

应用崩溃处理方法、装置、电子设备和可读存储介质

本申请要求于 2024 年 2 月 7 日递交的中国专利申请号为 2024101725789，发明名称为“应用崩溃处理方法、装置、电子设备和可读存储介质”的优先权，在此全文引用上述中国专利申请公开的内容以作为本申请的一部分。

5 技术领域

本申请涉及计算机技术领域，具体而言，涉及一种应用崩溃处理方法、装置、电子设备和可读存储介质。

背景技术

10 如果应用的启动阶段开关打开后发生了崩溃，则应用则不能够从配置平台获得新配置。如果崩溃不及时处理可能会导致应用 在启动时反复发生崩溃，更严重的可能会造成非常严重的用户流失。

发明内容

本申请的目的在于提供一种应用崩溃处理方法、装置、电子设备和可读存储介质，能够改善应用启动阶段频繁发生崩溃的情况。

15 第一方面，本发明提供一种应用崩溃处理方法，包括：在应用启动时，获取所述应用的标志位信息，其中，所述标志位信息通过预设埋点在所述应用启动时被读取；若所述应用的标志位信息为第一值，将所述应用由当前配置切换至历史配置，执行所述历史配置，其中，所述标志位信息为第一值表征所述应用在上次启动阶段发生崩溃。

20 在本申请实施例的方法中，通过设置标志位信息可以在启动时发现前一次启动是否发生过崩溃，如果发生过崩溃可以及时启动切换配置，执行历史配置，从而可以降低再次发生崩溃的概率，提高用户体验。

25 在可选的实施方式中，所述方法还包括：通过应用监控识别引起所述应用在上次启动阶段发生崩溃的第一功能模块；向配置平台发送所述第一功能模块的处理请求，所述处理请求用于指示所述配置平台停止下发所述第一功能模块对应的配置项。

上述实施方式中，可以及时关闭发生崩溃的功能模块，可以避免该功能模块再次引起崩溃，降低应用的崩溃频率，也能够提高用户体验。

30 在可选的实施方式中，所述方法还包括：针对在上次启动阶段发生崩溃的所述应用所获得的配置中对应的第二功能模块，在等待限制时限后执行所述第二功能模块的最新配置，其中，所述第二功能模块为上次启动阶段获得的配置中对应的更新功能模块，且所述第二功能模块与所述第一功能模块不同。

在可选的实施方式中，所述限制时限包括：实时时间戳和限制时长。

在可选的实施方式中，所述限制时限包括：倒计时器。

35 在上述实施方式中，在可能发生崩溃时为最新配置设置限制时间，限制最新配置的生效，以对应应用启动过程中发生崩溃的止血操作，避免应用再次发生崩溃。

在可选的实施方式中，在所述应用启动时，若所述应用监控识别到应用发生崩溃，重启所述应用。

在上述实施方式中，还可以在应用启动时，如果检测到崩溃还可以自动重启应用，可以降低用户体验到崩溃的情况。

40 在可选的实施方式中，所述方法还包括：若所述应用的标志位信息为第二值，获取所述应用的最新配置，其中，所述标志位信息为第二值表示所述应用在上次启动阶段未发生崩溃。

在上述实施方式中，在标志位信息即使表征应用上次没有发生崩溃的情况下也可以对最新配置是否存在限制进行识别，从而可以降低崩溃发生的概率，提高用户体验。

在可选的实施方式中，所述方法还包括：在所述应用启动时，若检测到所述应用发生崩溃，将所述标志位信息更新为所述第一值；在所述应用启动时，若所述应用未发生崩溃且从配置平台获得所述应用的最新配置后，将所述标志位信息更新为第二值。

在可选的实施方式中，在应用启动时，若未获得所述应用的标志位信息，从本地读取所述应用的配置文件，并执行所述配置文件。

在可选的实施方式中，所述方法还包括：从所述应用的配置文件中读取第三功能模块，其中，所述第三功能模块为所述应用中的发生更新的功能模块，所述第三功能模块包括第二功能模块，且不包含第一功能模块；若当前时间在所述第三功能模块的限制时限内，加载所述第三功能模块的历史配置，并执行所述第三功能模块的历史配置，其中，所述第三功能模块的限制时限为获取所述第三功能模块的当前配置时设置的时限。

