(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 107229552 B
(45)授权公告日 2021.01.26

(21)申请号 201710354853.9

(22)申请日 2017.05.18

(65)同一申请的已公布的文献号
申请公布号 CN 107229552 A

(43)申请公布日 2017.10.03

(73)专利权人 北京京东尚科信息技术有限公司
地址 100195 北京市海淀区杏石口路65号西杉创意园西区11C楼东段1-4层西段1-4层
专利权人 北京京东世纪贸易有限公司

(72)发明人 朱孝瑾 程剑锋 王孝满 吴迪

(74)专利代理机构 中科专利商标代理有限责任公司 11021
代理人 杨姗

(51)Int.Cl.
G06F 11/32(2006.01)

审查员 杨硕

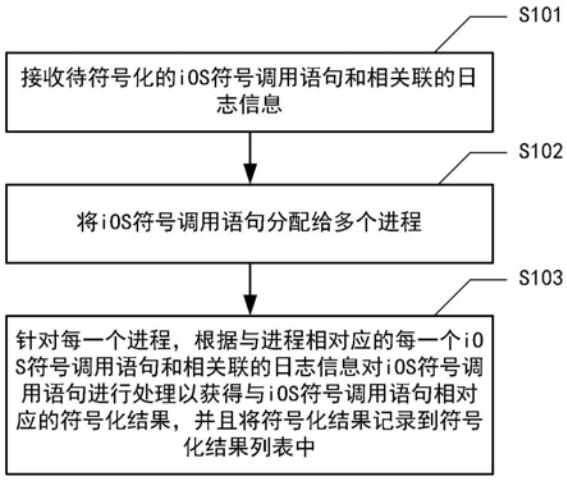
权利要求书3页 说明书9页 附图4页

(54)发明名称

用于对iOS符号调用语句进行符号化的方法和装置

(57)摘要

本公开的实施例公开了一种用于对iOS符号调用语句进行符号化的方法，包括：接收待符号化的iOS符号调用语句和相关联的日志信息；将所述iOS符号调用语句分配给多个进程；以及针对每一个进程，根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息对所述iOS符号调用语句进行处理以获得与所述iOS符号调用语句相对应的符号化结果，并且将所述符号化结果记录到符号化结果列表中。



CN 107229552 B

1.一种用于对iOS符号调用语句进行符号化的方法,包括:

接收待符号化的iOS符号调用语句和相关联的日志信息,所述iOS符号调用语句是iOS系统上的应用发生崩溃时的调用栈信息;

将所述iOS符号调用语句分配给多个进程;以及

针对每一个进程,根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息在缓存中查找与所述iOS符号调用语句相对应的符号化结果,并且将所述符号化结果记录到符号化结果列表中,其中,所述符号化结果是符号化的可视化语句。

2.根据权利要求1所述的方法,其中,根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息对所述iOS符号调用语句进行处理以获得与所述iOS符号调用语句相对应的符号化结果包括:

根据所述iOS符号调用语句和相关联的日志信息获取与所述iOS符号调用语句相对应的调用库名、系统版本、第一设备体系架构、调用语句偏移地址和调用语句基地址;

根据所述调用库名、所述第一设备体系架构和所述调用语句偏移地址在所述缓存中查找与所述iOS符号调用语句相对应的符号化结果;

如果找到相对应的符号化结果,则将所找到的符号化结果确定为与所述iOS符号调用语句相对应的符号化结果。

3.根据权利要求2所述的方法,其中,根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息对所述iOS符号调用语句进行处理以获得与所述iOS符号调用语句相对应的符号化结果还包括:

如果未找到相对应的符号化结果,则

根据所述调用库名、所述系统版本和所述第一设备体系架构获得符号映射表地址;

根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构;

根据所述符号映射表地址、所述第二设备体系架构、所述调用语句基地址和所述调用语句偏移地址利用符号化语句对所述iOS符号调用语句进行符号化以获得符号化结果;以及

将所获得的符号化结果与所述调用库名、所述第一设备体系架构和所述调用语句偏移地址相关联地存储在所述缓存中。

4.根据权利要求3所述的方法,其中,所述符号化语句是atos语句或symbolicate语句。

5.根据权利要求3所述的方法,其中,根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构包括:

对所述符号映射表地址执行lipo-info命令,以获得所述符号映射表地址支持的设备体系架构列表;

确定所述第一设备体系架构是否存在于所述设备体系架构列表中;以及

如果所述第一设备体系架构不存在于所述设备体系架构列表中,则将所述设备体系架构列表中比所述第一设备体系架构的版本更低的设备体系架构确定为所述第二设备体系架构。

6.根据权利要求5所述的方法,其中,根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构还包括:

如果所述第一设备体系架构存在于所述设备体系架构列表中,则将所述第一设备体系

架构确定为所述第二设备体系架构。

7.根据权利要求3所述的方法，其中，根据所述调用库名、所述系统版本和所述第一设备体系架构获得符号映射表地址包括：

根据所述调用库名、所述系统版本和所述第一设备体系架构利用预先建立的符号映射表字典查找符号映射表地址；

当未查找到符号映射表地址时，将所述系统版本改变为另一系统版本；以及

根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址。

8.根据权利要求7所述的方法，其中根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址是利用模糊匹配方法执行的。

