



(12)发明专利

(10)授权公告号 CN 104603750 B

(45)授权公告日 2018.05.01

(21)申请号 201380045529.9

(22)申请日 2013.08.19

(65)同一申请的已公布的文献号

申请公布号 CN 104603750 A

(43)申请公布日 2015.05.06

(30)优先权数据

13/598,616 2012.08.30 US

(85)PCT国际申请进入国家阶段日

2015.02.28

(86)PCT国际申请的申请数据

PCT/US2013/055636 2013.08.19

(87)PCT国际申请的公布数据

W02014/035711 EN 2014.03.06

(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 E·南丁格尔 K·斯里尼瓦桑

(74)专利代理机构 上海专利商标事务所有限公司 31100

代理人 罗婷婷

(51)Int.Cl.

G06F 9/4401(2018.01)

G06F 9/445(2018.01)

(56)对比文件

CN 101283330 A, 2008.10.08,

CN 101673245 A, 2010.03.17,

EP 0810522 A2, 1997.12.03,

US 6446254 B1, 2002.09.03,

审查员 黄成

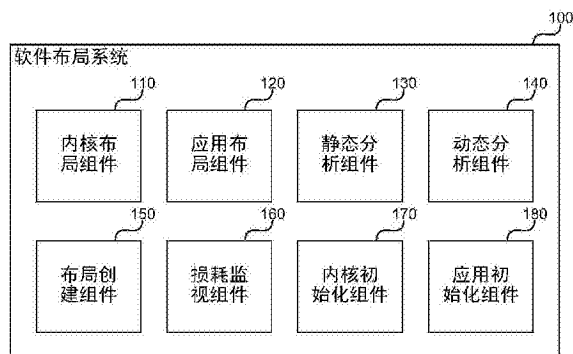
权利要求书2页 说明书14页 附图6页

(54)发明名称

使用BPRAM来布局和执行软件应用

(57)摘要

本文中描述了通过将恒定数据和可执行代码移动到字节可寻址的持久随机存取存储器(BPRAM)中来加速计算机系统引导时间和/或应用初始化时间的软件布局系统。该系统确定操作系统或应用的哪些组件和方面不频繁地改变。该系统从此信息构建高性能BPRAM高速缓存以提供对这些被频繁使用的组件(包括内核)的快速访问。结果是内核或应用代码和数据结构关于存储器取回具有高性能访问和执行时间。因此,软件布局系统为正常操作提供准备操作系统和应用的更快方式并降低了在初始化上所花的时间。



1. 一种检查应用以标识要预布局在字节可寻址的持久随机存取存储器BPRAM中的代码和数据的计算机实现的方法,该方法包括:

标识目标应用,对所述目标应用将执行快速布局以创建所述应用的快速初始化图像,其中所述目标应用具有包括在布局方面与所述快速初始化图像不同的一个或多个模块的存储格式;

对所标识的目标应用执行静态分析以确定与所述应用相关联的一个或多个静态和动态链接模块并标识所述应用所使用的不变到足以存储在BPRAM中的数据;

通过执行所述应用并监视所述应用的初始化以标识与所述应用相关联的附加模块和数据来执行动态分析;

标识适于存储从其初始化所述应用的预布局图像的BPRAM设备;

基于指示所述BPRAM设备的每个区域已经被写入多少次的损耗状态信息来选择所述BPRAM设备内的一个区域以供存储所述快速初始化图像;

基于在静态和动态分析期间所标识的所述目标应用的存储器内布局来创建快速初始化图像,所述存储器内布局与所述应用的所述存储格式不同,其中所述快速初始化图像准备好执行并且是在接收执行所述应用的请求之前被创建的;

将所述快速初始化图像复制到所标识的BPRAM设备的所选的区域中,

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,标识所述目标应用包括提供管理工具,所述管理工具可被信息技术(IT)人员运行来标识所述目标应用并启动创建快速初始化图像以供存储在所述BPRAM设备中的过程。

3. 如权利要求1所述的方法,其特征在于,标识所述目标应用包括提供具有与所述目标应用相关联的安装包的清单。

4. 如权利要求1所述的方法,其特征在于,执行静态分析包括检查扩展所述目标应用的功能的一个或多个插件模块。

5. 如权利要求1所述的方法,其特征在于,执行静态和动态分析包括检查与所述目标应用相关联的运行时,并将与所述运行时相关联的代码和数据放置在所述BPRAM设备中。

6. 如权利要求1所述的方法,其特征在于,执行动态分析包括收集关于所述目标应用如何初始化的信息,所述信息可用于产生所述应用可从其直接执行从而跳过了所述目标应用的典型初始化步骤中的至少一些步骤的所述快速初始化图像。

7. 如权利要求1所述的方法,其特征在于,执行静态和动态分析包括捕捉标识在静态和动态分析期间所标识的每个所述模块的签名信息,其中所述签名信息指示所述快速初始化图像相对于所述模块而言是否是最新的。

8. 如权利要求1所述的方法,其特征在于,执行静态和动态分析包括标识影响外部资源的初始化,以使得所述初始化可在从所述快速初始化图像初始化所述目标应用时被执行。

9. 如权利要求1所述的方法,其特征在于,标识所述BPRAM设备包括标识具有足以存储所述快速初始化图像的空闲空间的持久的且字节可寻址的存储器设备。

10. 如权利要求1所述的方法,其特征在于,创建所述快速初始化图像包括用一起存储在所述BPRAM设备中的一个或多个模块代码部分的内容来创建统一布局图像。

11. 如权利要求1所述的方法,其特征在于,复制所述快速初始化图像包括标识所述目

标应用内表示在所述目标应用的当从所述快速初始化图像启动时不必要的典型初始化步骤之后的点的执行位置,并基于所标识的执行位置来修改所述目标应用的入口点。

12.一种使用字节可寻址的持久随机存取存储器BPRAM来布局和执行软件应用的计算机系统,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;

应用布局组件,所述应用布局组件确定被应用用来初始化所述应用的应用数据和软件指令,并确定所述应用数据和软件指令的、将允许所述应用从BPRAM初始化的布局;

静态分析组件,所述静态分析组件静态地分析软件二进制码、数据模块或与所述应用相关联的其他存储的软件数据以确定所述软件如何访问指令和数据并标识适于存储在BPRAM中的指令和数据;

动态分析组件,所述动态分析组件动态地分析正在运行的所述应用以收集在静态分析期间未识别出的所述应用的行为有关的附加信息;

布局创建组件,所述布局创建组件创建用于基于由所述应用布局组件确定的所述布局来初始化所述应用的图像,基于指示所述BPRAM的每个区域已经被写入多少次的损耗状态信息来选择所述BPRAM设备内的一个区域,并将所述图像存储在BPRAM的所选区域中;以及

应用初始化组件,所述应用初始化组件在请求之际使用由所述布局创建组件创建的BPRAM中的图像来初始化所述应用。

13.如权利要求12所述的系统,其特征在于,所述应用布局组件允许使所述应用休眠并在恢复事件之际快速地恢复所述应用。

14.如权利要求12所述的系统,其特征在于,所述应用布局组件标识所述应用的不频繁改变的数据以供存储在BPRAM中。

15.一种具有指令的计算机可读存储介质,当所述指令被执行时使得机器执行如权利要求1-11中任一权利要求所述的方法。

16.一种计算机系统,包括用于执行如权利要求1-11中任一权利要求所述的方法的装置。

使用BPRAM来布局和执行软件应用

[0001] 背景

[0002] 各计算机系统在它们被打开时都会经历引导阶段,该引导阶段为使用准备计算机并接管硬件对操作系统的控制。一般来说,硬件包括一些初始化该硬件的固件,诸如基本输入/输出系统 (BIOS) 和扩展固件接口 (EFI)。固件随后一般关注盘驱动器以标识要运行的主引导记录 (MBR) 或其他第一指令集。这些指令可调用引导加载器,该引导加载器随后加载并初始化操作系统 (或潜在地提供操作系统的从其引导计算机系统的菜单)。类似地,一旦操作系统被加载,应用就也经历初始化阶段,在初始化阶段期间,与应用相关联的软件和数据模块被加载、存储器状态被 (例如用资源或其他数据) 初始化等等。