第二方面，本发明提供一种应用崩溃处理装置，包括获取模块，用于在应用启动时，获取所述应用的标志位信息，其中，所述标志位信息通过预设埋点在所述应用启动时被读取；执行模块，用于若所述应用的标志位信息为第一值，将所述应用由当前配置切换至历史配置，执行所述历史配置，其中，所述标志位信息为第一值表征所述应用在上次启动阶段发生崩溃。

第三方面，本发明提供一种电子设备，包括：处理器、存储器，所述存储器存储有所述处理器可执行的机器可读指令，当电子设备运行时，所述机器可读指令被所述处理器执行时执行如前述实施方式任一所述的方法的步骤。

第四方面，本发明提供一种计算机可读存储介质，该计算机可读存储介质上存储有计算机程序，该计算机程序被处理器运行时执行如前述实施方式任一所述的方法的步骤。

附图说明

为了更清楚地说明本申请实施例的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，应当理解，以下附图仅示出了本申请的某些实施例，因此不应被看作是对范围的限定，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相关的附图。

图 1 为本申请实施例提供的应用崩溃处理方法的运行环境示意图；

图 2 为本申请实施例提供的电子设备的方框示意图；

图 3 为本申请实施例提供的应用崩溃处理方法的流程图；

图 4 为本申请实施例提供的应用崩溃处理装置的功能模块示意图。

具体实施方式

下面将结合本申请实施例中附图，对本申请实施例中的技术方案进行描述。

应注意到：相似的标号和字母在下面的附图中表示类似项，因此，一旦某一项在一个附图中被定义，则在随后的附图中不需要对其进行进一步定义和解释。同时，在本申请的描述中，术语“第一”、“第二”等仅用于区分描述，而不能理解为指示或暗示相对重要性。

目前应用在更新迭代过程中，会增加很多功能，由于新功能的不确定性，且往往无法测试全面，就可能会导致某个 bug 没有被发现，从而将 bug 带到了线上，进而可能产生线上崩溃（Crash）的问题，给用户带来不好的使用体验。考虑到应用的启动阶

段执行的任务是重要的基础库初始化，而该阶段产生的崩溃大概率会导致后续崩溃或功能异常，如果不做处理，会造成应用反复启动崩溃，给用户造成非常不好的影响和体验。

基于此，目前针对应用启动阶段的崩溃的处理方式是在代码中增加 trycatch 保护，其中，trycatch 是一种错误处理机制，用于捕获和处理程序中的异常。在 try 块中，编写可能会引发异常的代码。在 try 块中发生异常，程序将跳转到 catch 块。在 catch 块中，编写有处理异常的代码，例如打印错误消息或执行其他操作。通过该处理机制可以防止程序崩溃，并提供更好的错误处理和用户体验。但是 trycatch 保护仅仅适用于 Java 层代码的防护。

基于此，本申请提供一种应用崩溃处理方法、装置、电子设备和可读存储介质，可以降低应用在启动时发生崩溃的频率。进一步地，在配置平台中存在各种功能的打开和关闭选项，当线上 Crash 监控平台发现有 Crash 时，及时通过配置平台下发相关功能的关闭配置，应用端通过拉取最新的配置后切换成原有逻辑，避免了执行功能的新配置导致的 Crash，从而降低应用下次启动时发生 Crash 的概率。

为便于对本实施例进行理解，首先对执行本申请实施例所公开的一种应用崩溃处理方法的运行环境进行详细介绍。

如图 1 所示，本申请实施例提供的应用崩溃处理方法的运行环境示意图。应用崩溃处理方法的运行环境包括服务器 110 和电子设备 120。该服务器 110 中运行有配置平台，该电子设备 120 中运行有应用。该服务器 110 通过网络与一个或多个电子设备 120 进行通信连接，以进行数据通信或交互。该服务器 110 可以是网络服务器、数据库服务器等。该电子设备 120 可以是个人电脑（personal computer，PC）、平板电脑、智能手机、个人数字助理（personal digital assistant，PDA）等。

如图 2 所示，是电子设备的方框示意图。电子设备 120 可以包括存储器 121、存储控制器 122、处理器 123、外设接口 124、输入输出单元 125、显示单元 126。本领域普通技术人员可以理解，图 2 所示的结构仅为示意，其并不对电子设备 120 的结构造成限定。例如，电子设备 120 还可包括比图 2 中所示更多或者更少的组件，或者具有与图 2 所示不同的配置。