9.一种用于对iOS符号调用语句进行符号化的装置，包括：

接收模块，被配置为接收待符号化的iOS符号调用语句和相关联的日志信息，所述iOS符号调用语句是iOS系统上的应用发生崩溃时的调用栈信息；

分配模块，被配置为将所述iOS符号调用语句分配给多个进程；以及

处理模块，被配置为针对每一个进程，根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息在缓存中查找与所述iOS符号调用语句相对应的符号化结果，并且将所述符号化结果记录到符号化结果列表中，其中，所述符号化结果是符号化的可视化语句。

10.根据权利要求9所述的装置，其中，所述处理模块被进一步配置为：

根据所述iOS符号调用语句和相关联的日志信息获取与所述iOS符号调用语句相对应的调用库名、系统版本、第一设备体系架构、调用语句偏移地址和调用语句基地址；

根据所述调用库名、所述第一设备体系架构和所述调用语句偏移地址在所述缓存中查找与所述iOS符号调用语句相对应的符号化结果；

如果找到相对应的符号化结果，则将所找到的符号化结果确定为与所述iOS符号调用语句相对应的符号化结果。

11.根据权利要求10所述的装置，其中，所述处理模块被进一步配置为：

如果未找到相对应的符号化结果，则

根据所述调用库名、所述系统版本和所述第一设备体系架构获得符号映射表地址；

根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构；

根据所述符号映射表地址、所述第二设备体系架构、所述调用语句基地址和所述调用语句偏移地址利用符号化语句对所述iOS符号调用语句进行符号化以获得符号化结果；以及

将所获得的符号化结果与所述调用库名、所述第一设备体系架构和所述调用语句偏移地址相关联地存储在所述缓存中。

12.根据权利要求11所述的装置，其中，所述符号化语句是atos语句或symbolicate语句。

13.根据权利要求11所述的装置，其中，所述处理模块被进一步配置为：

对所述符号映射表地址执行lipo-info命令，以获得所述符号映射表支持的设备体系架构列表；

确定所述第一设备体系架构是否存在于所述设备体系架构列表中;以及

如果所述第一设备体系架构不存在于所述设备体系架构列表中,则将所述设备体系架构列表中比所述第一设备体系架构的版本更低的设备体系架构确定为所述第二设备体系架构。

14.根据权利要求13所述的装置,其中,所述处理模块被进一步配置为:

如果所述第一设备体系架构存在于所述设备体系架构列表中,则将所述第一设备体系架构确定为所述第二设备体系架构。

15.根据权利要求11所述的装置,其中,所述处理模块被进一步配置为:

根据所述调用库名、所述系统版本和所述第一设备体系架构利用预先建立的符号映射表字典查找符号映射表地址;

当未查找到符号映射表地址时,将所述系统版本改变为另一系统版本;以及

根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址。

16.根据权利要求15所述的装置,其中根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址是利用模糊匹配方法执行的。

17.一种计算机可读存储介质,其上存储有可执行指令,该指令被处理器执行时使处理器执行根据权利要求1至8中任一项所述的方法。

18.一种用于对iOS符号调用语句进行符号化的服务器,包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序,

其中,当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现根据权利要求1至8中任一项所述的方法。

用于对iOS符号调用语句进行符号化的方法和装置

技术领域

[0001] 本公开涉及计算机技术领域，更具体地，涉及一种用于对iOS符号调用语句进行符号化的方法和装置。

背景技术

[0002] 目前运行在iOS系统上的app通常会在崩溃发生(即，应用闪退)之后将崩溃发生时的调用栈信息上报到服务器。上报的调用栈信息是一种符号地址信息，例如appName 0x00009404 0x4000+21508的形式，但是用户希望看到的是例如-[SeckillListViewController collectionView:cellForItemAtIndexPath:](in appName)(SeckillListViewController.m:470)的形式的符号化的可视化语句，通过符号化的可视化语句，用户可以明确地知道具体是哪个函数的哪一行出了问题，且如果整个符号地址调用栈都被符号化为这样的可视化语句，那么调用栈看起来会非常的明确并且代码调用逻辑清晰，从而对定位崩溃原因和修复都有很大的帮助。

[0003] 目前，苹果公司提供了一种符号化工具atos，它是一个命令行工具，使用如下命令:atos-arch<体系架构>-o<符号表文件路径>-l<0x程序基地址><0x语句地址>。该工具可以帮助用户将符号地址符号化为可视化的代码调用语句。但是atos有一个非常致命的缺点，即其执行速度极慢，当用户分析崩溃的调用栈时，特别是针对大公司的崩溃的调用栈，往往一天的数据量就达到几十万行调用栈语句，而每一次语句解析都需要调用atos，这需要花费非常多的时间并且效率极低。

[0004] 除了atos之外，目前还存在以下符号化工具:Python、lipo、symbolicate等。Python是一种面向对象的解释型计算机程序设计语言，常用于快速生成程序的原型。lipo可以查询符号表支持的体系架构。