[0003] 这些引导和初始化过程通常是耗时的。不仅复杂的硬件可花费时间来初始化并到达操作状态,而且软件也要花费时间来从盘加载并花费时间来运行以执行各种初始化任务。计算机系统的用户无法使用系统或应用,直到该过程完成,因此时间被浪费了。此外,在诸如将计算机从睡眠状态、低功率状态或休眠唤醒之类的其他时段期间,系统可执行相似的类引导过程,该类引导过程延迟了用户对计算机系统的使用。对于移动设备、游戏控制台、路由器和其他设备 (它们被内部地实现为计算机系统,虽然消费者可能将它们看成为集成的消费者电子设备) 而言,用于引导的时间可显著地降低该设备的可用性。

[0004] 在标准操作系统中,诸如内核之类几乎不被更新的恒定代码以与用户应用代码相同的方式从传统驱动器中被读出。这导致小于优化引导时间,因为操作系统经由复杂的加载序列将其自身从硬盘驱动器或其他相对慢速的资源带入操作状态。即使使用现代的基于闪存的盘驱动器,用于加载软件代码并设置初始配置的读写数量也可消耗用户可察觉的时间量。这对于可涉及访问和加载许多库并随后用数据来初始化这些库所指定的存储器区域的应用而言同样正确。该数据中的大部分都是每次操作系统或应用加载时都相同的恒定数据。过去的技术已关注在现有的、字节不可寻址的硬盘驱动器上重新排列该数据的方式 (该方式仍涉及可察觉的延迟),或关注在应用被用户请求之前尽早执行普通初始化序列 (参见例如微软™ WINDOWS™ Prefetch和SuperFetch特征)。

[0005] 概述

[0006] 本文中描述了通过以下方式加速计算机系统引导时间和/或应用初始化时间的软件布局系统:将恒定数据和可执行代码移入非易失性的高性能存储器 (例如,ReRAM或PCM) 中,而将改变数据保留在易失性存储 (诸如,DRAM) 上。该系统确定操作系统或应用的哪些组件和方面是恒定的并且不遭受改变或遭受最少的改变。该系统从此信息构建高性能存储器ReRAM/PCM高速缓存以提供对这些被频繁使用的组件 (包括内核) 的更快速访问。结果是内核或应用代码和数据结构关于存储器取回具有高性能访问和执行时间。对于应用而言,该系统可执行类似的过程。软件应用还经历常见的启动序列来初始化可使用本文中描述的技术来预测和加速的过程数据结构、线程、资源等。在一些情况下,诸如在高性能高速缓存存储为字节直接可寻址的情况下,该系统甚至可直接从该高速缓存引导或执行。因此,该软件布局系统提供了更快的方式来为正常操作准备操作系统和应用,并降低了在初始化上所花的时间。

[0007] 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概念选择。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。

[0008] 附图简述

[0009] 图1是示出一个实施例中的软件布局系统的各组件的框图。

[0010] 图2是示出一个实施例中的软件布局系统用于检查操作系统内核以标识要预布局在BPRAM中的代码和数据的处理的流程图。

[0011] 图3是示出一个实施例中的软件布局系统用于从BPRAM初始化操作系统内核的处理的流程图。

[0012] 图4是示出一个实施例中的软件布局系统用于检查应用以标识要预布局在BPRAM中的代码和数据的处理的流程图。

[0013] 图5是示出一个实施例中的软件布局系统用于从BPRAM初始化应用的处理的流程图。

[0014] 图6是示出一个实施例中的各种类型的存储与软件布局系统所使用的数据的关系的框图。

[0015] 详细描述

[0016] 本文中描述了通过以下方式来加速计算机系统引导时间和/或应用初始化时间的软件布局系统:将恒定数据和可执行代码移入非易失性的高性能存储器(例如,ReRAM或PCM)中,而将改变数据保留在易失性存储(诸如,DRAM)上。该系统确定操作系统或应用的哪些组件和方面是恒定的并且不遭受改变或遭受最少的改变。该系统从此信息构建高性能存储器ReRAM/PCM高速缓存以提供对这些被频繁使用的组件(包括内核)的更快速访问。例如,该系统可将内核数据结构和代码保持在非易失性和高性能ReRAM中,而将动态/改变数据保留在易失性DRAM存储器中。结果是内核或应用代码和数据结构关于存储器取回具有高性能访问和执行时间。

[0017] 该系统可对应用执行类似的过程。软件应用还经历常见的启动序列来初始化过程数据结构、线程、资源等,这些启动序列可使用本文中描述的技术来预测和加速。例如,该系统可标识应用的启动序列中的引用几乎不改变的数据的各部分,并可将该数据高速缓存非易失性、高性能存储器高速缓存中,其中每次该应用启动时该数据都可被访问。在一些实施例中,该系统还以供快速转移到存储器中使得内核或过程变得活跃的方式来对数据进行布局。例如,该系统可确定操作系统或应用在操作期间在存储器中的最终布局,将那个布局的图像存储在高性能高速缓存中,并随后在下一次操作系统或应用被选择启动时,该系统可使用直接存储器存取(DMA)或其他高性能复制技术来将预布局引导/初始化图像转移到存储器中以供执行。在一些情况下,诸如在高性能高速缓存为字节直接可寻址的情况下,该系统甚至可直接从该高速缓存引导或执行。因此,软件布局系统提供为正常操作准备操作系统和应用的更快方式并降低了在初始化上所花的时间。

[0018] 如本文中所描述的,软件布局系统在两个概念阶段中操作:布局 and 启动。在布局阶段期间,该系统分析一个或多个操作系统或应用以确定操作系统或应用如何初始化其自身。该系统可使用本领域公知的各种静态和动态分析技术来分析二进制的可执行文件、存储器内数据、实况执行或其他信息以标识操作系统或应用在启动期间所使用的指令和数

据。例如，MICROSOFT™ WINDOWS™操作系统所使用的可移植可执行文件格式(PE)包含可被标记为只读或读/写的数据部分，并且该软件布局系统可使用此信息来将这些只读部分放置在ReRAM或另一高速缓存中。该系统还可跟踪历史改变信息来构建关于哪些指令或数据几乎不改变的图片。此几乎不改变的数据随后被定向用于从更快速的存储(诸如ReRAM)中加载。该系统创建描述分析的结果且提供软件指令和/或数据的布局图像或其他数据结构以供使用更快速的存储器来进行存储。

[0019] 在启动阶段期间，该系统利用先前执行的布局工作来使得操作系统或应用的启动更快。这可包括定向BIOS或其他启动指令以从ReRAM高速缓存直接访问初始化图像，或者可包括用于使用DMA或其他快速转移方法来将准备执行的图像从高速缓存复制到易失性存储器中以进行传统执行的指令。由于与加载许多图像碎片相比，这样的转移更快，因此存在初始化过程的显著加速以及关于应用何时准备好被用户使用的时间方面的降低。

[0020] 在任何现代操作系统中，数据要么是可变得要么是不可变的。诸如应用的堆内包含的数据之类的数据通常是可变的，因为它在应用执行期间改变。可执行图像(软件代码)和全局常量是不可变数据的示例。目前，可变和不可变数据两者都被保持在耐久存储(诸如闪存或硬盘驱动器)中，并且在执行期间被高速缓存在DRAM中使得它对应用而言是可访问的。硬盘驱动器和闪存通常是字节不可寻址的，因此执行通常无法从这样的设备直接进行。这是为什么这样的数据首先被复制到DRAM(其是字节可寻址的)原因，在DRAM中可通过执行软件代码来执行并访问这样的数据。

[0021] 一般来说，目前的存储器分层结构在慢速、持久、批量存取技术和快速、字节可寻址、非持久技术之间具有区分。字节可寻址的持久存储器技术(诸如，相变存储器、ReRAM或忆阻器)通过提供即字节可寻址又持久的存储来提供桥接DRAM和持久存储之间的间隙的存储器。在本文中将这些技术称为BPRAM(字节可寻址的持久随机存取存储器)。这些不同的存储器技术全部都具有变化的性能以及耐久性(耗损)保证，但它们引入了可在现代计算机中使用的一种新型存储器。使用BPRAM来存储不可变对象可改善性能、释放DRAM并节省功率，并且该软件布局系统标识应用或操作系统内可被迁移到BPRAM中以得到这些优点的数据结构。