上述的存储器 121、存储控制器 122、处理器 123、外设接口 124、输入输出单元 125 及显示单元 126 各元件相互之间直接或间接地电性连接，以实现数据的传输或交互。例如，这些元件相互之间可通过一条或多条通讯总线或信号线实现电性连接。上述的处理器 123 用于执行存储器中存储的可执行模块。

其中，存储器 121 可以是，但不限于，随机存取存储器（Random Access Memory，简称 RAM），只读存储器（Read Only Memory，简称 ROM），可编程只读存储器（Programmable Read-Only Memory，简称 PROM），可擦除只读存储器（Erasable Programmable Read-Only Memory，简称 EPROM），电可擦除只读存储器（Electric Erasable Programmable Read-Only Memory，简称 EEPROM）等。其中，存储器 121 用于存储程序，所述处理器 123 在接收到执行指令后，执行所述程序，本申请实施例任一实施例揭示的过程定义的电子设备 120 所执行的方法可以应用于处理器 123 中，或者由处理器 123 实现。

上述的处理器 123 可能是一种集成电路芯片，具有信号的处理能力。上述的处理器 123 可以是通用处理器，包括中央处理器（Central Processing Unit，简称 CPU）、网络处理器（Network Processor，简称 NP）等；还可以是数字信号处理器（digital signal processor，简称 DSP）、专用集成电路（Application Specific Integrated Circuit，简称 ASIC）、现场可编程门阵列（Field Programmable Gate Array，简称 FPGA）或者其他可编

程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

上述的外设接口 124 将各种输入/输出装置耦合至处理器 123 以及存储器 121。在
5 一些实施例中，外设接口 124，处理器 123 以及存储控制器 122 可以在单个芯片中实现。在其他一些实例中，他们可以分别由独立的芯片实现。

上述的输入输出单元 125 用于提供给用户输入数据。所述输入输出单元 125 可以是，但不限于，鼠标和键盘等。

上述的显示单元 126 在电子设备 120 与用户之间提供一个交互界面（例如用户操作界面）或用于显示图像数据给用户参考。在本实施例中，所述显示单元可以是液晶显示器或触控显示器。若为触控显示器，其可为支持单点和多点触控操作的电容式触控屏或电阻式触控屏等。支持单点和多点触控操作是指触控显示器能感应到来自该触控显示器上一个或多个位置处同时产生的触控操作，并将该感应到的触控操作交由处理器进行计算和处理。

15 本实施例中的电子设备 120 可以用于执行本申请实施例提供的各个方法中的各个步骤。下面通过几个实施例详细描述应用崩溃处理方法的实现过程。

请参阅图 3，是本申请实施例提供的应用崩溃处理方法的流程图。本实施例中的应用崩溃处理方法可以应用于电子设备中，该电子设备中运行了应用。下面将对图 3 所示的具体流程进行详细阐述。

20 步骤 210，在应用启动时，获取应用的标志位信息。
其中，标志位信息通过预设埋点在该应用启动时被读取。
可选地，该预设埋点的埋点位置可以是在应用的代码的起始位置。例如，该代码的起始位置可以表示执行该应用时的首行代码所在位置。

25 该标志位信息为第一值表示应用在上次启动发生崩溃。
该标志位信息的第一值和第二值的取值可以按需设置，具体可以按照使用习惯设置该第一值和第二值。例如，该第一值可以取值为 flag1，该第二值可以取值为 flag2。再例如，该第一值也可以取值为 1，该第二值也可以取值为 0。

30 在其它情况中，若应用的标志位信息为空，或者该应用不存在标志位信息，可以表示当前启动该应用为新增用户设备，则可以直接读取该当前设备的本地的默认配置文件。

可选地，若该应用的标志位信息未被赋值，可以为该应用的标志位信息设置初始值。该初始值可以是一个不同与第一值和第二值的值。例如，该初始值可以取值为 flag0。

35 可选地，若该应用的标志位信息为空，可以为该标志位信息设置值，该值可以根据此次启动该应用是否发生崩溃，如果未发生崩溃则可以将该标志位信息设置为第二值，如果发生崩溃则可以将该标志位信息设置为第一值。