[0005] 然而，目前市面上的工具都相对简单，例如，atos或者symbolicate只能根据符号地址调用栈对属于自己app的调用语句进行符号化。因此，在实现本发明过程中，发明人发现这些解决方案至少存在以下技术缺点:首先，这些工具没有对调用栈中的系统调用进行符号化，只符号化了自己app部分的调用。如果能够对系统调用也进行符号化，那么就可以提供一个完整的调用栈，可以更有效的帮助开发者知道崩溃发生时的上下文环境与函数调用顺序，从而更方便定位问题。其次，这些工具没有针对大量符号地址调用栈的批量符号化处理工具，即不存在可以一次性输入大量符号地址调用栈，然后可以在全部解析完成之后一次性输出所有符号化结果的工具。例如，atos只支持一次解析一个符号表的崩溃地址，symbolicate也只能一次解析一个符号调用栈文件。最后，这些工具的效率很低，atos和symbolicate没有针对符号化结果做缓存，在海量数据符号化时无法提升速度。

发明内容

[0006] 有鉴于此，本公开提供了一种用于对iOS符号调用语句进行符号化的方法和装置。

[0007] 本公开的一个方面提供了一种用于对iOS符号调用语句进行符号化的方法，包括：

接收待符号化的iOS符号调用语句和相关联的日志信息;将所述iOS符号调用语句分配给多个进程;以及针对每一个进程,根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息对所述iOS符号调用语句进行处理以获得与所述iOS符号调用语句相对应的符号化结果,并且将所述符号化结果记录到符号化结果列表中。

[0008] 根据本公开的实施例,根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息对所述iOS符号调用语句进行处理以获得与所述iOS符号调用语句相对应的符号化结果包括:根据所述iOS符号调用语句和相关联的日志信息获取与所述iOS符号调用语句相对应的调用库名、系统版本、第一设备体系架构、调用语句偏移地址和调用语句基地址;根据所述调用库名、所述第一设备体系架构和所述调用语句偏移地址在缓存中查找与所述iOS符号调用语句相对应的符号化结果;如果找到相对应的符号化结果,则将所找到的符号化结果确定为与所述iOS符号调用语句相对应的符号化结果。

[0009] 根据本公开的实施例,根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息对所述iOS符号调用语句进行处理以获得与所述iOS符号调用语句相对应的符号化结果还包括:如果未找到相对应的符号化结果,则根据所述调用库名、所述系统版本和所述第一设备体系架构获得符号映射表地址;根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构;根据所述符号映射表地址、所述第二设备体系架构、所述调用语句基地址和所述调用语句偏移地址利用符号化语句对所述iOS符号调用语句进行符号化以获得符号化结果;以及将所获得的符号化结果与所述调用库名、所述第一设备体系架构和所述调用语句偏移地址相关联地存储在所述缓存中。

[0010] 根据本公开的实施例,所述符号化语句是atos语句或symbolicate语句。

[0011] 根据本公开的实施例,根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构包括:对所述符号映射表地址执行lipo-info命令,以获得所述符号映射表支持的设备体系架构列表;确定所述第一设备体系架构是否存在于所述设备体系架构列表中;以及如果所述第一设备体系架构不存在于所述设备体系架构列表中,则将所述设备体系架构列表中比所述第一设备体系架构的版本更低的设备体系架构确定为所述第二设备体系架构。

[0012] 根据本公开的实施例,根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构还包括:如果所述第一设备体系架构存在于所述设备体系架构列表中,则将所述第一设备体系架构确定为所述第二设备体系架构。

[0013] 根据本公开的实施例,根据所述调用库名、所述系统版本和所述第一设备体系架构获得符号映射表地址包括:根据所述调用库名、所述系统版本和所述第一设备体系架构利用预先建立的符号映射表字典查找符号映射表地址;当未查找到符号映射表地址时,将所述系统版本改变为另一系统版本;以及根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址。

[0014] 根据本公开的实施例,根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址是利用模糊匹配方法执行的。

[0015] 本公开的另一个方面提供了一种用于对iOS符号调用语句进行符号化的装置,包括:接收模块,被配置为接收待符号化的iOS符号调用语句和相关联的日志信息;分配模块,被配置为将所述iOS符号调用语句分配给多个进程;以及处理模块,被配置为针对每一个进

程,根据与所述进程相对应的每一个iOS符号调用语句和相关联的日志信息对所述iOS符号调用语句进行处理以获得与所述iOS符号调用语句相对应的符号化结果,并且将所述符号化结果记录到符号化结果列表中。

[0016] 根据本公开的实施例,所述处理模块被进一步配置为:根据所述iOS符号调用语句和相关联的日志信息获取与所述iOS符号调用语句相对应的调用库名、系统版本、第一设备体系架构、调用语句偏移地址和调用语句基地址;根据所述调用库名、所述第一设备体系架构和所述调用语句偏移地址在缓存中查找与所述iOS符号调用语句相对应的符号化结果;如果找到相对应的符号化结果,则将所找到的符号化结果确定为与所述iOS符号调用语句相对应的符号化结果。