[0022] 由对BPRAM的这种使用所引入的一个新的考虑是耗损。由于在耗损之前向BPRAM设备写入的次数通常远小于DRAM或硬盘，因此通常需要关注以确保该数据不会由于该技术耗损而被丢失。因此，对BPRAM的最简单使用是首先使得该BPRAM在与DRAM并排的存储器总线上可用。接着，可考虑仅将应用可执行代码放在可用的持久存储器上。不是从较慢速的批量存储中提取应用二进制代码并将这些二进制代码高速缓存在易失性存储器中，这些可执行代码可被简单地存储在BPRAM中并直接从BPRAM执行。由于可执行代码与应用数据相比被不频繁得多地更新，因此可以肯定BPRAM将具有长生存期。一个问题可能是执行将变慢，因为BPRAM通常比DRAM更慢，然而CPU预取和高速缓存可隐藏大多数差异，并且BPRAM可比硬盘驱动器快得多。BPRAM中高速缓存的可执行代码可包括任何类型的二进制码，包括应用、二进制代码、动态链接库(DLL)或构成操作系统的组件。

[0023] 除了简单地高速缓存可执行代码外，计算环境的其他部分也可被高速缓存在BPRAM中。例如，当代码被编译时，被定义为“常量”或“只读”(如用可执行格式来标记或通过另一种方法来确定)的数据可被放置在BPRAM中而非被保持在DRAM中。使用大量静态数据来

进行计算的应用可释放DRAM并改为将那些数据结构放置在BPRAM中。

[0024] 最后,运行时可被设计成将数据结构迁移到BPRAM中以及将数据结构从BPRAM中迁移出,这取决于它们的存取模式。例如,散列表可真正地被表示成状态机,其中该散列表被频繁地更新、随后被读取者排他地访问、并随后被再次更新。这些类型的存取模式可意味着数据结构可在它被读取者访问的同时被放在BPRAM中。然而,当应用尝试通过对这些数据结构进行写入来更新这些数据结构时,这些数据结构将被移回易失性 (DRAM) 存储器。

[0025] 应用以几种不同的方式从这种可用性中获益。首先,它消除了存在过程时应用将各数据结构串行化到盘中的需要。当然,这仅在教育应用可推断数据结构的一致性的情况下可用。当应用崩溃时,可能数据结构不再可用,而必须从保持在另一位置的副本 (或快照) 中重新创建。第二,如果应用对其数据结构的持久版本具有即时访问权,则应用性能可被改善。最后,对于与DRAM相比具有更低能量覆盖区的DPRAM而言,DRAM的有意义部分可被替换为BPRAM以通过将其数据的一部分保持在BPRAM中来降低计算机器的总能量覆盖区。

[0026] 在标准操作系统中,以与用户应用代码相同的方式将诸如内核之类几乎不被更新的恒定代码从传统驱动器中读出。软件布局系统确定操作系统的哪些组件是恒定的并且没有遭受改变或遭受了最少的改变,并且构建高性能ReRAM/PCM高速缓存来提供对这些被频繁使用的组件 (包括内核) 的快速访问。因此,该系统将内核数据结构和代码保持在非易失性和高性能ReRAM中,同时将改变数据保留在易失性DRAM或其他系统。

[0027] 传统上,在操作系统已被初始化和稳定化后,用户空间应用才被新鲜加载。然而,稍后加载应用或者将应用加载在另一操作系统中要求该应用加载来自库的碎片,并且随后被加载到存储器中的用户数据形成完整的实况图像。软件布局系统将应用及其存储器快速地持久保存到ReRAM/PCM中以允许快速转移到同一或另一系统上并稍后在同一或另一系统上快速重构。这种挂起和恢复移除了使用操作系统加载序列来形成应用图像的成本和负担,并将用户数据持久保存在实况状态中。该系统的这些和其它元素将在下文中更详细地描述。

[0028] 图1是示出一个实施例中的软件布局系统的各组件的框图。系统100包括内核布局组件110、应用布局组件120、静态分析组件130、动态分析组件140、布局创建组件150、耗损监视组件160、内核初始化组件170和应用初始化组件180。这些组件中的每一个都在此处进一步详细讨论。

[0029] 内核布局组件110确定被操作系统内核用来进行引导的数据和软件指令,并为字节可寻址的持久随机存取存储器 (BPRAM) 确定内核数据和软件指令的、将允许内核从BPRAM引导的布局。在字节可寻址的存储器快到足以允许令人满意的性能的情况下该内核可直接从BPRAM引导,或者该内核可通过首先将BPRAM中存储的布局复制到另一存储器 (例如,DRAM) 中来间接形成BPRAM。一些操作系统可具有会影响从何处运行代码的决策的附加要求,诸如安全引导过程。内核布局组件110调用静态分析组件130、动态分析组件140和布局创建组件150来确定内核的布局图像并将所确定的图像复制到BPRAM中。内核布局组件110还可更改引导数据结构或其他信息以使得计算机系统从被复制到BPRAM的图像中引导。内核布局组件110还可确定针对内核的其他类型的初始化 (诸如,重启、休眠、从睡眠恢复等) 的布局。

[0030] 应用布局组件120确定被应用用来初始化该应用的数据和软件指令,并为BPRAM确

定应用数据和软件指令的、将允许该应用从BPRAM初始化的布局。类似于内核,应用可直接从BPRAM运行,或者操作系统可首先将BPRAM中存储的应用布局复制到另一存储器设备中并从那里执行该应用。该布局还可用于使应用状态休眠并快速地恢复该应用状态,如同有时在当今的移动设备上使用字节不可寻址的闪存来实现的那样。应用可运行有改变数据和相对不改变的数据的组合,并且应用布局组件120可决定将相对不改变的数据放置在BPRAM中,而将改变数据放置在易失性存储器(诸如DRAM)中。每次应用运行时诸如应用二进制码、常量和其他类型的数据之类的数据项可以保持相同,或者仅被很少地改变,诸如应用二进制码可每年被更新若干次或更少。与当今通常遵循的加载过程相比,将该信息以可直接执行的布局保持在BPRAM中允许该应用快速得多地进入运行状态,而在当今通常遵循的加载过程期间,许多具有指令和数据的二进制代码可被分开地从一个存储位置(例如,硬盘驱动器)加载,并在该应用准备好运行之前被映射到存储器中。

[0031] 应用布局组件120调用静态分析组件130、动态分析组件140和布局创建组件150来确定应用的布局图像并将所确定的图像复制到BPRAM中。应用布局组件120还可更改操作系统数据结构或其他信息以使得操作系统将运行应用的请求重定向至被复制到BPRAM的图像。由内核布局组件110和应用布局组件120创建的图像对其他计算机系统和/或操作系统而言可能是轻便的,从而图像可被创建在与将最终运行该图像的计算机系统不同的计算机系统中,并且可随后被复制到将运行该图像的一个或多个计算机系统。这可允许虚拟化环境、应用主机、移动设备制造商或其他各方产生可被复制到任何特定设备的BPRAM上的快速加载应用和操作系统图像,以实现本文中描述的优点。

[0032] 静态分析组件130静态地分析软件二进制码、数据模块或其他所存储的软件数据以确定软件如何访问指令和数据。组件130可分析二进制代码(例如,汇编代码)、中间代码(例如,MICROSOFT™中间语言(IL)代码)或应用或操作系统的其它经编译或可运行版本。静态分析基本上在过去的几年中已有进展。系统100使用静态分析技术来特别关注应用用来存储代码、数据和其他资源的区域。静态分析组件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发现了对外部模块的调用,则组件130可以用访问BPRAM中的外部模块的图像的调用来替换该调用。通过这种方式,系统100可发现应用操作并修改允许使用BPRAM来加速应用和操作系统初始化的行为。组件130还可访问元数据,诸如可移植可执行文件(PE)头部或软件数据中描述特定代码和/或数据如何被正在分析的软件使用的其他信息。例如,PE格式允许特定部分被标记成可执行代码、只读数据、读/写数据、资源(例如,按钮、图标、对话框等)等等。

[0033] 动态分析组件140动态地分析正在运行的应用以收集与难以用静态分析来确定的该应用的行为有关的附加信息。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在编译后变成如此)。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、由应用使用的存储器的实际内容、和用户配置信息,对于这些内容在静态分析期间只能进行猜测或逼近。在一些情况下,该系统基于性能开销或其他已知的资源约束来确定执行静态分析、动态分析还是两者。动态分析可潜在地发现在静态分析期间没有被找到的应用行为。动态分析组件140还可使用动态分析来确认静态分析的结果。动态分析组件140将所确定的信息提供给布局创建组件150以标识适合布局在BPRAM中的特定应用代码和数

据。例如,动态布局组件150可确定特定数据(虽然被标记为读/写)实际上没有被应用改变。反之,动态分析还可标识应用的数据中被频繁改变的各部分。应用还可包含诸如指示应用代码是否为只读并因此适合存储在BPRAM中的自修改代码之类的行为。