步骤 220，若应用的标志位信息为第一值，将应用由当前配置切换至历史配置，执行历史配置。

40 若应用的标志位信息为第一值，则可以表示当前设备中上次启动应用时发生了崩溃。为了避免此次因为获取并执行最新配置而引发崩溃，可以将此次启动执行该应用的历史配置。

其中，该历史配置可以表示应用历史上成功启动且成功运行的配置。示例性地，该历史配置可以是引起应用崩溃之前应用运行所使用的配置。

本实施例中，在应用启动时，检测到应用发生了崩溃且已经对标志位信息进行了

识别之后，可以将标志位信息更新为第一值。

考虑到在对标志位信息识别之后，才能够基于标志位信息进行崩溃处理，因此，标志位信息的更新时机确定在对标志位信息进行识别之后。

5 在应用启动时，若所述应用未发生崩溃且从配置平台获得应用的最新配置后，将标志位信息更新为第二值。通过该第二值可以来标记应用的启动没有发生崩溃。

通过上述设置方式，可以在已检查到应用上次发生了崩溃，为了避免重复发生崩溃，则可以先不执行应用的最新配置，就用户而言可以避免应用启动时重复发生崩溃，也就能降低因为应用崩溃的所导致的客户流失率。

10 考虑到导致崩溃的功能模块可能是持续性的，不会因为应用的重启而消失，针对这种情况，本申请提供的应用崩溃处理方法还可以包括步骤 230 和步骤 240。

步骤 230，通过应用监控识别引起应用在上次启动阶段发生崩溃的第一功能模块。示例性地，可以通过应用监控监控该应用的启动情况。在该监控到引起应用发生崩溃的是第一功能模块时，则执行该步骤 240。

15 示例性地，该第一功能模块可以是该应用中的一个功能，例如，该第一功能模块可以是登录功能模块、数据获取功能模块等。

步骤 240，向配置平台发送第一功能模块的处理请求。

20 该处理请求用于指示所述配置平台停止下发所述第一功能模块对应的配置项。例如，配置平台在接收到第一功能模块的处理请求后可以关闭该第一功能模块，从而在电子设备下次发送该应用的配置获取请求后，不再下发第一功能模块的对应更新的配置项。

可选地，可以在监控到崩溃时，在配置平台点选关闭。

通过上述方法，可以及时关闭引起崩溃的功能模块，配置平台不再下发该功能模块的更新配置，降低因为一个功能模块的异常而导致的频繁崩溃的情况的发生。

25 应用在启动时如果发生了崩溃，可能是部分功能模块的更新配置所存在的异常所导致的，也可能是功能模块的更新配置的生效滞后所导致的崩溃。基于此，本申请实施例提供的应用崩溃处理还可以包括：针对在上次启动阶段发生崩溃的所述应用所获得的配置中对应的第二功能模块，在等待限制时限后执行所述第二功能模块的最新配置。

30 其中，所述第二功能模块为上次启动阶段获得的配置中对应的更新功能模块，且所述第二功能模块与所述第一功能模块不同。

示例性地，可以为在上次启动发生崩溃的应用，在上次启动时获得的最新配置对应的第二功能模块设置限制时限。

35 该限制时限可以用来限定更新配置的生效时间，如果功能模块的更新配置的时间还在该限制时限内可以表示该功能模块的更新配置还没有生效，则可以等待该功能模块的更新配置生效后再使用该功能模块的更新配置。

在一种实施方式中，该限制时限可以通过实时时间戳和限制时长来实现。上述的为发生崩溃的应用的当前最新配置对应的功能模块设置限制时限可以包括：为应用的第二功能模块设置实时时间戳和限制时长。

40 针对第二功能模块，在该第二功能模块产生当前最新配置或者获得该当前最新配置时，为该第二功能模块设置实时时间戳以及限制时长。

该实时时间戳可以是产生当前最新配置或者获得该当前最新配置的实时时间。该限制时长可以按需设置，例如，该限制时长是 5 分钟、6 分钟、7 分钟等时长。

在需要使用第二功能模块时，从配置文件中读取该应用的功能模块时，先识别该功能模块是否具有实时时间戳，在识别当该第二功能模块的实时时间戳后，再判断当

前时间与实时时间戳的时差是否大于限制时长，如果当前时间与实时时间戳的时差大于限制时长，则可以启用该功能模块的更新配置，如果当前时间与实时时间戳的时差不大于限制时长，则可以先不启用该第二功能模块的更新配置，仍然使用该第二功能模块的历史配置，以降低由该第二功能模块所导致的启动阶段的崩溃的概率。