[0017] 根据本公开的实施例,所述处理模块被进一步配置为:如果未找到相对应的符号化结果,则根据所述调用库名、所述系统版本和所述第一设备体系架构获得符号映射表地址;根据所述符号映射表地址和所述第一设备体系架构获得第二设备体系架构;根据所述符号映射表地址、所述第二设备体系架构、所述调用语句基地址和所述调用语句偏移地址利用符号化语句对所述iOS符号调用语句进行符号化以获得符号化结果;以及将所获得的符号化结果与所述调用库名、所述第一设备体系架构和所述调用语句偏移地址相关联地存储在所述缓存中。

[0018] 根据本公开的实施例,所述符号化语句是atos语句或symbolicate语句。

[0019] 根据本公开的实施例,所述处理模块被进一步配置为:对所述符号映射表地址执行lipo-info命令,以获得所述符号映射表支持的设备体系架构列表;确定所述第一设备体系架构是否存在于所述设备体系架构列表中;以及如果所述第一设备体系架构不存在于所述设备体系架构列表中,则将所述设备体系架构列表中比所述第一设备体系架构的版本更低的设备体系架构确定为所述第二设备体系架构。

[0020] 根据本公开的实施例,所述处理模块被进一步配置为:如果所述第一设备体系架构存在于所述设备体系架构列表中,则将所述第一设备体系架构确定为所述第二设备体系架构。

[0021] 根据本公开的实施例,所述处理模块被进一步配置为:根据所述调用库名、所述系统版本和所述第一设备体系架构利用预先建立的符号映射表字典查找符号映射表地址;当未查找到符号映射表地址时,将所述系统版本改变为另一系统版本;以及根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址。

[0022] 根据本公开的实施例,根据所述调用库名、改变后的另一系统版本和所述第一设备体系架构获得符号映射表地址是利用模糊匹配方法执行的。

[0023] 本公开的另一方面提供了一种计算机可读存储介质,其上存储有可执行指令,该指令被处理器执行时使处理器执行如上所述的方法。

[0024] 本公开的另一方面提供了一种用于对iOS符号调用语句进行符号化的服务器,包括:一个或多个处理器;存储装置,用于存储一个或多个程序,其中,当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如上所述的方法。

[0025] 根据本公开的实施例,通过精细化的符号表文件管理、模糊匹配算法、符号化缓存和多进程并发,可以至少部分地解决无法对系统调用进行符号化以及无法快速地处理大量符号调用语句的问题,并因此可以实现支持对系统符号地址的符号化从而实现整个调用栈

的完全符号化以帮助开发者更好的定位问题、批量处理符号地址调用栈(非常适合符号化大批量的崩溃数据)、对得到的符号化结果列表进行缓存以帮助下一次解析时快速符号化以及极大的提高了符号化的效率(效率提升在50倍以上,并且效率提升会随着运行次数的增多而越发的明显)的技术效果。

附图说明

[0026] 通过以下参照附图对本公开实施例的描述,本公开的上述以及其他目的、特征和优点将更为清楚,在附图中:

[0027] 图1示意性示出了根据本公开实施例的用于对iOS符号调用语句进行符号化的方法100的流程图;

[0028] 图2示意性地示出了根据本公开实施例的用于对iOS符号调用语句进行符号化的方法100中的操作S103的流程图;

[0029] 图3示意性示出了根据本公开实施例的用于对iOS符号调用语句进行符号化的装置300的框图;

[0030] 图4示出了可以应用本公开实施例的用于对iOS符号调用语句进行符号化的方法和装置的示例性系统架构400;以及

[0031] 图5示出了适于用来实现本公开实施例的终端设备的计算机系统500的结构示意图。

具体实施方式

[0032] 以下,将参照附图来描述本公开的实施例。但是应该理解,这些描述只是示例性的,而并非要限制本公开的范围。此外,在以下说明中,省略了对公知结构和技术的描述,以避免不必要地混淆本公开的概念。

[0033] 在此使用的术语仅仅是为了描述具体实施例,而并非意在限制本公开。这里使用的词语“一”、“一个(种)”和“该”等也应包括“多个”、“多种”的意思,除非上下文另外明确指出。此外,在此使用的术语“包括”、“包含”等表明了所述特征、步骤、操作和/或部件的存在,但是并不排除存在或添加一个或多个其他特征、步骤、操作或部件。

[0034] 在此使用的所有术语(包括技术和科学术语)具有本领域技术人员通常所理解的含义,除非另外定义。应注意,这里使用的术语应解释为具有与本说明书的上下文相一致的含义,而不应以理想化或过于刻板的方式来解释。

[0035] 图1示意性示出了根据本公开的实施例的用于对iOS符号调用语句进行符号化的方法100的流程图。

[0036] 如图1所示,该方法包括在操作S101,接收待符号化的iOS符号调用语句和相关联的日志信息。待符号化的iOS符号调用语句可以是以文件的形式接收的,例如,接收多个iOS符号调用语句文件,每一个文件中包括多个iOS符号调用语句。iOS符号调用语句可以标识调用库名、调用语句基地址和调用语句偏移地址,并且日志信息可以包括与iOS符号调用语句相关联的系统版本和第一设备体系架构。

[0037] 然后,在操作S102,将iOS符号调用语句分配给进程池中的多个进程。该进程池中的并发进程的数量可以与CPU内核的数量相同。当以文件的形式接收到iOS符号调用语句