[0034] 布局创建组件150创建用于基于由内核布局组件或应用布局组件所确定的布局来初始化操作系统或应用的BPRAM中的图像,并存储所创建的BPRAM中的图像。布局创建组件150可从一个或多个所存储的模块、数据存储等中访问数据、二进制代码和其他软件信息,以创建表示应用或操作系统的存储器内数据中用于执行该操作系统或应用的至少一部分的图像。布局创建组件150在运行操作系统或应用之前执行由当今加载器、堆和其他软件组件执行的许多步骤。例如,加载器通常负责每次应用加载时都检索硬盘驱动器上存储的二进制码模块并将那些模块的内容放置在字节可寻址的存储器堆中,而布局创建组件150执行类似的步骤一次,并将得到的存储器图像放置在BPRAM中,在BPRAM中每次该应用被请求时它都可以是准备好的,而无需每次都从较慢的存储中加载这些模块。这显著地加速了应用加载时间。

[0035] 损耗监视组件160可任选地监视BPRAM的使用信息以避免写入BPRAM达超过BPRAM的损耗规范的次数。BPRAM通常具有它可接受的有限次数的写入,并因此它有助于执行损耗平均和类似技术以确保BPRAM被均匀地使用(例如,每次当其他部分未被使用时,一个部分也不被使用)。此外,组件160可基于通过静态和/或动态分析所确定的数据的写频率来在哪些数据要存储在哪些设备中之间作出权衡。如果特定BPRAM设备可处理特定数据的写加载达合适长的生存期(例如,该设备可在特定的计算机系统中使用的5-10年),则系统100可将该数据存储在该位置处。对于具有较长生存期(例如,较多写循环)的BPRAM设备而言,系统100可选择将一些不被频繁写入的数据包括在BPRAM中,即使这些数据不完全是只读的。例如,应用配置参数可能很少改变,因此BPRAM是合适的。由于具有更长生存期的BPRAM随时间而发展,因此该系统可调适以将BPRAM用于更多数据。

[0036] 内核初始化组件170使用由布局创建组件创建的BPRAM中的图像来初始化操作系统。不是按传统的方式通过从基于扇区的持久存储中加载操作系统二进制代码和数据(诸如,旋转硬盘驱动器或固态硬盘)来引导操作系统,组件170直接从BPRAM图像或通过从BPRAM图像快速地复制到DRAM或其他存储器中来执行操作系统。由于由布局创建组件150创建的布局基于代码和数据在执行期间如何被使用而非代码和数据最初被存储在何处,因此从所存储的图像初始化操作系统要快得多。不是潜在地对较慢存储有许多调用,内核初始化组件170可利用字节可寻址性来从BPRAM直接执行操作系统,或可使用快速直接存储器存取(DMA)转移来将BPRAM图像快速复制到DRAM或其他存储器中。

[0037] 应用初始化组件180使用由布局创建组件创建的BPRAM中的图像来初始化应用。不是按传统的方式通过从基于扇区的持久存储中加载应用二进制代码和数据(诸如,旋转硬盘驱动器或固态硬盘)来初始化应用,组件180直接从BPRAM图像或通过从BPRAM图像快速地复制到DRAM或其他存储器中来加载应用。由于由布局创建组件150创建的布局基于代码和数据在执行期间如何被使用而非代码和数据最初被存储在何处,因此从所存储的图像初始化应用要快得多。不是潜在地对较慢存储有许多调用,应用初始化组件180可利用字节可寻址性来从BPRAM直接执行应用,或可使用快速直接存储器存取(DMA)转移来将BPRAM图像快速复制到DRAM或其他存储器中。

[0038] 其上实现软件布局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在计算机可读存储介质上。本文所要求的任何计算机可读介质包括仅那些落在法定可受专利保护的类别中的介质。该系统还可包括一个或多个通信链路,通过通信链路可传送数据。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。

[0039] 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。

[0040] 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。

[0041] 图2是示出一个实施例中的软件布局系统用于检查操作系统内核以标识要预布局在BPRAM中的代码和数据的处理的流程图。图2和图3示出该系统在被应用到操作系统内核时的两个不同的操作阶段。图2示出确定快速引导布局并将该布局复制到BPRAM的阶段,而图3示出从先前通过图2的步骤创建的BPRAM图像引导内核的阶段。

[0042] 在框210开始,该系统标识目标操作系统内核,对该目标操作系统将执行快速布局以创建该内核的快速引导图像,其中该内核具有包括在布局上与快速引导图像不同的一个或多个模块的存储格式。该系统可提供管理工具,该管理工具可由信息技术(IT)人员运行或者可作为操作系统的一部分而自动运行以标识该内核并启动创建快速引导图像布局以供存储在BPRAM中的过程。标识内核可包括检查引导记录或现有数据存储设备或其他数据结构、监视计算机系统的引导或接收指示目标内核的模块路径。

[0043] 在框220继续,该系统对所标识的目标操作系统内核执行静态分析以确定与该内核相关联的一个或多个静态和动态链接模块,并标识该内核所使用的不变到足以存储在字节可寻址的、持久随机存取存储器(BPRAM)中的数据。本领域中存在用于执行静态分析的各种技术,但该过程一般始于:加载与内核相关联的主模块、检查可执行文件格式以标识该模块的各部分(其可包括资源、恒定数据、引用其他模块的链接模块部分、以及一个或多个包含可执行代码的代码部分),并可能分解所标识出的任何机器代码或其他分析技术。在一些实施例中,该系统还可加载所引用的任何模块,诸如动态链接库、硬件的驱动程序或该内核所使用的其他资源。

[0044] 在框230继续,该系统可任选地通过以下方式来执行动态分析:执行操作系统内核并监视对该内核的初始化以标识与该内核相关联的附加模块和数据。动态分析可找到除通过静态分析找到的那些模块和/或数据以外的模块和/数据,以及提供对通过静态分析标识出的项的确认。动态分析可提供通过静态分析难以确定的信息,诸如内核所使用的动态分配的存储器结构、仅在某些条件下被动态加载的模块、该内核的基于特定硬件环境而不同

的各部分等等。在静态和动态分析期间,该系统收集关于该内核如何初始化的信息,该信息可用于产生该内核可从其直接执行从而跳过了传统初始化步骤中的至少一些步骤的预布局图像。

[0045] 在一些实施例中,该系统还捕捉标识被找到的资源中的每一个的签名信息。标识信息可包括文件系统时的戳、校验和、或指示特定资源何时已改变的其他值。该系统随后可使用此信息来确定何时(诸如在操作系统文件或驱动程序被来自制造商的更新更新以后)再次执行布局分析。该系统可在每一引导期间或在特定次数的引导后执行以下过程:遍历对该布局有贡献的每一已知资源、确定任何资源是否已改变、并且如果任何资源已改变则发起新的布局过程以更新BPRAM图像。

[0046] 在静态和动态分析期间,该系统还可观察影响外部资源(诸如,硬件设备)的任何初始化并对此作笔记。诸如图形卡、输入设备、显示器和其他设备之类的硬件可期望特定初始化序列以将该硬件置于供接收进一步请求的已知状态。尽管该软件布局系统可提前执行多个典型的初始化步骤,但影响外部资源的步骤在每次操作系统引导时都可能需要被执行。因此,该系统可跟踪这样的初始化例程并将指针或其他引用存储到这些例程中,以使得它们可在操作系统正从预布局的快速引导图像加载时被执行。

[0047] 在框240继续,该系统标识BPRAM设备,该BPRAM设备适于存储从其快速引导操作系统内核的预布局图像。BPRAM与先前可用的典型闪存、硬盘驱动器和动态RAM(DRAM)的不同之处至于BPRAM是字节可寻址的、但也是持久的。字节可寻址意味着不是从其一个扇区接一个扇区地加载(如对硬盘驱动器和像硬盘驱动器那样动作的闪存而言是典型的),而是BPRAM的每一字节都被直接寻址。典型地,计算机处理器可仅执行字节可寻址的存储器中存储的软件代码。持久意味着BPRAM在各功率循环(即,将设备的电源关闭并回过头再次打开)之间保持存储在那里的数据。尽管典型的DRAM是字节可寻址的但它不是持久的,并且尽管典型的硬盘驱动器是持久的但它们不是字节可寻址的。BPRAM因此是新的,因为它同时拥有这两个特质。因此,要被执行的一些事物的图像可在请求执行它之前被形成、被存储在BPRAM中、并随后从那里直接执行或在复制到另一存储器设备后执行。

