(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 109388552 B
(45)授权公告日 2024.03.26

(21)申请号 201710667951.8

(22)申请日 2017.08.07

(65)同一申请的已公布的文献号
申请公布号 CN 109388552 A

(43)申请公布日 2019.02.26

(73)专利权人 中兴通讯股份有限公司
地址 518057 广东省深圳市南山区科技南路55号

(72)发明人 胡晓宇

(74)专利代理机构 北京天昊联合知识产权代理有限公司 11112
专利代理师 彭瑞欣 冯建基

(51)Int.Cl.
*G06F 11/36*(2006.01)
*G06F 9/48*(2006.01)

(56)对比文件
CN 106095489 A,2016.11.09
CN 105939225 A,2016.09.14
CN 103309740 A,2013.09.18
CN 105487779 A,2016.04.13
CN 106528216 A,2017.03.22
CN 106658051 A,2017.05.10
CN 102831045 A,2012.12.19
JP 2011234049 A,2011.11.17

审查员 李晓曼

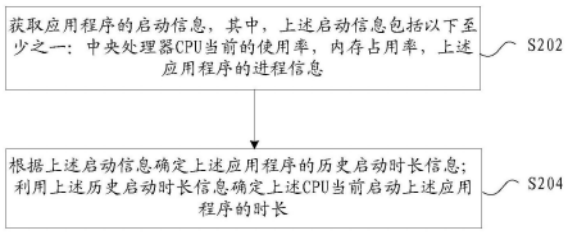
权利要求书2页 说明书8页 附图3页

(54)发明名称

# 启动应用程序的时长的确定方法、装置及存储介质

(57)摘要

本发明提供了一种启动应用程序的时长的确定方法、装置及存储介质，该方法包括：获取应用程序的启动信息，其中，启动信息包括以下至少之一：中央处理器CPU当前的使用率，内存占用率，应用程序的进程信息；根据启动信息确定应用程序的历史启动时长信息；利用历史启动时长信息确定CPU当前启动应用程序的时长。通过本发明，解决了相关技术中CPU在启动应用程序时，启动时间不准确的技术问题，达到更加准确的确定应用程序的启动时间的效果。




CN 109388552 B

1.一种启动应用程序的时长的确定方法,其特征在于,包括:

获取应用程序的启动信息,其中,所述启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,所述应用程序的进程信息;

根据所述启动信息确定所述应用程序的历史启动时长信息;

利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长;

利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长包括:

当所述历史启动时长信息的数目大于预定数目时,按照时长对所述历史启动时长信息采用聚类算法进行分类;

确定包括所述历史启动时长信息的数目最多的分类;

根据所述分类中的多个所述历史启动时长信息确定所述应用程序的平均启动时间或中心启动时间;

将所述平均启动时间或所述中心启动时间作为所述CPU当前启动所述应用程序的时长;

根据确定的所述CPU当前启动所述应用程序的时长控制所述CPU在预定频率以及预定核数上启动所述应用程序;

记录所述CPU当前启动所述应用程序的时长;

在所述平均启动时间或所述中心启动时间与所述CPU当前启动所述应用程序的时长的误差小于预设阈值的情况下,将所述平均启动时间或所述中心启动时间作为所述应用的固定启动时长。

2.根据权利要求1所述的方法,其特征在于,根据所述启动信息确定所述应用程序的历史启动时长信息包括:

在所述启动信息与记录的所述应用程序的历史启动时长信息匹配时,将所述历史启动时长作为所述历史启动时长信息。

3.根据权利要求1所述的方法,其特征在于,利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长包括:

当所述历史启动时长信息的数目小于或者等于预定数目时,读取所述应用程序启动时间的预设值;

将所述预设值作为所述CPU当前启动所述应用程序的时长。

4.一种启动应用程序的时长的确定装置,其特征在于,包括:

获取模块,用于获取应用程序的启动信息,其中,所述启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,所述应用程序的进程信息;

第一确定模块,用于根据所述启动信息确定所述应用程序的历史启动时长信息;

第二确定模块,用于利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长;

其中,第二确定模块包括:

分类单元,用于当所述历史启动时长信息的数目大于预定数目时,按照时长对所述历史启动时长信息采用聚类算法进行分类;

第一确定单元,用于确定包括所述历史启动时长信息的数目最多的分类;

第二确定单元,用于根据所述分类中的多个所述历史启动时长信息确定所述应用程序

的平均启动时间或中心启动时间;