时，可以针对每一个进程分配一个iOS符号调用语句文件。

[0038] 在操作S103，针对每一个进程，根据与该进程相对应的每一个iOS符号调用语句和相关联的日志信息对iOS符号调用语句进行处理以获得与iOS符号调用语句相对应的符号化结果，并且将符号化结果记录到符号化结果列表中。在得到了针对所有iOS符号调用语句的符号化结果之后，可以将符号化结果列表输出到硬盘中以便持久化。此外，在得到了针对所有iOS符号调用语句的符号化结果之后，还可以输出符号映射表文件未匹配信息以及符号映射表文件模糊匹配信息以及统计结果，其中，符号映射表文件未匹配信息指示未能找到匹配的或模糊匹配的符号映射表文件的iOS符号调用语句或文件的信息和数量，符号映射表文件模糊匹配信息指示找到模糊匹配的符号映射表文件的iOS符号调用语句或文件的信息和数量，以及统计结果指示经符号化的iOS符号调用语句或文件的数量、未经符号化的iOS符号调用语句或文件的数量等。

[0039] 在执行操作S103之前，可以将硬盘中已持久化的符号化结果列表读出到缓存中。此外，在执行操作S103之前，还可以对文件的合法性进行校验，仅当文件是合法的时，才执行操作S103。下面将参照图2具体描述操作S103。

[0040] 操作S103包括在操作S201，根据iOS符号调用语句和相关联的日志信息获取与iOS符号调用语句相对应的调用库名、系统版本、第一设备体系架构、调用语句偏移地址和调用语句基地址。

[0041] 然后，在操作S202，根据调用库名、第一设备体系架构和调用语句偏移地址在缓存中查找与iOS符号调用语句相对应的符号化结果。

[0042] 在操作S203，确定是否找到相对应的符号化结果。

[0043] 如果找到相应的符号化结果，则在操作S204，将所找到的符号化结果确定为与iOS符号调用语句相对应的符号化结果。

[0044] 如果未找到相对应的符号化结果，则在操作S205，根据调用库名、系统版本和第一设备体系架构获得符号映射表地址。

[0045] 在操作S206，根据符号映射表地址和第一设备体系架构获得第二设备体系架构。

[0046] 在操作S207，根据符号映射表地址、第二设备体系架构、调用语句基地址和调用语句偏移地址利用符号化语句对iOS符号调用语句进行符号化以获得符号化结果。符号化语句可以是atos语句或symbolicate语句。

[0047] 在操作S208，将所获得的符号化结果与调用库名、第一设备体系架构和调用语句偏移地址相关联地存储在缓存中。

[0048] 操作S206可以包括：对符号映射表地址执行lipo-info命令，以获得符号映射表支持的设备体系架构列表；确定第一设备体系架构是否存在于设备体系架构列表中；以及如果第一设备体系架构不存在于设备体系架构列表中，则将设备体系架构列表中比第一设备体系架构的版本更低的设备体系架构确定为第二设备体系架构。例如，可以将设备体系架构列表中比第一设备体系架构的版本更低的设备体系架构中的最高版本设备体系架构确定为第二设备体系架构。备选地，如果第一设备体系架构存在于设备体系架构列表中，则将第一设备体系架构确定为第二设备体系架构。

[0049] 操作S205可以包括：根据调用库名、系统版本和第一设备体系架构利用预先建立的符号映射表字典查找符号映射表地址；当查找到符号映射表地址时，将查找到的符号映

射表地址作为要获得的符号映射表地址;当未查找到符号映射表地址时,将系统版本改变为另一系统版本;以及根据调用库名、改变后的另一系统版本和第一设备体系架构获得符号映射表地址。

[0050] 符号映射表字典是按以下方式预先建立的:扫描硬盘上指定的符号表文件夹下的所有符号映射表文件,并将与符号映射表文件相关的设备体系结构、系统版本和调用库名与该符号映射表文件地址一起相关联地存储在符号映射表字典中。在一个示例中,符号表文件夹可以包括四个子文件夹,分别为【arm64】、【armv7】、【armv7s】和【self】。每个子文件夹又对应于一个具体字典。系统符号映射表文件和程序符号映射表文件按照体系架构不同,依次存储在前三个文件夹中。每个文件的命名方式按照【库名_系统version】。用户自己打包的程序符号映射表文件按照【程序名_version】存储在【self】文件夹中。

[0051] 根据调用库名、改变后的另一系统版本和第一设备体系架构获得符号映射表地址可以是利用模糊匹配方法执行的。具体地,针对特定体系架构,将当前系统版本改变为另一系统版本,并且根据调用库名、改变后的另一系统版本和第一设备体系架构利用预先建立的符号映射表字典查找符号映射表地址,当查找到符号映射表地址时,将查找到的符号映射表地址作为要获得的符号映射表地址;当未查找到符号映射表地址时,可以继续改变当前系统版本,直到找到要获得的符号映射表地址为止。备选地,也可以预先设置系统版本的改变次数(例如,6次)和改变的方式(例如,减小系统版本、增加系统版本等),当系统版本被连续改变预定次数之后仍然未能找到符号映射表地址,则过程结束,并且返回空值,以指示无法找到匹配的符号映射表文件。在执行模糊算法匹配的过程中,如果成功找到匹配的符号映射表文件,那么需要记录模糊匹配的映射关系和计数。如果没有成功找到匹配的符号映射表文件,那么也记录未能匹配的文件和计数。