[0048] 在框250继续,该系统可任选地确定与该BPRAM设备有关的损耗状态信息。虽然这可随时间用更好的设备来解决,但当前的BPRAM具有以下缺点:BPRAM可被成功写入仅达有限的次数。尽管该次数足以允许对BPRAM的不频繁更新,但它不足以允许与典型的DRAM设备或硬盘驱动器一样频繁向BPRAM进行写入。因此,该系统可跟踪损耗状态信息以管理对BPRAM设备的写入,并利用一种或多种技术来延长BPRAM设备的使用生存期。该信息可包括设备的每一区域已被写入了多少次、BPRAM的哪些部分当前在使用或不在使用、对每一区域可用的预测写入次数等等。基于损耗状态信息,该系统选择BPRAM内用于存储布局图像的位置。

[0049] 在框260继续,该系统基于内核的与该内核的存储格式不同的存储器内布局来创建快速引导图像的布局,其中该快速引导图像准备好执行并且是在接收执行内核的请求之前被创建的。内核在存储器中的典型图像是通过将存储的模块从硬盘驱动器存储加载到DRAM中来创建的,但仅在执行内核的请求之际被创建。在本文中,该系统提前执行这些步骤,并将结果放置在BPRAM中,从而通过不用每次都重复那些步骤来使得引导内核的将来请求更快。布局图像可包括一个或多个模块代码部分、一个或多个数据部分、由内核创建的一

个或多个只读堆区域等等的内容。

[0050] 在框270继续,该系统将所创建的布局图像复制到标识出的BPRAM设备中。BPRAM设备可包括该系统通过其将图像复制到该设备的特定写过程。该系统还可标识该内核内表示在该内核的所有典型初始化步骤之后的点的特定执行位置。由于布局图像以特定量的初始化已被执行的状态包含该内核,因此当该内核从BPRAM运行时,它可在与通常相比更后面的执行位置处启动。因此,该系统可盖写内核图像内的入口点地址,或在该内核从BPRAM运行时作出类似的修改以指令该内核从合适的点启动,由此跳过了典型的初始化步骤。在框270之后,这些步骤结束。

[0051] 图3是示出一个实施例中的软件布局系统用于从BPRAM初始化操作系统内核的处理的流程图。如先前所述,图3示出了从先前通过图2的步骤创建的BPRAM图像引导内核的阶段。

[0052] 在框310开始,该系统接收引导操作系统的请求。不同于通常在操作系统的控制下进行初始化的应用,操作系统通常在计算机系统的基本输入/输出系统(BIOS)、引导加载器、管理程序(在虚拟化的情况中)或其他低级引导代码的控制下进行初始化。低级引导代码可从主引导记录(MBR)、全局唯一标识符(GUID)分区表(GPT)或指示可用的可引导操作系统的其他存储数据结构中标识操作系统。一旦特定操作系统被标识出,该低级引导代码通常就标识该操作系统的主模块、加载该模块并将控制传递给该模块的入口点。在软件布局系统的情况下,低级引导代码可通过调用BPRAM图像的入口点来请求操作系统从BPRAM引导。

[0053] 在框320继续,该系统标识BPRAM内操作系统可从其执行而无需加载各个体模块的先前布局图像。与操作系统的被存储成在没有完成加载和初始化过程的情况下没有准备好执行的各个体模块的典型存储格式相反,布局图像准备好执行。加载和初始化过程将各模块映射在存储器中、在堆或其他存储器区域中创建数据区等等。相反,软件布局系统的执行阶段开始于先前在某个较早的时间已经完成的那些步骤。软件布局系统因此可直接移动至要么直接从BPRAM执行操作系统,要么通过将布局图像快速地从BPRAM复制到另一存储器设备并从那里执行操作系统来执行操作系统。

[0054] 在决策框330继续,如果该系统标识出了先前布局的图像则该系统在框340继续以执行快速引导,否则该系统完成并返回传统的引导过程以加载与操作系统相关联的存储模块。该系统还可确定(未示出)标识出的先前布局的图像相对于该操作系统的存储格式而言是否是最新的。如果该操作系统的模块的存储版本已被更新,则先前布局图像可能也需要被更新,并且该系统可在此时执行这样的更新或者可标记要在稍后的时间(例如,在下一次引导之前)执行更新的需要。

[0055] 在决策框340继续,该系统确定标识出的先前布局的图像是否可被就地执行,并且如果如此,则跳到框360以执行BPRAM图像,否则该系统在框350继续。即使BPRAM是字节可寻址的,仍存在在别处执行该图像的好理由。例如,如果BPRAM设备比DRAM更慢,则(例如,使用直接存储器存取(DMA)或其他方法来)执行到DRAM的快速复制并随后从那里执行可能更快。替换地或另选地,BPRAM图像可包含将不必要地损耗BPRAM的可写入部分,但可在DRAM中很好地执行而没有损耗顾虑。出于这些和其他理由,软件布局系统的特定实现可选择直接从BPRAM执行。

[0056] 在框350继续,该系统将标识出的该操作系统的先前布局的图像复制到另一存储设备以供执行。该复制可包括可加速复制操作的DMA或其他快速转移技术。

[0057] 在框360继续,该系统使用标识出的先前布局的图像来启动对该操作系统的执行。此时,该图像要么仍在BPRAM中并被直接执行,要么是从BPRAM到其他存储设备的副本。执行操作系统可包括:基于由于预布局过程而不再被需要的初始化步骤来标识操作系统的替换入口点,以及调用外部资源的任何初始化例程,这些初始化例程被执行以使外部资源与操作系统的通过提前执行布局而导致的状态一致。

[0058] 在框370继续,该系统使操作系统进入准备好状态以使用标识出的先前布局的图像来以与从基于扇区的存储设备上存储的各模块中加载操作系统的传统引导相比更快的时间执行应用。由于该系统提前执行引导过程的大部分,并且这些部分由于涉及与存储器相比相对更慢的存储设备而通常是最慢的,因此该引导过程可通过本文中描述的过程来显著地加速。在框370之后,这些步骤结束。

[0059] 图4是示出一个实施例中的软件布局系统用于检查应用以标识要预布局在BPRAM中的代码和数据的处理的流程图。图4和图5示出该系统在被应用到应用时的两个不同的操作阶段。图4示出确定应用的准备好执行布局并将该布局复制到BPRAM的阶段,而图5示出从先前通过图4的步骤创建的BPRAM图像初始化应用的阶段。

[0060] 在框410开始,该系统标识目标应用,对该目标应用将执行快速布局以创建该应用的快速初始化图像,其中该应用具有包括在布局上与快速初始化图像不同的一个或多个模块的存储格式。该系统可提供管理工具,该管理工具可由信息技术(IT)人员运行或可作为操作系统的一部分而自动运行,以标识该应用并启动创建快速初始化图像布局以供存储在BPRAM中的过程。标识应用可包括检查操作系统上安装的应用列表、截取执行应用的请求、监视已经运行的应用或接收指示目标应用的模块路径。

[0061] 在框420继续,该系统对标识出的目标应用执行静态分析以确定与该应用相关联的一个或多个静态和动态链接模块,并标识该应用所使用的不变到足以存储在字节可寻址的、持久随机存取存储器(BPRAM)中的数据。本领域中存在用于执行静态分析的各种技术,但该过程一般始于:加载与应用相关联的主模块、检查可执行文件格式以标识该模块的各部分(其可包括资源、恒定数据、引用其他模块的链接模块部分、以及一个或多个包含可执行代码的代码部分),并可能分解所标识出的任何机器代码或其他分析技术。在一些实施例中,该系统还可加载所引用的任何模块,诸如动态链接库、插件扩展模块或该应用所使用的其他资源。

[0062] 诸如文字处理器之类的应用通常允许第三方可扩展性以通过一个或多个插件模块来向该应用添加新功能,并且该系统可将所安装的任何插件包括在布局图像中。此外,各应用可使用一个或多个框架来构建,该一个或多个框架可包括该应用所依赖的一个或多个模块形式的运行时。例如,MICROSOFT™.NET包括各应用可用于公共数据结构和功能的公共语言运行时(CLR),并且这些公共数据结构和功能可与该应用一起被包括在BPRAM中,或以多个这样的应用都可利用的共同方式被包括在BPRAM中。

[0063] 在框430继续,该系统可任选地通过以下方式来执行动态分析:执行该应用并监视对该应用的初始化以标识与该应用相关联的附加模块和数据。动态分析可找到除通过静态分析找到的那些模块和/或数据以外的模块和/数据,以及提供对通过静态分析标识出的项