5 在一种实施方式中，该限制时限可以通过倒计时器来实现。上述的为发生崩溃的应用的当前最新配置对应的功能模块设置限制时限可以包括为应用的当前最新配置对应的更新功能模块设置倒计时器。

 针对第二功能模块，在获得该第二功能模块的最新配置时，为该目标功能模块设置倒计时器。

10 在应用启动时，从配置文件中读取该应用的功能模块时，先识别该功能模块是否具有倒计时器，在识别该功能模块的倒计时器后，再判断倒计时器是否计时结束，如果倒计时器计时结束，则可以启用该功能模块的更新配置，如果倒计时器计时未结束，则可以先不启用该功能模块的更新配置，仍然使用该目标功能模块的历史配置，以降低由该功能模块所导致的启动阶段的崩溃。

15 通过设置该限制时限可以延缓应用中的功能模块的更新配置的生效时间，降低应用因为更新配置的延缓生效所带来的启动阶段的崩溃。

 应用如果在上次启动时发生了崩溃，可以通过上述处理流程降低再次启动发生崩溃的概率，降低应用崩溃的发生。可以理解的是，如果上次启动未发生崩溃，则可以通过按照常规流程去获得配置文件。基于此，本申请实施例提供的方法还包括：步骤
20 250，若应用的标志位信息为第二值，获取应用的最新配置。

 其中，应用的标志位信息为第二值表示在上次应用启动时未发生崩溃，该在上次应用启动时未发生崩溃则可以表示该应用在上次启动时成功从配置平台获得了最新配置。

25 可选地，在每次启用应用的任意功能模块时，可以先确定该功能模块是否存在限制时限。基于此，本申请实施例提供的应用崩溃处理方法还可以包括步骤 260 至步骤 270。

 步骤 260，从应用的配置文件中读取第三功能模块。

 其中，第三功能模块为应用中的发生更新的功能模块。

30 考虑前述的第一功能模块可以暂时不被执行，该第三功能模块可以包括上述的第二功能模块，且不包含上述的第一功能模块。

 若第三功能模块被设置了限制时限，且当前时间在限制时限内，则可以表示该第三功能模块的最新配置还未生效，则可以执行步骤 270。

 步骤 270，若当前时间在第三功能模块的限制时限内，加载第三功能模块的历史配置，并执行第三功能模块的历史配置。

35 本实施例中，该第三功能模块的限制时限为获取所述第三功能模块的当前配置时设置的时限。

 可选地，该第三功能模块的历史配置可以为应用上次正常启动所使用的该第三功能模块的配置。

40 在一些情况下，应用在启动阶段可能会发生一些未知的崩溃，上述的处理方式也不能够改善崩溃的发生时，本申请实施例提供的应用崩溃处理方法还可以包括：在应用启动时，应用监控识别到应用发生崩溃，可以重启应用。

 可选地，应用监控检测到程序崩溃时，可以按需设置重启次数，在该重启次数限定内，检测到程序崩溃时，可以重新启应用。若在检测到程序崩溃，且连续启动了该重启次数次应用，还是检测到崩溃，则可以输出提示，以提醒用户该应用存在故障。

可选地，还可以将该应用存在的故障内容输出，以提示用户。例如，故障内容可以是应用版本不对、数据源不对等。

通过自动对应用重启，就用户而言可以无需感知到应用崩溃的情况下就能够实现应用的启动，仅仅能够感知的是应用启动时间相对较长而已，也可以不用通过清空应用缓存或者卸载应用重装等复杂方式处理，降低用户的流失。