第二处理单元,用于将所述平均启动时间或所述中心启动时间作为所述CPU当前启动所述应用程序的时长;根据确定的所述CPU当前启动所述应用程序的时长控制所述CPU在预定频率以及预定核数上启动所述应用程序;记录所述CPU当前启动所述应用程序的时长;在所述平均启动时间或所述中心启动时间与所述CPU当前启动所述应用程序的时长的误差小于预设阈值的情况下,将所述平均启动时间或所述中心启动时间作为所述应用的固定启动时长。

5.根据权利要求4所述的装置,其特征在于,所述第一确定模块包括:

第一处理单元,用于在所述启动信息与记录的所述应用程序的历史启动时长匹配时,将所述历史启动时长作为所述历史启动时长信息。

6.根据权利要求4所述的装置,其特征在于,第二确定模块还包括:

读取单元,用于当所述历史启动时长信息的数目小于或者等于预定数目时,读取所述应用程序启动时间的预设值;

第三处理单元,用于将所述预设值作为所述CPU当前启动所述应用程序的时长。

# 启动应用程序的时长的确定方法、装置及存储介质

## 技术领域

[0001] 本发明涉及通信领域,具体而言,涉及一种启动应用程序的时长的确定方法、装置及存储介质。

## 背景技术

[0002] 目前终端平台都使用多核中央处理器(Central Processing Unit,简称为CPU),CPU在线核数越多,工作频率越高,运行能力越强,但是功耗高,发热多;CPU在线核数越少,工作频率越低,运行能力越弱,但是功耗低,发热少。所以CPU工作频率和核数一般会根据负载及温度等因素做动态调频的。

[0003] 目前Android平台在应用冷启动时统一提高CPU频率,并持续一段时间,但是对于不同的应用启动时间不同,这样一刀切的配置,对于启动较快的应用会导致过多的电池消耗,降低续航时间,对于启动较慢的应用,CPU高频率持续时间太短,启动时间还是较长。另外,终端运行状态和应用实现方式对应用启动会有很大影响。

[0004] 针对上述中存在CPU在启动应用程序时,启动时间不准确的技术问题,相关技术中并未提出有效的解决方案。

## 发明内容

[0005] 本发明实施例提供了一种启动应用程序的时长的确定方法、装置及存储介质,以至少解决相关技术中CPU在启动应用程序时,启动时间不准确的技术问题。

[0006] 根据本发明的一个实施例,提供了一种启动应用程序的时长的确定方法,包括:获取应用程序的启动信息,其中,所述启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,所述应用程序的进程信息;根据所述启动信息确定所述应用程序的历史启动时长信息;利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长。

[0007] 可选地,根据所述启动信息确定所述应用程序的历史启动时长信息包括:在所述启动信息与记录的所述应用程序的历史启动时长匹配时,将所述历史启动时长作为所述历史启动时长信息。

[0008] 可选地,利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长包括:当所述历史启动时长信息的数目大于预定数目时,按照时长对所述历史启动时长信息进行分类;确定包括所述历史启动时长信息的数目最多的分类;根据所述分类中的多个所述历史启动时长信息确定所述应用程序的平均启动时间或中心启动时间;将所述平均启动时间或所述中心启动时间作为所述CPU当前启动所述应用程序的时长。

[0009] 可选地,利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长包括:当所述历史启动时长信息的数目小于或者等于预定数目时,读取所述应用程序启动时间的预设值;将所述预设值作为所述CPU当前启动所述应用程序的时长。

[0010] 可选地,利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长之后,所述方法还包括:根据确定的所述CPU当前启动所述应用程序的时长控制所述CPU在

预定频率以及预定核数上启动所述应用程序。

[0011] 可选地,利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长之后,所述方法还包括:记录所述CPU当前启动所述应用程序的时长。

[0012] 根据本发明的另一个实施例,还提出一种启动应用程序的时长的确定装置,包括:获取模块,用于获取应用程序的启动信息,其中,所述启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,所述应用程序的进程信息;第一确定模块,用于根据所述启动信息确定所述应用程序的历史启动时长信息;第二确定模块,用于利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长。

[0013] 可选地,所述第一确定模块包括:第一处理单元,用于在所述启动信息与记录的所述应用程序的历史启动时长匹配时,将所述历史启动时长作为所述历史启动时长信息。

[0014] 可选地,第二确定模块包括:分类单元,用于当所述历史启动时长信息的数目大于预定数目时,按照时长对所述历史启动时长信息进行分类;第一确定单元,用于确定包括所述历史启动时长信息的数目最多的分类;第二确定单元,用于根据所述分类中的多个所述历史启动时长信息确定所述应用程序的平均启动时间或中心启动时间;第二处理单元,用于将所述平均启动时间或所述中心启动时间作为所述CPU当前启动所述应用程序的时长。