的确认。动态分析可提供通过静态分析难以确定的信息,诸如该应用所使用的动态分配的存储器结构、仅在某些条件下被动态加载的模块、该应用的基于特定硬件或操作系统环境而不同的各部分等等。在静态和动态分析期间,该系统收集关于该应用如何初始化的信息,该信息可用于产生该应用可从其直接执行从而跳过了传统的初始化步骤中的至少一些步骤的预布局图像。

[0064] 在一些实施例中,该系统还捕捉标识被找到的资源中的每一个的签名信息。标识信息可包括文件系统中的时戳、校验和、或指示特定资源何时已改变的其他值。该系统随后可使用此信息来确定何时(诸如在应用文件被来自厂商的更新更新以后)再次执行布局分析。该系统可在应用的每一次执行期间或在特定次数的执行后执行以下过程:遍历对该布局有贡献的每一已知资源、确定任何资源是否已改变、并且如果任何资源已改变则发起新的布局过程以更新BPRAM图像。

[0065] 在静态和动态分析期间,该系统还可观察影响外部资源(诸如,系统配置数据库)的任何初始化并对此作笔记。尽管该软件布局系统可提前执行多个典型的初始化步骤,但影响外部资源的步骤可在每次该应用初始化时都需要被执行。因此,该系统可跟踪这样的初始化例程并存储指向这些例程的指针或其他引用,使得它们可在该应用正从预布局的图像加载时被执行。

[0066] 在框440继续,该系统标识BPRAM设备,该BPRAM设备适于存储从其初始化该应用的预布局图像。BPRAM与先前可用的典型闪存、硬盘驱动器和动态RAM(DRAM)的不同之处在于BPRAM是字节可寻址的但还是持久的。字节可寻址意味着不是从其一个扇区接一个扇区地加载(如对硬盘驱动器和像硬盘驱动器那样动作的闪存而言是典型的),而是BPRAM的每一字节都可被直接寻址。典型地,计算机处理器仅可执行字节可寻址的存储器中存储的软件代码。持久意味着BPRAM在各功率循环(即,将设备的电源关闭并回过头再次打开)之间保持存储在那里的数据。尽管典型的DRAM是字节可寻址的但它不是持久的,并且尽管典型的硬盘驱动器是持久的但它们不是字节可寻址的。BPRAM因此是新的,因为它同时拥有这两个特质。因此,要被执行的一些事物的图像可在请求执行它之前被形成、被存储在BPRAM中、并随后从那里直接执行或在复制到另一存储器设备后执行。

[0067] 在框450继续,该系统可任选地确定与该BPRAM设备有关的损耗状态信息。虽然这可随时间用更好的设备来解决,但当前的BPRAM具有以下缺点:BPRAM可被成功写入仅达有限的次数。尽管该次数足以允许对BPRAM的不频繁更新,但它不足以允许与典型的DRAM设备或硬盘驱动器一样频繁向BPRAM进行写入。因此,该系统可跟踪损耗状态信息以管理对BPRAM设备的写入,并利用一种或多种技术来延长BPRAM设备的使用生存期。该信息可包括设备的每一区域已被写入了多少次、BPRAM的哪些部分当前在使用或不在使用、对每一区域可用的预测写入次数等等。基于损耗状态信息,该系统选择BPRAM内用于存储布局图像的位置。

[0068] 在框460继续,该系统基于该应用的与该应用的存储格式不同的存储器内布局来创建快速初始化图像的布局,其中该快速初始化图像准备好执行并且是在接收执行该应用的请求之前被创建的。应用在存储器中的典型图像是通过将存储的模块从硬盘驱动器存储加载到DRAM中来创建的,但仅在执行该应用的请求之际被创建。在本文中,该系统提前执行这些步骤,并将结果放置在BPRAM中,从而通过不用每次都重复那些步骤来使得执行该应用

的将来请求更快。布局图像可包括一个或多个模块代码部分、一个或多个数据部分、由该应用创建的一个或多个只读堆区域等等的内容。

[0069] 在框470继续,该系统将所创建的布局图像复制到标识出的BPRAM设备中。BPRAM设备可包括该系统通过其将图像复制到该设备的特定写过程。该系统还可标识该应用内表示在该应用的所有典型初始化步骤之后的点的特定执行位置。由于布局图像以特定量的初始化已被执行的状态包含该应用,因此当该应用从BPRAM运行时,它可在与通常相比更后面的执行位置处启动。因此,该系统可盖写该应用图像内的入口点地址,或在该应用从BPRAM运行时作出类似的修改以指令该应用从合适的点启动,由此跳过了典型的初始化步骤。在框470之后,这些步骤结束。

[0070] 图5是示出一个实施例中的软件布局系统用于从BPRAM初始化应用的处理的流程图。如先前所述,图5示出了从先前通过图4的步骤创建的BPRAM图像初始化应用的阶段。

[0071] 在框510开始,该系统接收运行应用的请求。应用典型地在操作系统的控制下进行初始化。操作系统可向用户提供可通过其来启动应用的壳以及应用可通过其来请求其他应用运行的应用程序接口(API)。操作系统还可包括在初始化操作系统之际或在特定事件发生时自动启动的服务或端口监控程序的概念。不管应用如何启动,该系统都会接收应用已被请求的通知。这可通过运行操作系统内挂钩加载器请求的设备驱动程序或一般响应于请求应用运行而执行的其他步骤来实现。在一些实施例中,软件布局系统被直接构建在操作系统内作为本机特征。

[0072] 在框520继续,该系统标识BPRAM内该应用可从其执行而无需加载各个体模块的先前布局图像。与应用的被存储成在没有完成加载和初始化过程的情况下没有准备好执行的各个体模块的典型存储格式相反,布局图像准备好执行。加载和初始化过程将各模块映射在存储器中、在堆或其他存储器区域中创建数据区等等。相反,软件布局系统的执行阶段开始于先前在某个较早的时间已经完成的那些步骤。软件布局系统因此可直接移动至要么直接从BPRAM执行该应用,要么通过将布局图像快速地从BPRAM复制到另一存储器设备并从那里执行应用来执行应用。

[0073] 在决策框530继续,如果该系统标识出了先前布局的图像则该系统在框540继续以执行该应用的快速初始化,否则该系统完成并返回传统的初始化过程以加载与该应用相关联的存储模块。该系统还可确定(未示出)标识出的先前布局的图像相对于该应用的存储格式而言是否是最新的。如果该应用的模块的存储版本已被更新,则先前布局的图像可能也需要被更新,并且该系统可在此时执行这样的更新或者可标记要在稍后的时间(例如,在应用的下一次执行之前)执行更新的需要。

[0074] 在决策框540继续,该系统确定标识出的先前布局的图像是否可被就地执行,并且如果如此,则跳到框560以执行BPRAM图像,否则该系统在框550继续。即使BPRAM是字节可寻址的,仍存在在别处执行图像的好理由。例如,如果BPRAM设备比DRAM更慢,则(例如,使用直接存储器存取(DMA)或其他方法来)执行到DRAM的快速复制并随后从那里执行可能更快。替换地或另选地,BPRAM图像可包含将不必要地损耗BPRAM的可写入部分,但可在DRAM中很好地执行而没有损耗顾虑。出于这些和其他理由,软件布局系统的特定实现可选择不直接从BPRAM执行。

[0075] 在框550继续,该系统将标识出的该应用的先前布局的图像复制到另一存储器设

备以供执行。该复制可包括可加速复制操作的DMA或其他快速转移技术。

[0076] 在框560继续,该系统使用标识出的先前布局的图像来启动对该应用的执行。此时,该图像要么仍在BPRAM中并被直接执行,要么是从BPRAM到其他存储器设备的副本。执行应用可包括:基于由于预布局过程而不再被需要的初始化步骤来标识该应用的替换入口点,以及调用外部资源的任何初始化例程,这些初始化例程被执行以使外部资源与该应用的通过提前执行布局而导致的状态一致。

[0077] 在框570继续,该系统使应用进入准备好状态以供用户以与从基于扇区的存储设备上存储的各模块中加载应用的传统初始化相比更快的时间使用标识出的先前布局的图像。由于该系统提前执行应用初始化过程的大部分,并且这些部分由于涉及与存储器相比相对更慢的存储设备而通常是最慢的,因此初始化过程可通过本文中描述的过程来显著地加速。在框570之后,这些步骤结束。

