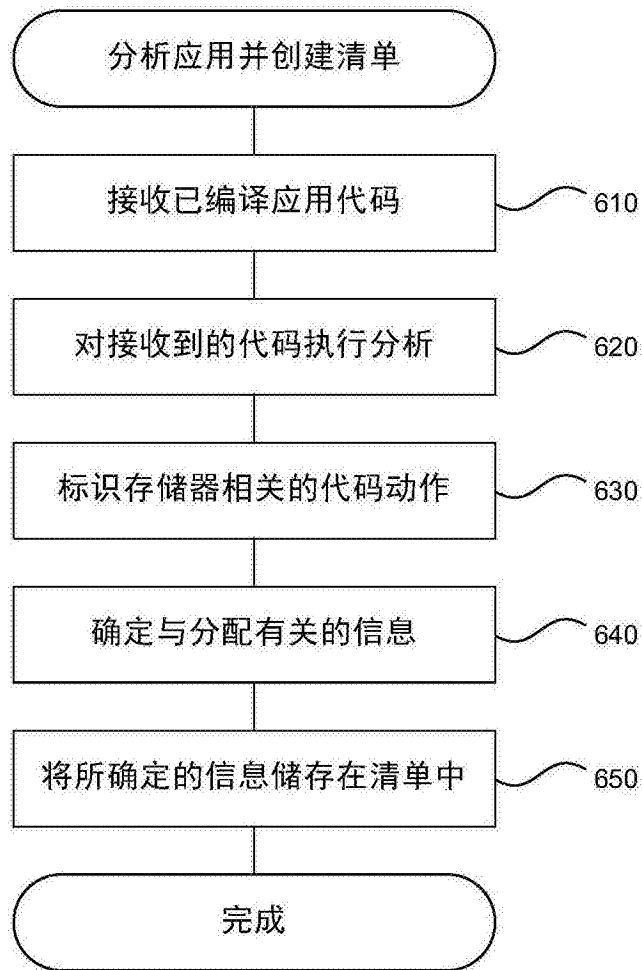


图6

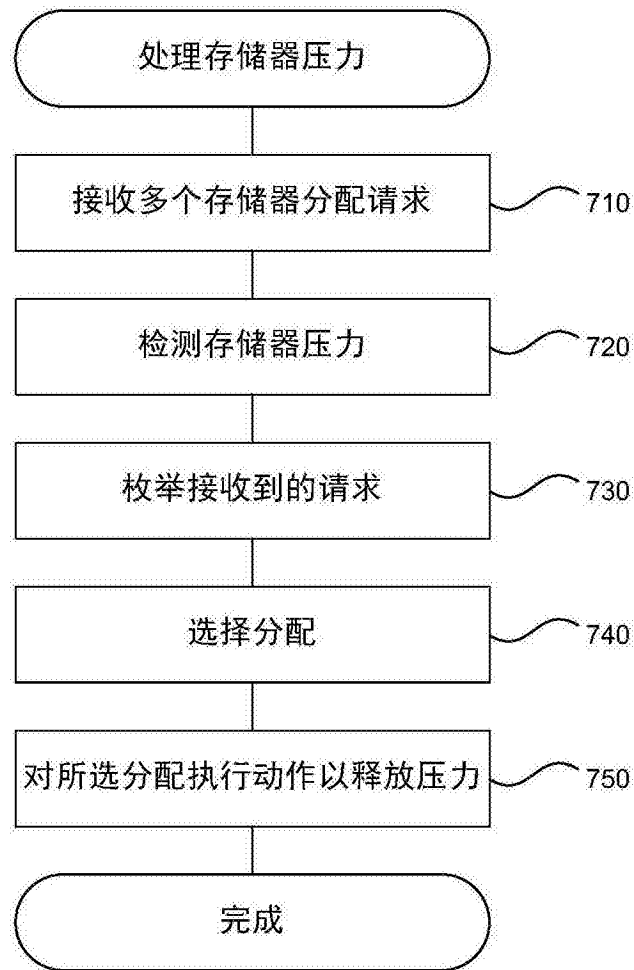


图7

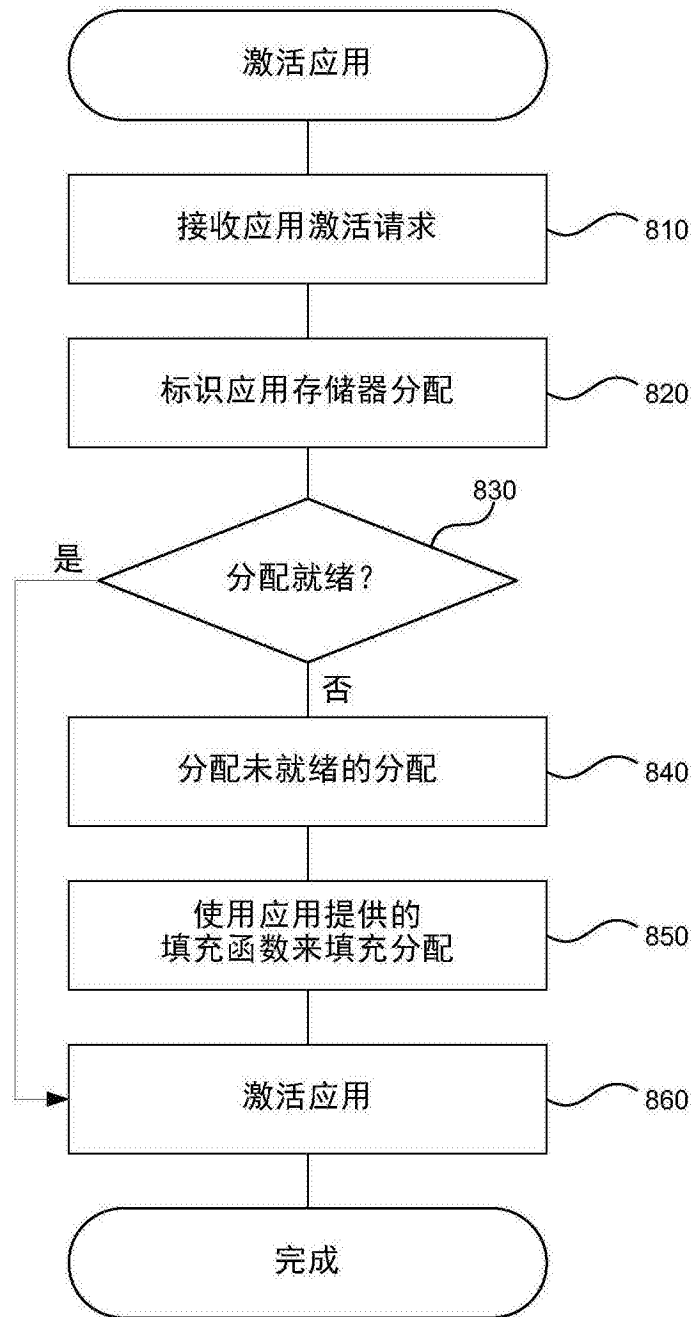


图8