[0052] 根据本公开的实施例,通过精细化的符号表文件管理、模糊匹配算法、符号化缓存和多进程并发,可以至少部分地解决无法对系统调用进行符号化以及无法快速地处理大量符号调用语句的问题,并因此可以实现支持对系统符号地址的符号化从而实现整个调用栈的完全符号化以帮助开发者更好的定位问题、批量处理符号地址调用栈(非常适合符号化大批量的崩溃数据)、对得到的符号化结果列表进行缓存以帮助下一次解析时快速符号化以及极大的提高了符号化的效率(效率提升在50倍以上,并且效率提升会随着运行次数的增多而越发的明显)的技术效果。

[0053] 图3示意性示出了根据本公开的实施例的用于对iOS符号调用语句进行符号化的装置300的框图。

[0054] 如图3所示,装置300可以包括接收模块310、分配模块320、和处理模块330。

[0055] 接收模块310可以被配置为接收待符号化的iOS符号调用语句和相关联的日志信息。

[0056] 分配模块320可以被配置为将所述iOS符号调用语句分配给多个进程。

[0057] 处理模块330可以被配置为针对每一个进程,根据与该进程相对应的每一个iOS符号调用语句和相关联的日志信息对iOS符号调用语句进行处理以获得与iOS符号调用语句相对应的符号化结果,并且将符号化结果记录到符号化结果列表中。

[0058] 处理模块330可以被进一步配置为:根据iOS符号调用语句和相关联的日志信息获取与iOS符号调用语句相对应的调用库名、系统版本、第一设备体系架构、调用语句偏移地

址和调用语句基地址;根据调用库名、第一设备体系架构和调用语句偏移地址在缓存中查找与iOS符号调用语句相对应的符号化结果;如果找到相对应的符号化结果,则将所找到的符号化结果确定为与iOS符号调用语句相对应的符号化结果。

[0059] 处理模块330可以被进一步配置为:如果未找到相对应的符号化结果,则根据调用库名、系统版本和第一设备体系架构获得符号映射表地址;根据符号映射表地址和第一设备体系架构获得第二设备体系架构;根据符号映射表地址、第二设备体系架构、调用语句基地址和调用语句偏移地址利用符号化语句对iOS符号调用语句进行符号化以获得符号化结果;以及将所获得的符号化结果与调用库名、第一设备体系架构和调用语句偏移地址相关联地存储在缓存中。符号化语句可以是atos语句或symbolicate语句。

[0060] 处理模块330可以被进一步配置为:对符号映射表地址执行lipo-info命令,以获得符号映射表支持的设备体系架构列表;确定第一设备体系架构是否存在于设备体系架构列表中;以及如果第一设备体系架构不存在于设备体系架构列表中,则将设备体系架构列表中比第一设备体系架构的版本更低的设备体系架构确定为第二设备体系架构。

[0061] 处理模块330可以被进一步配置为:如果第一设备体系架构存在于设备体系架构列表中,则将第一设备体系架构确定为第二设备体系架构。

[0062] 处理模块330可以被进一步配置为:根据调用库名、系统版本和第一设备体系架构利用预先建立的符号映射表字典查找符号映射表地址;当未查找到符号映射表地址时,将系统版本改变为另一系统版本;以及根据调用库名、改变后的另一系统版本和第一设备体系架构获得符号映射表地址。根据调用库名、改变后的另一系统版本和第一设备体系架构获得符号映射表地址是利用模糊匹配方法执行的。

[0063] 该装置与方法100相对应。以上对方法100的具体描述和解释同样适用于该装置,在此不再赘述。

[0064] 根据本公开的实施例,通过精细化的符号表文件管理、模糊匹配算法、符号化缓存和多进程并发,可以至少部分地解决无法对系统调用进行符号化以及无法快速地处理大量符号调用语句的问题,并因此可以实现支持对系统符号地址的符号化从而实现整个调用栈的完全符号化以帮助开发者更好的定位问题、批量处理符号地址调用栈(非常适合符号化大批量的崩溃数据)、对得到的符号化结果列表进行缓存以帮助下一次解析时快速符号化以及极大的提高了符号化的效率(效率提升在50倍以上,并且效率提升会随着运行次数的增多而越发的明显)的技术效果。

[0065] 图4示出了可以应用本公开实施例的用于对iOS符号调用语句进行符号化的方法和装置的示例性系统架构400。

[0066] 如图4所示,系统架构400可以包括终端设备401、402、403,网络404和服务器405。网络404用以在终端设备401、402、403和服务器405之间提供通信链路的介质。网络404可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0067] 用户可以使用终端设备401、402、403通过网络404与服务器405交互,以接收或发送消息等。终端设备401、402、403上可以安装有各种通讯客户端应用,例如购物类应用、网页浏览器应用、搜索类应用、即时通信工具、邮箱客户端、社交平台软件等(仅为示例)。