在本申请实施例提供的方法中，通过对标志位信息的识别以及基于标志位信息额值适应性调整应用的启动策略，可以缓解应用反复启动崩溃导致只能清空应用缓存或者卸载重装，避免造成用户流失的情况。在出现崩溃之后可以执行自动重启的操作，让用户视角其实是观察不到闪退的，只是会感觉到启动的耗时约是平时相对更少，提高用户体验。相较于 trycatch 无法保护到的 Native 层，本申请的实施例提供的方案增加了显著有效的保护机制，使应用的启动的崩溃更少。且本申请实施例提供的方法的逻辑都可以在应用中完成，不需要对后端进行二次改造，降低实施的难度。且本申请实施例提供的方法中对应用启动的耗时和性能相对影响更小。

基于同一申请构思，本申请实施例中还提供了与应用崩溃处理方法对应的应用崩溃处理装置，由于本申请实施例中的装置解决问题的原理与前述的应用崩溃处理方法实施例相似，因此本实施例中的装置的实施可以参见上述方法的实施例中的描述，重复之处不再赘述。

请参阅图 4，是本申请实施例提供的应用崩溃处理装置的功能模块示意图。本实施例中的应用崩溃处理装置中的各个模块用于执行上述方法实施例中的各个步骤。应用崩溃处理装置包括：获取模块 310 和执行模块 320；其中各个模块的内容如下所示：获取模块 310，用于在应用启动时，获取应用的标志位信息，其中，标志位信息通过预设埋点在应用启动时被读取；执行模块 320，用于若应用的标志位信息为第一值，将应用由当前配置切换至历史配置，执行历史配置，其中，标志位信息为第一值表征应用在上次启动阶段发生崩溃。

一种可能的实施方式中，本实施例提供的应用崩溃处理装置还可以包括：确定模块，用于确通过应用监控识别引起应用在上次启动阶段发生崩溃的第一功能模块；请求模块，用于向配置平台发送第一功能模块的处理请求，处理请求用于指示配置平台停止下发第一功能模块对应的配置项。

一种可能的实施方式中，本实施例提供的应用崩溃处理装置还可以包括：执行模块，用于针对在上次启动阶段发生崩溃的应用所获得的配置中对应的第二功能模块，在等待限制时限后执行第二功能模块的最新配置，其中，第二功能模块为上次启动阶段获得的配置中对应的更新功能模块，且第二功能模块与第一功能模块不同。

一种可能的实施方式中，本实施例提供的应用崩溃处理装置还可以包括：重启模块，用于在应用启动时，若应用监控识别到应用发生崩溃，重启应用。

一种可能的实施方式中，本实施例提供的应用崩溃处理装置还可以包括：若应用的标志位信息为第二值，获取应用的最新配置，其中，标志位信息为第二值表示应用在上次启动阶段未发生崩溃。

一种可能的实施方式中，本实施例提供的应用崩溃处理装置还可以包括：更新模块，用于在所述应用启动时，若检测到所述应用发生崩溃，将所述标志位信息更新为所述第一值；在所述应用启动时，若所述应用未发生崩溃且从配置平台获得所述应用的最新配置后，将所述标志位信息更新为第二值。

一种可能的实施方式中，本实施例提供的应用崩溃处理装置还可以包括：执行模块，还用于在应用启动时，若未获得应用的标志位信息，从本地读取应用的配置文

件，并执行配置文件。

一种可能的实施方式中，本实施例提供的应用崩溃处理装置还可以包括：读取模块，用于从应用的配置文件中读取第三功能模块，其中，第三功能模块为应用中的发生更新的功能模块，所述第三功能模块包括第二功能模块，且不包含第一功能模块；

5 执行模块，用于若当前时间在第三功能模块的限制时限内，加载第三功能模块的历史配置，并执行第三功能模块的历史配置，其中，第三功能模块的限制时限为获取第三功能模块的当前配置时设置的时限。

此外，本申请实施例还提供一种计算机可读存储介质，该计算机可读存储介质上存储有计算机程序，该计算机程序被处理器运行时执行上述方法实施例中所述的应用崩溃处理方法的步骤。

10 本申请实施例所提供的应用崩溃处理方法的计算机程序产品，包括存储了程序代码的计算机可读存储介质，所述程序代码包括的指令可用于执行上述方法实施例中所述的应用崩溃处理方法的步骤，具体可参见上述方法实施例，在此不再赘述。

在本申请所提供的几个实施例中，应该理解到，所揭露的装置和方法，也可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的，例如，附图中的流程图和框图显示了根据本申请的多个实施例的装置、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现方式中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以用执行规定的功能或动作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