[0015] 可选地,第二确定模块包括:读取单元,用于当所述历史启动时长信息的数目小于或者等于预定数目时,读取所述应用程序启动时间的预设值;第三处理单元,用于将所述预设值作为所述CPU当前启动所述应用程序的时长。

[0016] 可选地,所述装置还包括:控制模块,用于在利用所述历史启动时长信息确定所述CPU当前启动所述应用程序的时长之后,根据确定的所述CPU当前启动所述应用程序的时长控制所述CPU在预定频率以及预定核数上启动所述应用程序。

[0017] 根据本发明的又一个实施例,还提供了一种存储介质,所述存储介质包括存储的程序,其中,所述程序运行时执行上述任一项所述的方法。

[0018] 根据本发明的又一个实施例,还提供了一种处理器,所述处理器用于运行程序,其中,所述程序运行时执行上述任一项所述的方法。

[0019] 通过本发明,获取应用程序的启动信息后,其中,启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,应用程序的进程信息;根据启动信息确定应用程序的历史启动时长信息;利用历史启动时长信息确定CPU当前启动应用程序的时长。从而可以根据确定的时长启动应用程序,解决了相关技术中CPU在启动应用程序时,启动时间不准确的技术问题,达到更加准确的确定应用程序的启动时间的效果。

## 附图说明

[0020] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0021] 图1是本发明实施例的一种启动应用程序的时长的确定方法的移动终端的硬件结构框图;

[0022] 图2是根据本发明实施例的启动应用程序的时长的确定方法的流程图;

[0023] 图3是本实施例中的处理器频率控制的装置示意图;

[0024] 图4是本实施例中的处理器频率控制方法的流程图;

[0025] 图5是根据本发明实施例的启动应用程序的时长的确定装置的结构框图。

## 具体实施方式

[0026] 下文中将参考附图并结合实施例来详细说明本发明。需要说明的是,在不冲突的情况下,本申请中的实施例及实施例中的特征可以相互组合。

[0027] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。

[0028] 本申请实施例所提供的方法实施例可以在移动终端、计算机终端或者类似的运算装置中执行。以运行在移动终端上为例,图1是本发明实施例的一种启动应用程序的时长的确定方法的移动终端的硬件结构框图。如图1所示,移动终端10可以包括一个或多个(图1中仅示出一个)处理器102(处理器102可以包括但不限于微处理器MCU或可编程逻辑器件FPGA等的处理装置)、用于存储数据的存储器104、以及用于通信功能的传输装置106。本领域普通技术人员可以理解,图1所示的结构仅为示意,其并不对上述电子装置的结构造成限定。例如,移动终端10还可包括比图1中所示更多或者更少的组件,或者具有与图1所示不同的配置。

[0029] 存储器104可用于存储应用软件的软件程序以及模块,如本发明实施例中的启动应用程序的时长的确定方法对应的程序指令/模块,处理器102通过运行存储在存储器104内的软件程序以及模块,从而执行各种功能应用以及数据处理,即实现上述的方法。存储器104可包括高速随机存储器,还可包括非易失性存储器,如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中,存储器104可进一步包括相对于处理器102远程设置的存储器,这些远程存储器可以通过网络连接至移动终端10。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

[0030] 传输装置106用于经由一个网络接收或者发送数据。上述的网络具体实例可包括移动终端10的通信供应商提供的无线网络。在一个实例中,传输装置106包括一个网络适配器(Network Interface Controller,NIC),其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中,传输装置106可以为射频(Radio Frequency,RF)模块,其用于通过无线方式与互联网进行通讯。

[0031] 在本实施例中提供了一种启动应用程序的时长的确定方法,图2是根据本发明实施例的启动应用程序的时长的确定方法的流程图,如图2所示,该流程包括如下步骤:

[0032] 步骤S202,获取应用程序的启动信息,其中,上述启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,上述应用程序的进程信息;

[0033] 步骤S204,根据上述启动信息确定上述应用程序的历史启动时长信息;利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长。

[0034] 通过上述步骤,获取应用程序的启动信息后,其中,启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,应用程序的进程信息;根据启动信息确定应用程序的历史启动时长信息;利用历史启动时长信息确定CPU当前启动应用程序的时长。从而可以根据确定的时长启动应用程序,解决了相关技术中CPU在启动应用程序时,启动时间不准确的技术问题,达到更加准确的确定应用程序的启动时间的效果。