[0068] 终端设备401、402、403可以是具有显示屏并且支持网页浏览的各种电子设备,包括但不限于智能手机、平板电脑、膝上型便携计算机和台式计算机等等。

[0069] 服务器405可以是提供各种服务的服务器，例如对用户利用终端设备401、402、403所浏览的购物类网站提供支持的后台管理服务器(仅为示例)。后台管理服务器可以对接收到的产品信息查询请求等数据进行分析等处理，并将处理结果(例如目标推送信息、产品信息--仅为示例)反馈给终端设备。

[0070] 需要说明的是，本申请实施例所提供的用于对iOS符号调用语句进行符号化的方法一般由服务器405执行，相应地，用于对iOS符号调用语句进行符号化的装置一般设置于服务器405中。

[0071] 应该理解，图4中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要，可以具有任意数目的终端设备、网络和服务器。

[0072] 下面参考图5，其示出了适于用来实现本公开实施例的终端设备的计算机系统500的结构示意图。图5示出的计算机系统仅仅是一个示例，不应对本公开实施例的功能和使用范围带来任何限制。

[0073] 如图5所示，计算机系统500包括中央处理单元(CPU)501，其可以根据存储在只读存储器(ROM)502中的程序或者从存储部分508加载到随机访问存储器(RAM)503中的程序而执行各种适当的动作和处理。在RAM 503中，还存储有系统500操作所需的各种程序和数据。CPU 501、ROM 502以及RAM 503通过总线504彼此相连。输入/输出(I/O)接口505也连接至总线504。

[0074] 以下部件连接至I/O接口505：包括键盘、鼠标等的输入部分506；包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分507；包括硬盘等的存储部分508；以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分509。通信部分509经由诸如因特网的网络执行通信处理。驱动器510也根据需要连接至I/O接口505。可拆卸介质511，诸如磁盘、光盘、磁光盘、半导体存储器等等，根据需要安装在驱动器510上，以便于从其上读出的计算机程序根据需要被安装入存储部分508。

[0075] 特别地，根据本公开的实施例，上文参考流程图描述的过程可以被实现为计算机软件程序。例如，本公开的实施例包括一种计算机程序产品，其包括承载在计算机可读介质上的计算机程序，该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中，该计算机程序可以通过通信部分509从网络上被下载和安装，和/或从可拆卸介质511被安装。在该计算机程序被中央处理单元(CPU)501执行时，执行本公开的系统中限定的上述功能。

[0076] 需要说明的是，本申请所述的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本申请中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本申请中，计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限

于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括但不限于：无线、电线、光缆、RF等等，或者上述的任意合适的组合。

[0077] 作为另一方面，本申请还提供了一种计算机可读介质，该计算机可读介质可以是上述实施例中描述的设备中所包含的；也可以是单独存在，而未装配入该设备中。上述计算机可读介质承载有一个或者多个程序，当上述一个或者多个程序被一个该设备执行时，使得该设备执行根据本本公开实施例的用于对iOS符号调用语句进行符号化的方法。

[0078] 附图中的流程图和框图，图示了按照本申请各种实施例的系统、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，上述模块、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，两个接连地表示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。也要注意的是，框图或流程图中的每个方框、以及框图或流程图中的方框的组合，可以用执行规定的功能或操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。

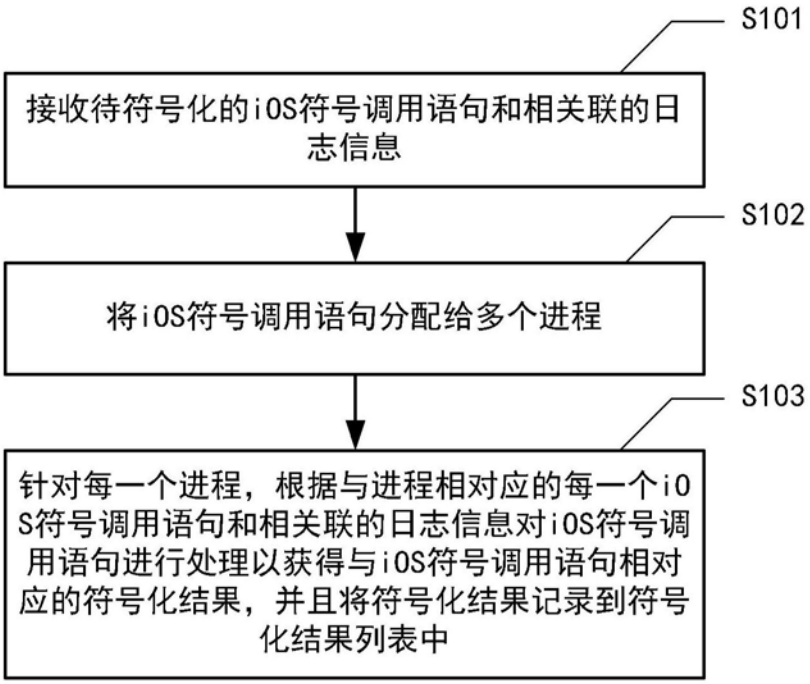

S101
接收待符号化的iOS符号调用语句和相关联的日志信息
S102
将iOS符号调用语句分配给多个进程
S103
针对每一个进程，根据与进程相对应的每一个iOS符号调用语句和相关联的日志信息对iOS符号调用语句进行处理以获得与iOS符号调用语句相对应的符号化结果，并且将符号化结果记录到符号化结果列表中