[0078] 图6是示出一个实施例中的软件布局系统所使用的各种类型的存储和数据的关系的框图。典型的计算机系统包括处理器610、非持久动态RAM 620和基于扇区的持久存储630。各文件被存储在基于扇区的持久存储630中,这些文件包括操作系统和应用可执行模块以及数据文件。在操作期间,计算机系统通过处理器610上执行的指令将操作系统文件和应用文件从基于扇区的持久存储630加载到动态RAM 640中。这些指令将这些可执行模块从其存储格式重新格式化成准备好由处理器来执行的格式。软件布局系统引入了一种附加类型的存储器,即BPRAM 640,该BPRAM 640即是持久的也是字节可寻址的(即,不是基于扇区的)。计算机系统可直接从BPRAM执行操作系统和应用,而无需首先经历从存储格式复制到准备好执行的格式的初始化过程。相反,BPRAM640已经包括准备好执行可应用的操作系统和/或应用的格式。

[0079] 在一些实施例中,软件布局系统包括与操作系统相关联的描述供执行的应用所使用的信息的经修改的安装器。与应用相关联的安装文件可包括提供关于操作系统的信息的清单以及用于创建从其快速执行该应用的合适布局图像的安装器。不像静态和动态分析(其仍可被执行),安装信息直接来自应用制造商,并允许制造商定制布局过程以更好地标识该应用的依赖性和其他布局考虑。

[0080] 在一些实施例中,软件布局系统使用BPRAM,该BPRAM具有基于扇区的区域和字节可寻址的区域的组合。虽然之前在本文中详细描述了字节可寻址的部分,但BPRAM可按允许基于扇区的部分和字节可寻址的部分共存的方式来分区。基于扇区的部分可允许存储应用或操作系统的以下部分:即无法被提前布局,但与较慢速的基于扇区的介质(例如,可旋转的硬盘驱动器)相比从基于扇区的BPRAM仍可更快速地加载的部分。

[0081] 在一些实施例中,软件布局系统截取来自多个应用(甚至那些没有使用本文中描述的全布局过程的应用)的调用,以从BPRAM提供至少一些引用应用的模块。例如,该系统可截取公共WIN32™ API调用GetProcAddress(获得过程地址),其返回从动态链接库(DLL)加载的功能的地址,并且不是加载该DLL并从该DLL的存储器映射的部分返回地址,而可改为返回指向BPRAM中已经以准备好执行的格式布局了DLL的区域的指针。类似地,该系统可截取内核代码以加载诸如设备驱动程序之类的模块,并将这些调用重定向到BPRAM。

[0082] 在一些实施例中,软件布局系统提供BPRAM的虚拟存储器地址到正在运行的应用的透明映射。对于很少被改变并被存储在BPRAM中的应用和/或操作系统数据而言,该系统

可将BPRAM中存储的各区域映射至标准的虚拟存储器地址空间。通过这种方式,使用指向该数据的指针的应用或操作系统不知道该指针所引用的数据是被存储在BPRAM、DRAM,还是别处,并继续如之前那样发挥作用但是是用潜在地在新位置处的数据来发挥作用。

[0083] 在一些实施例中,如果在快速初始化期间发生了任何错误,则软件布局系统通过返回传统的初始化过程来提供容错性。模块的更新、隐错或其他错误可使得快速初始化过程不能将操作系统或应用置于运行状态。在这样的情况下,该系统可返回传统初始化过程并从基于扇区的介质上存储的模块中加载操作系统或应用。

[0084] 在一些实施例中,软件布局系统允许应用的更多实例在特定计算机硬件上运行。由于应用的大部分被存储在BPRAM的可共享部分中,因此应用的每一新实例所消耗的资源量被大大降低。这意味着该应用的正在运行的多个实例不需要消耗相当大量的资源。对于一些应用来说,运行许多实例是常见的,并且运行甚至更多实例的能力具有很大的用处。

[0085] 从上文将会认识到,虽然在此已出于说明目的描述了软件布局系统的特定实施例,但是可以做出各种修改而不背离本发明的精神和范围。因此,本发明只受所附权利要求限制。