[0035] 可选地,上述步骤的执行主体可以为终端等,但不限于此。

[0036] 在本实施例中,上述中的内存占用率是指整个设备的内存占用率。

[0037] 在一个可选的实施例中,根据上述启动信息确定上述应用程序的历史启动时长信息包括:在上述启动信息与记录的应用程序的历史启动时长匹配时,将上述历史启动时长作为上述历史启动时长信息。在本实施例中,上述中的启动时间包括的信息还可以包括其他的信息,并不限于上述中的信息。

[0038] 在一个可选的实施例中,利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长包括:当上述历史启动时长信息的数目大于预定数目时,按照时长对上述历史启动时长信息进行分类;确定包括上述历史启动时长信息的数目最多的分类;根据上述分类中的多个上述历史启动时长信息确定上述应用程序的平均启动时间或中心启动时间;将上述平均启动时间或上述中心启动时间作为上述CPU当前启动上述应用程序的时长。在本实施例中,可以根据聚类算法进行分类,或者是根据其他的算法分类。

[0039] 在一个可选的实施例中,利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长包括:当上述历史启动时长信息的数目小于或者等于预定数目时,读取上述应用程序启动时间的预设值;将上述预设值作为上述CPU当前启动上述应用程序的时长。在本实施例中,上述中的预设值可以是启动应用程序时,系统的默认值,也可以进行设定。

[0040] 在一个可选的实施例中,利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长之后,上述方法还包括:根据确定的上述CPU当前启动上述应用程序的时长控制上述CPU在预定频率以及预定核数上启动上述应用程序。在本实施例中,预定频率可以是2GHZ,预定核数可以是8核。

[0041] 在一个可选的实施例中,利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长之后,上述方法还包括:记录上述CPU当前启动上述应用程序的时长。在本实施例中,记录的时长用于之后启动应用程序的参考信息。

[0042] 下面结合具体实施例对本发明进行详细说明:

[0043] 具体实施例1:

[0044] 本实施例描述了一种处理器频率控制的方法,在应用程序冷启动之前,获取当前的CPU使用率,内存使用率,及应用相关进程是否启动等信息,根据这些条件,分析该条件下历史启动时间,应用聚类算法,把数据分为n类,对于数据最多的类,计算其平均值,并设置为本次应用冷启动时,CPU一定数目核的核打开,并设置CPU在某个固定频率的持续时间。本实施例结合了影响应用启动的主要因素,并应用聚类算法,使应用启动时间估算更加准确,这样既可以提升用户体验,又可以有效的控制功耗。

[0045] 终端开机后,应用启动时,后台没有该应用的进程,这是系统会重新创建一个新的进程分配给应用,这种启动方式为冷启动。如果应用启动时后台已经有该应用的进程,从已有进程来启动,这种启动方式为热启动。在android系统上应用冷启动时,系统会从zygote进程中分开fork创建出一个新的进程分配给该应用,之后会依次创建应用Application类,创建主活动MainActivity类等,整个过程对CPU资源消耗较多,此时打开CPU一定数量的核,并使CPU工作某个较高的频率,即可使应用冷启动时间大大缩短。本实施例即用于此类场景。

[0046] 下面对zygote进程进行说明:在Android系统中,所有的应用程序进程以及系统服务进程SystemServer都是由Zygote进程分支(fork)出来的。在Java中,不同的虚拟机实例

会为不同的应用分配不同的内存。假如Android应用应该尽可能快地启动,但如果Android系统为每一个应用启动不同的Dalvik虚拟机实例,就会消耗大量的内存以及时间。Zygote让Dalvik虚拟机共享代码、低内存占用以及最小的启动时间成为可能。Zygote是一个虚拟器进程,在系统引导的时候启动。Zygote预加载以及初始化核心库类。通常,这些核心类一般是只读的,也是Android SDK或者核心框架的一部分。在Java虚拟机中,每一个实例都有它自己的核心库类文件和堆对象的拷贝。

[0047] 图3是本实施例中的处理器频率控制的装置示意图,如图3所示,本系统包括处理器S301,内部存储器S302,外部存储器S303,总线S304。

[0048] 总线S304用来连接处理器S301,内部存储器S302,外部存储器S303。

[0049] 外部存储器S303保存了应用程序执行文件;应用名称,应用历次冷启动时间;应冷启动预设默认值。

[0050] 应用启动时处理器S301读取外部存储器S303里的应冷启动预设默认值或该应用对应的历次冷启动的时间,这些值会暂存在内部存储器S302里,处理器S301把该时间值设置给处理器S301的对应寄存器。

[0051] 图4是本实施例中的处理器频率控制方法的流程图,如图4所示,流程具体为:

[0052] 步骤S401,用户冷启动某个应用。

[0053] 步骤S402,获取当前设备的CPU和内存占用率,并获取和该应用相关的进程数等可能影响应用冷启动的数据。

[0054] 步骤S403,读取当前条件对应的历史启动时长数据,并与要分类的分类数对比,如果要分为n类,则至少需要n+1条数据才有效,其中,n大于或者等于1。

[0055] 步骤S404,如果历史时长数据不足n+1条,则使用默认配置idea启动时间。

[0056] 步骤S405,读取对应条件下的所有历史时长数据。

[0057] 步骤S406,把读取的数据分类。

[0058] 步骤S407,取数据最多的一类,并计算此类数据的均值或者中心点。

[0059] 步骤S408,把S407计算的时长平均值设置为CPU在某些核上特殊频率上运行的时长。

[0060] 步骤S409,应用开始启动。

[0061] 步骤S410,记录应用本次的冷启动时长。

[0062] 具体实施例2:

[0063] 本实施例以冷启动为例进行详细说明,具体包括以下步骤:

[0064] 步骤S101,用户冷启动某个应用,常见的方式有点击触摸屏或者按键。

[0065] 步骤S102,一般应用启动速度主要与CPU占用率,内存占用率等有关系,在Android上,某些应用会有后台服务,这些进程在开机时就已经启动,应用冷启动时依赖于这些进程,但是如果这些后台进程没有被启动,或者已经被杀死,这样应用启动的时候,就要先启动该服务,然后才启动主线程,这样应用启动耗时就会很长,一般进程名的前几个信息是一样的。后台依赖服务会对启动速度造成明显影响,因此必须统计后台进程的情况。

[0066] 步骤S103,根据步骤S102中的条件,查找符合此条件的所有历史数据,看历史数据是否满足分类需要,比如要分为n类,最好有n+1个数据,这样方便选出数据最多的分类。

[0067] 步骤S104,如果数据个数不满足n+1,则读取事先配置的默认时长值,依次数据,在

S108配置为CPU工作在特殊条件的时长,然后启动应用。

[0068] 步骤S105,把S103读取的历史数据,通过聚类算法进行分类,可以分为n类,比如3类。聚类算法较多,可以选用K-MEANS算法。首先从n个数据对象任意选择k个对象作为初始聚类中心;而对于所剩下其它对象,则根据它们与这些聚类中心的相似度(距离),分别将它们分配给与其最相似的(聚类中心所代表的)聚类;然后再计算每个所获新聚类的聚类中心(该聚类中所有对象的均值);不断重复这一过程直到标准测度函数开始收敛为止。也可以选用K-MEDOIDS,它对于脏数据和异常数据不敏感,但是计算量较K-MEANS算法大。

[0069] 步骤S107,分类完成后,把分到数据最多的类,计算分类所有数据的平均值,把它作为S108中要配置的时长,

[0070] 也可以求分类中数据的中心点,这个点到其他所有点的距离之和最小,这样对于脏数据和异常数据不敏感。

[0071] 步骤S108,把S107种计算得到值,作为CPU将要运行在某个特殊配置的时长,根据CPU类型不同,可以配置要运行的核数,频率等参数。

[0072] 步骤S109,开始正常的冷应用启动流程。

[0073] 步骤S110,记录本次应用启动的时长,为下次应用冷启动时长做参考。

[0074] 对于此方法,也可以用于对已知应用做训练,并对S108里设置的时长,和S110实际启动时长,做对比,直到两个值满足一定误差后,认为已经获取到确定值,以此时长,作为此类条件下的固定启动时长配置到正式版本里,这样可以节约分类计算时间。

[0075] 综上所述,通过上述中的对CPU运行时间进行的确定,可以提升应用启动速度。并且可以降低功耗,减少发热。

[0076] 上述中的实施例在其它嵌入式设备上也可以应用。并且,可以使用其它新的算法来统计预估应用启动时CPU在最高频工作的时间。

[0077] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,或者网络设备等)执行本发明各个实施例所述的方法。

[0078] 在本实施例中还提供了一种启动应用程序的时长的确定装置,该装置用于实现上述实施例及优选实施方式,已经进行过说明的不再赘述。如以下所使用的,术语“模块”可以实现预定功能的软件和/或硬件的组合。尽管以下实施例所描述的装置较佳地以软件来实现,但是硬件,或者软件和硬件的组合的实现也是可能并被构想的。

[0079] 图5是根据本发明实施例的启动应用程序的时长的确定装置的结构框图,如图5所示,该装置包括:获取模块502,第一确定模块504和第二确定模块506。下面对该装置进行详细说明。