20 另外，在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部分，也可以是各个模块单独存在，也可以两个或两个以上模块集成形成一个独立的部分。

所述功能如果以软件功能模块的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读存储介质中。基于这样的理解，本申请的技术方案本质上或者

30 或者说对现有技术做出贡献的部分或者该技术方案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中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（ROM，Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、磁碟或者光盘等各种可以存储程序代码的介质。需要说明的是，在本文中，诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

40 以上所述仅为本申请的优选实施例而已，并不用于限制本申请，对于本领域的技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原则之内，所作的

任何修改、等同替换、改进等，均应包含在本申请的保护范围之内。应注意到：相似的标号和字母在下面的附图中表示类似项，因此，一旦某一项在一个附图中被定义，则在随后的附图中不需要对其进行进一步定义和解释。

- 5 以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以权利要求的保护范围为准。

10

15

20

25

30

35

权 利 要 求 书

- 1.一种应用崩溃处理方法，其特征在于，包括：
- 5 在应用启动时，获取所述应用的标志位信息，其中，所述标志位信息通过预设埋点在所述应用启动时被读取；

 若所述应用的标志位信息为第一值，将所述应用由当前配置切换至历史配置，执行所述历史配置，其中，所述标志位信息为第一值表征所述应用在上次启动阶段发生崩溃。
- 10 2. 根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

 通过应用监控识别引起所述应用在上次启动阶段发生崩溃的第一功能模块；

 向配置平台发送所述第一功能模块的处理请求，所述处理请求用于指示所述配置平台停止下发所述第一功能模块对应的配置项。
- 15 3. 根据权利要求 2 所述的方法，其特征在于，所述方法还包括：

 针对在上次启动阶段发生崩溃的所述应用所获得的配置中对应的第二功能模块，在等待限制时限后执行所述第二功能模块的最新配置，其中，所述第二功能模块为上次启动阶段获得的配置中对应的更新功能模块，且所述第二功能模块与所述第一功能模块不同。
- 20 4. 根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

 在所述应用启动时，若应用监控识别到应用发生崩溃，重启所述应用。
5. 根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

 若所述应用的标志位信息为第二值，获取所述应用的最新配置，其中，所述标志位信息为第二值表示所述应用在上次启动阶段未发生崩溃。
- 25 6. 根据权利要求 1-5 任意一项所述的方法，其特征在于，所述方法还包括：

 在所述应用启动时，若检测到所述应用发生崩溃，将所述标志位信息更新为所述第一值；

在所述应用启动时，若所述应用未发生崩溃且从配置平台获得所述应用的最新配置后，将所述标志位信息更新为第二值。

7. 根据权利要求 1-5 任意一项所述的方法，其特征在于，所述方法还包括：

在应用启动时，若未获得所述应用的标志位信息，从本地读取所述应用的配置文件，并执行所述配置文件。

8. 根据权利要求 1-5 任意一项所述的方法，其特征在于，所述方法还包括：

从所述应用的配置文件中读取第三功能模块，其中，所述第三功能模块为所述应用中的发生更新的功能模块，所述第三功能模块包括第二功能模块，且不包含第一功能模块；

若当前时间在所述第三功能模块的限制时限内，加载所述第三功能模块的历史配置，并执行所述第三功能模块的历史配置，其中，所述第三功能模块的限制时限为获取所述第三功能模块的当前配置时设置的时限。

9. 一种应用崩溃处理装置，其特征在于，包括：

获取模块，用于在应用启动时，获取所述应用的标志位信息，其中，所述标志位信息通过预设埋点在所述应用启动时被读取；

执行模块，用于若所述应用的标志位信息为第一值，将所述应用由当前配置切换至历史配置，执行所述历史配置，其中，所述标志位信息为第一值表征所述应用在上次启动阶段发生崩溃。

10. 一种电子设备，其特征在于，包括：处理器、存储器，所述存储器存储有所述处理器可执行的机器可读指令，当电子设备运行时，所述机器可读指令被所述处理器执行时执行如权利要求 1 至 8 任意一项所述的方法的步骤。

11. 一种计算机可读存储介质，其特征在于，该计算机可读存储介质上存储有计算机程序，该计算机程序被处理器运行时执行如权利要求 1 至 8 任意一项所述的方法的步骤。