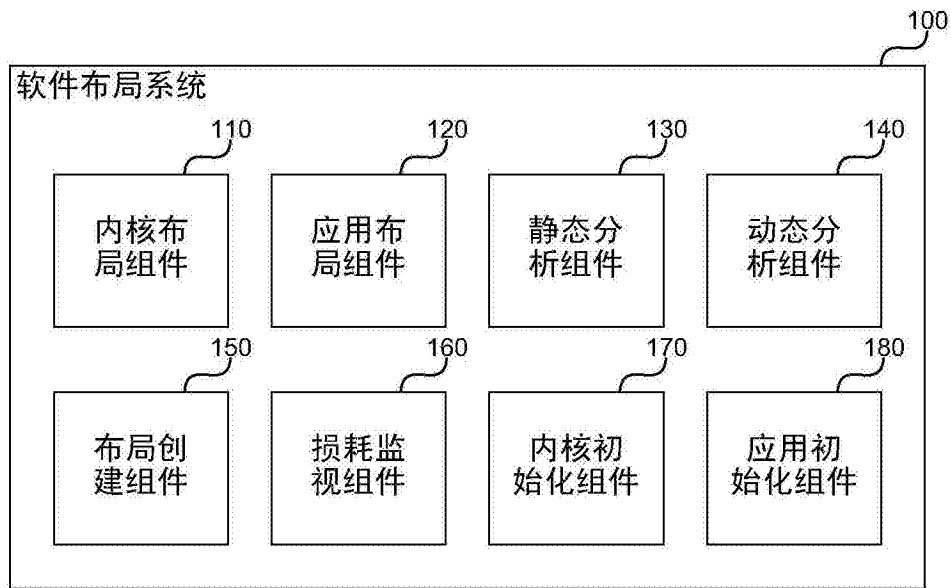


图1

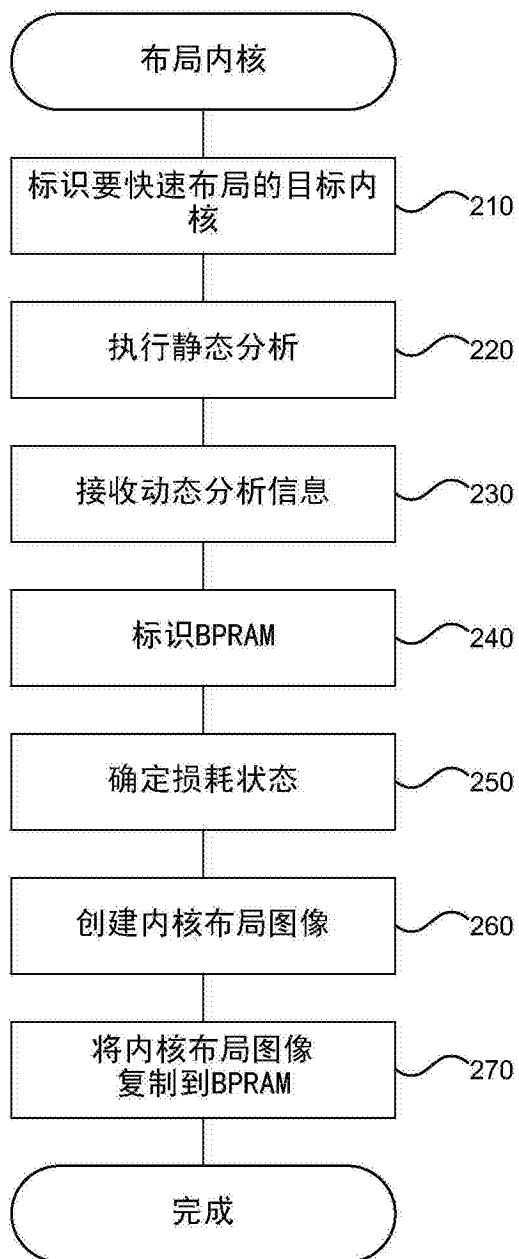


图2

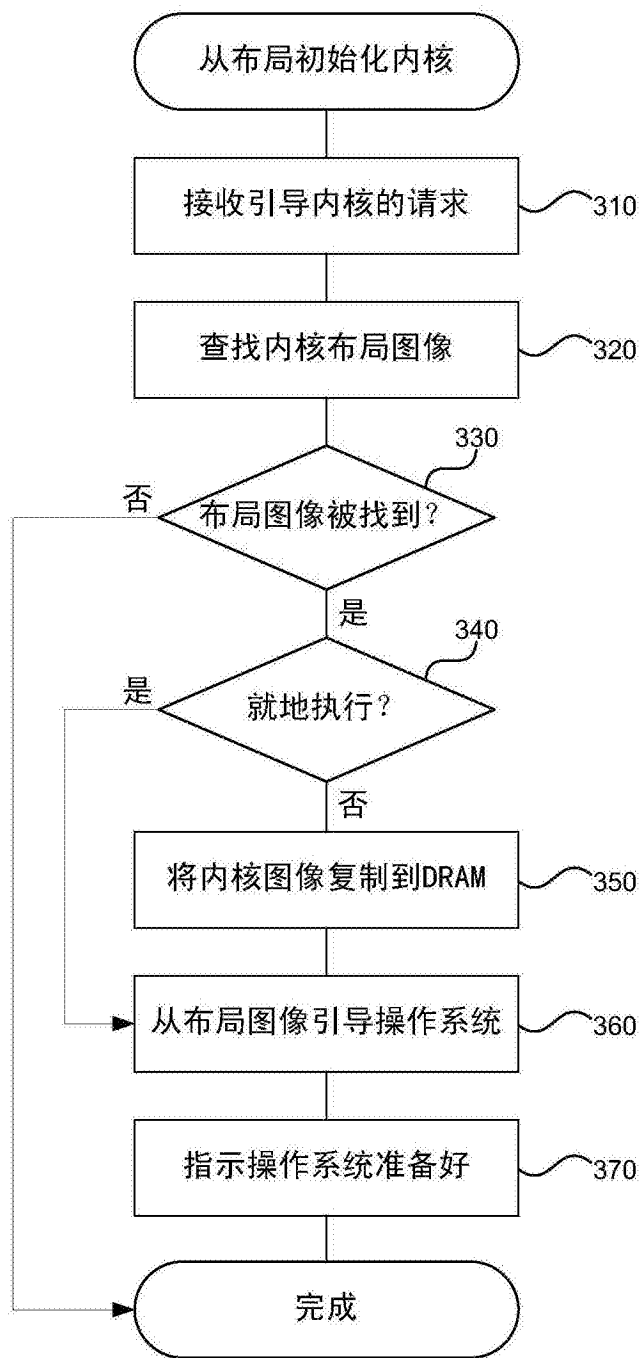


图3

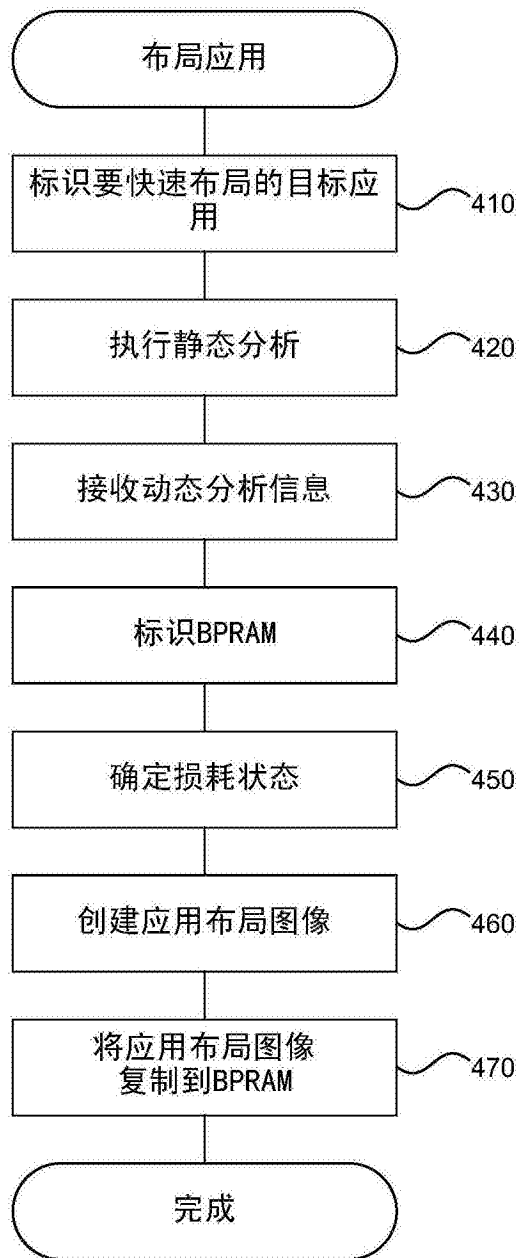


图4

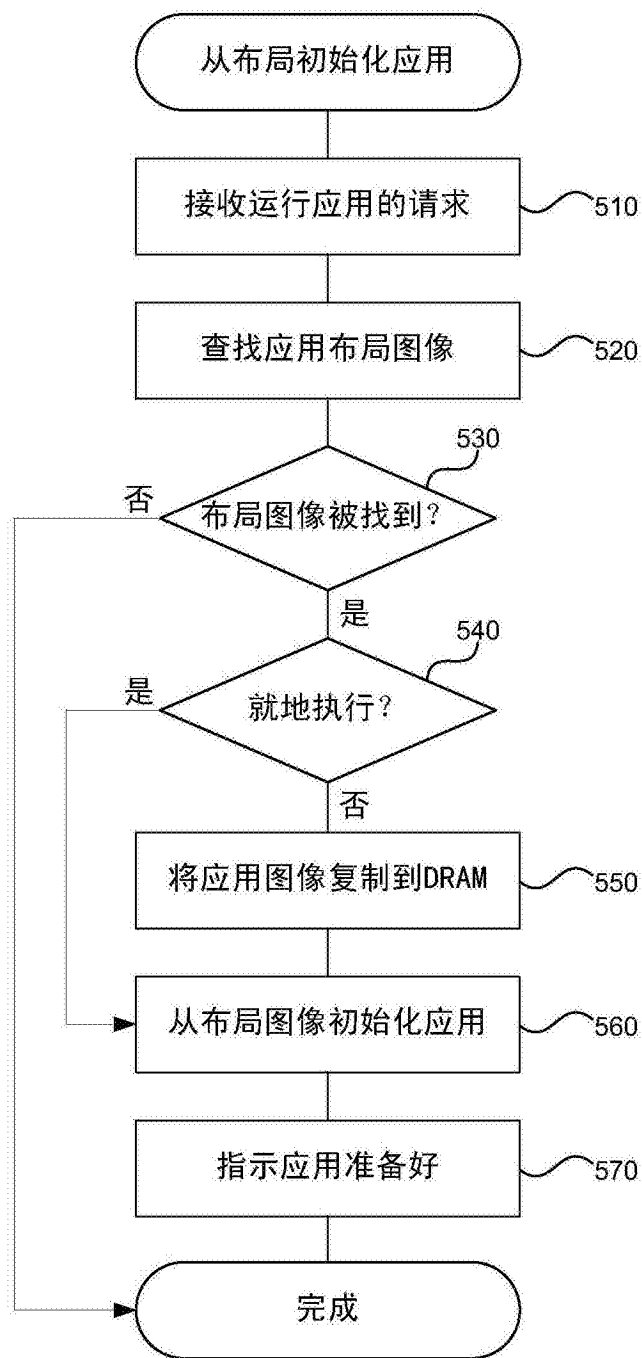


图5

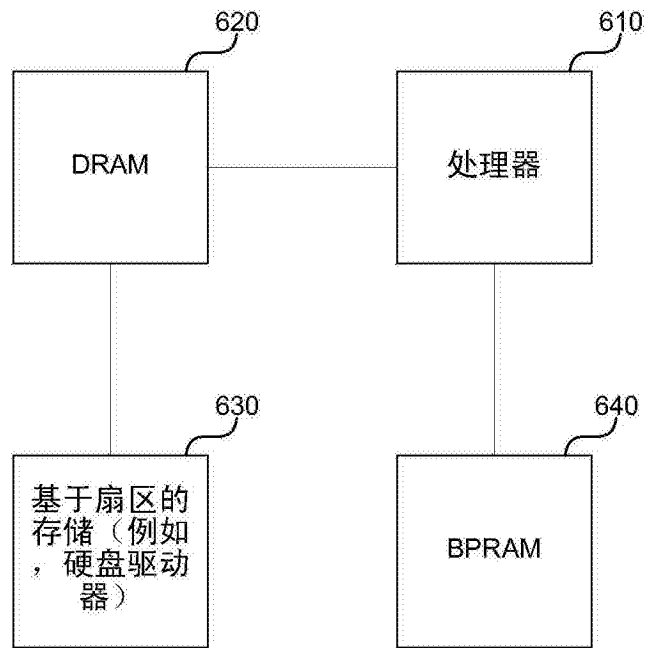


图6