[0080] 获取模块502,用于获取应用程序的启动信息,其中,上述启动信息包括以下至少之一:中央处理器CPU当前的使用率,内存占用率,上述应用程序的进程信息;第一确定模块504,连接至上述中的获取模块502,用于根据上述启动信息确定上述应用程序的历史启动时长信息;第二确定模块506,连接至上述中的第一确定模块504,用于利用上述历史启动时

长信息确定上述CPU当前启动上述应用程序的时长。

[0081] 在一个可选的实施例中,上述第一确定模块504包括:第一处理单元,用于在上述启动信息与记录的历史启动时长匹配时,将上述历史启动时长作为上述历史启动时长信息。

[0082] 在一个可选的实施例中,第二确定模块506包括:分类单元,用于当上述历史启动时长信息的数目大于预定数目时,按照时长对上述历史启动时长信息进行分类;第一确定单元,用于确定包括上述历史启动时长信息的数目最多的分类;第二确定单元,用于根据上述分类中的多个上述历史启动时长信息确定上述应用程序的平均启动时间或中心启动时间;第二处理单元,用于将上述平均启动时间或上述中心启动时间作为上述CPU当前启动上述应用程序的时长。

[0083] 在一个可选的实施例中,第二确定模块506包括:读取单元,用于当上述历史启动时长信息的数目小于或者等于预定数目时,读取上述应用程序启动时间的预设值;第三处理单元,用于将上述预设值作为上述CPU当前启动上述应用程序的时长。

[0084] 在一个可选的实施例中,上述装置还包括:控制模块,用于在利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长之后,根据确定的上述CPU当前启动上述应用程序的时长控制上述CPU在预定频率以及预定核数上启动上述应用程序。

[0085] 在一个可选的实施例中,上述装置还包括:记录模块,用于在利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长之后,记录上述CPU当前启动上述应用程序的时长。

[0086] 需要说明的是,上述各个模块是可以通过软件或硬件来实现的,对于后者,可以通过以下方式实现,但不限于此:上述模块均位于同一处理器中;或者,上述各个模块以任意组合的形式分别位于不同的处理器中。

[0087] 本发明的实施例还提供了一种存储介质,该存储介质包括存储的程序,其中,上述程序运行时执行上述任一项所述的方法。

[0088] 可选地,在本实施例中,上述存储介质可以被设置为存储用于执行以上各步骤的程序代码。

[0089] 可选地,在本实施例中,上述存储介质可以包括但不限于:U盘、只读存储器(Read-Only Memory,简称为ROM)、随机存取存储器(Random Access Memory,简称为RAM)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0090] 本发明的实施例还提供了一种处理器,该处理器用于运行程序,其中,该程序运行时执行上述任一项方法中的步骤。

[0091] 可选地,本实施例中的具体示例可以参考上述实施例及可选实施方式中所描述的示例,本实施例在此不再赘述。

[0092] 显然,本领域的技术人员应该明白,上述的本发明的各模块或各步骤可以用通用的计算装置来实现,它们可以集中在单个的计算装置上,或者分布在多个计算装置所组成的网络上,可选地,它们可以用计算装置可执行的程序代码来实现,从而,可以将它们存储在存储装置中由计算装置来执行,并且在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤,或者将它们分别制作成各个集成电路模块,或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样,本发明不限制于任何特定的硬件和软件结合。

[0093] 以上所述仅为本发明的优选实施例而已，并不用于限制本发明，对于本领域的技术人员来说，本发明可以有各种更改和变化。凡在本发明的原则之内，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。

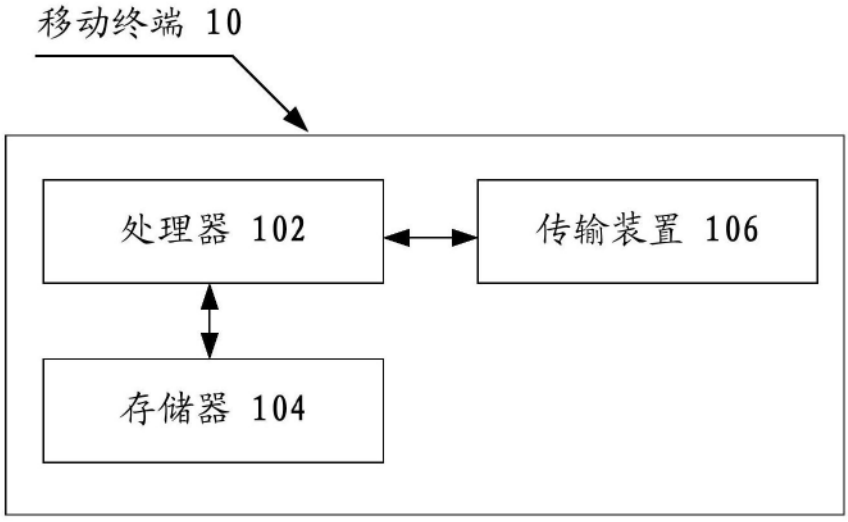

移动终端 10
处理器 102
传输装置 106
存储器 104

图1

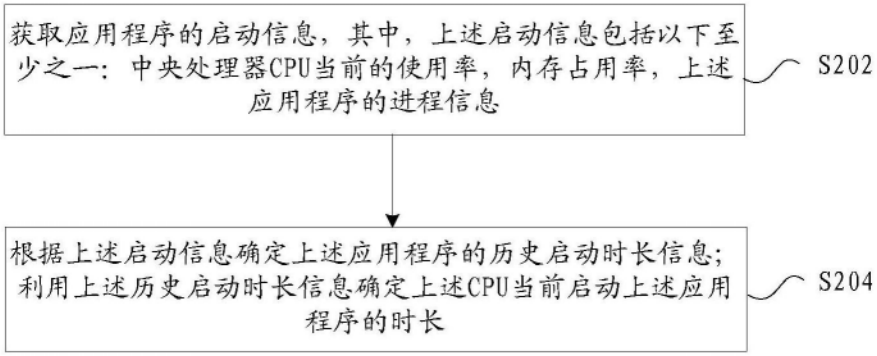

获取应用程序的启动信息，其中，上述启动信息包括以下至少之一：中央处理器CPU当前的使用率，内存占用率，上述应用程序的进程信息
S202
根据上述启动信息确定上述应用程序的历史启动时长信息；利用上述历史启动时长信息确定上述CPU当前启动上述应用程序的时长
S204

图2

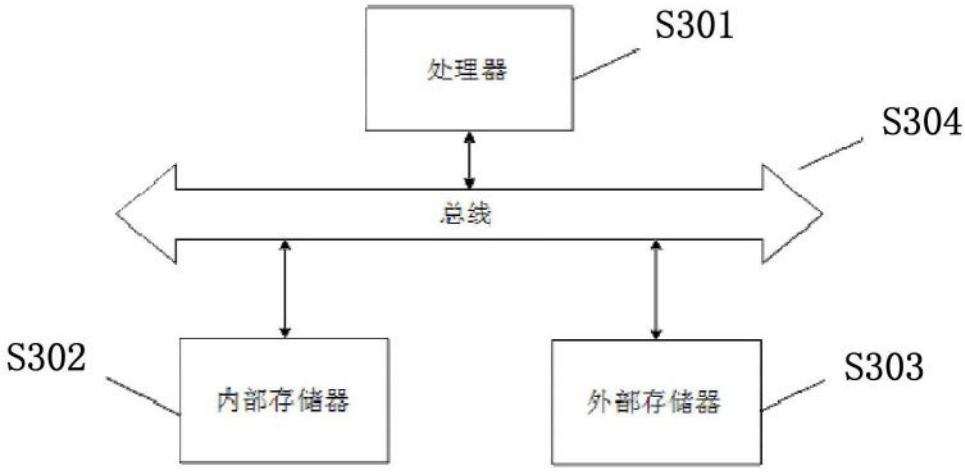

处理器
S301
S304
总线
S302
内部存储器
外部存储器
S303

图3

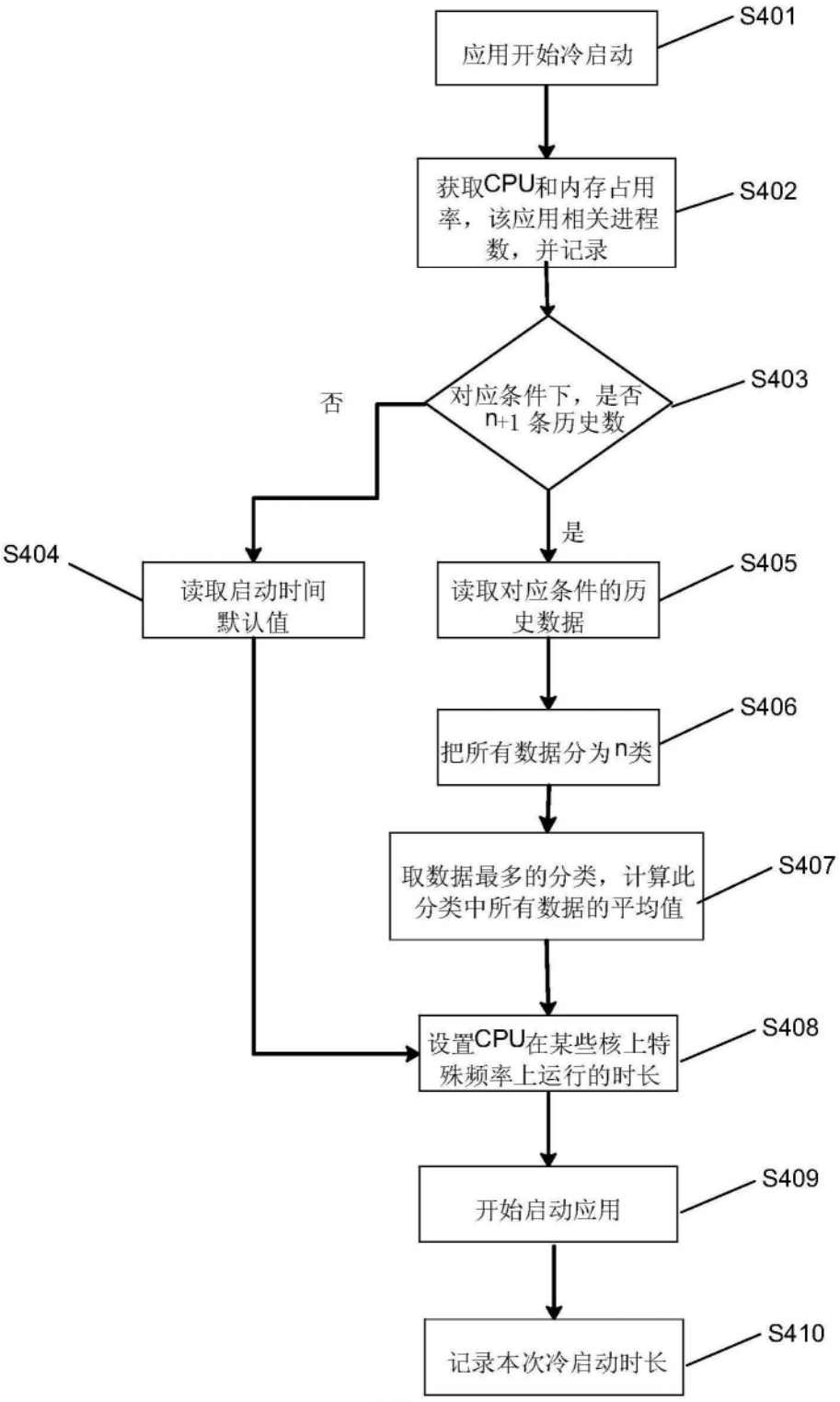

S401
应用开始冷启动
S402
获取CPU和内存占用率，该应用相关进程数，并记录
S403
对应条件下，是否n+1 条历史数
否
是
S404
读取启动时间默认值
S405
读取对应条件的历史数据
S406
把所有数据分为n类
S407
取数据最多的分类，计算此分类中所有数据的平均值
S408
设置CPU在某些核上特殊频率上运行的时长
S409
开始启动应用
S410
记录本次冷启动时长

图4

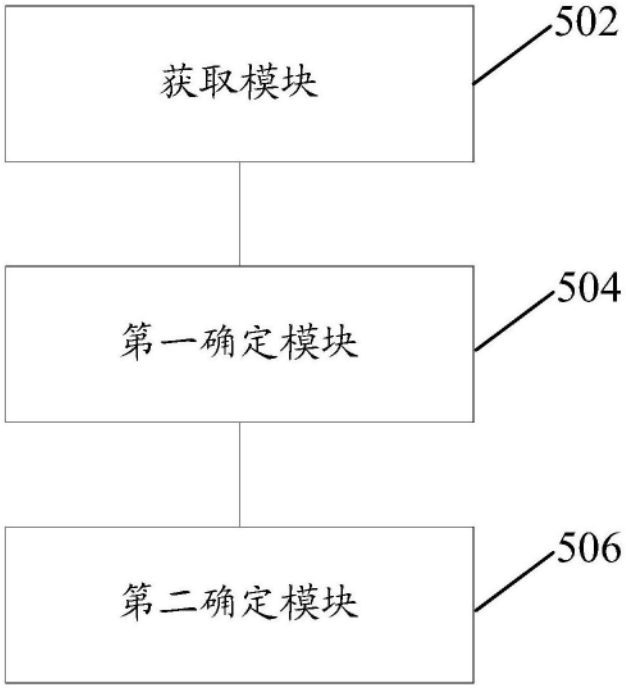



图5