图1

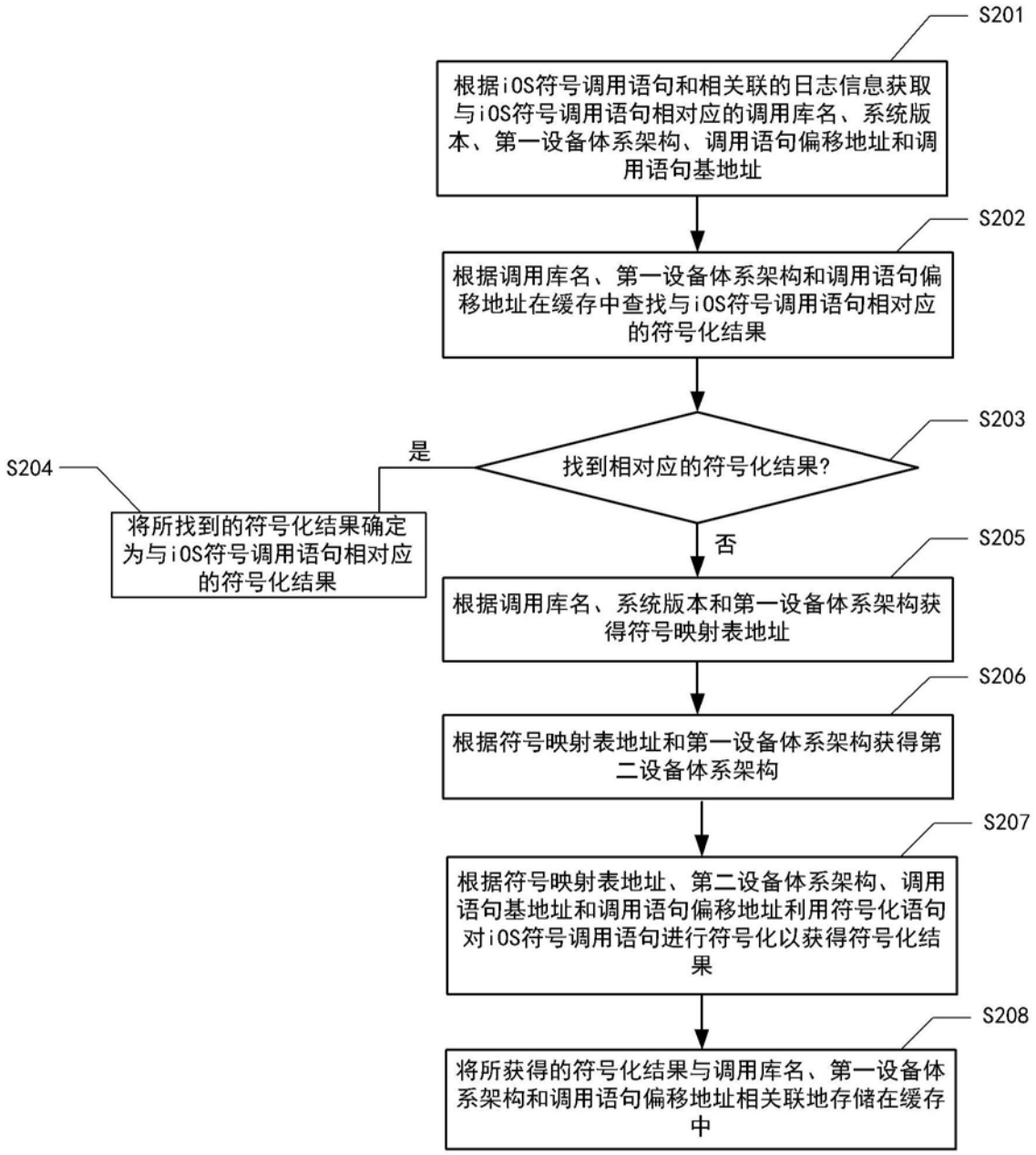

S201
根据iOS符号调用语句和相关联的日志信息获取与iOS符号调用语句相对应的调用库名、系统版本、第一设备体系架构、调用语句偏移地址和调用语句基地址
S202
根据调用库名、第一设备体系架构和调用语句偏移地址在缓存中查找与iOS符号调用语句相对应的符号化结果
S203
找到相对应的符号化结果?
是
S204
将所找到的符号化结果确定为与iOS符号调用语句相对应的符号化结果
否
S205
根据调用库名、系统版本和第一设备体系架构获得符号映射表地址
S206
根据符号映射表地址和第一设备体系架构获得第二设备体系架构
S207
根据符号映射表地址、第二设备体系架构、调用语句基地址和调用语句偏移地址利用符号化语句对iOS符号调用语句进行符号化以获得符号化结果
S208
将所获得的符号化结果与调用库名、第一设备体系架构和调用语句偏移地址相关联地存储在缓存中

图2

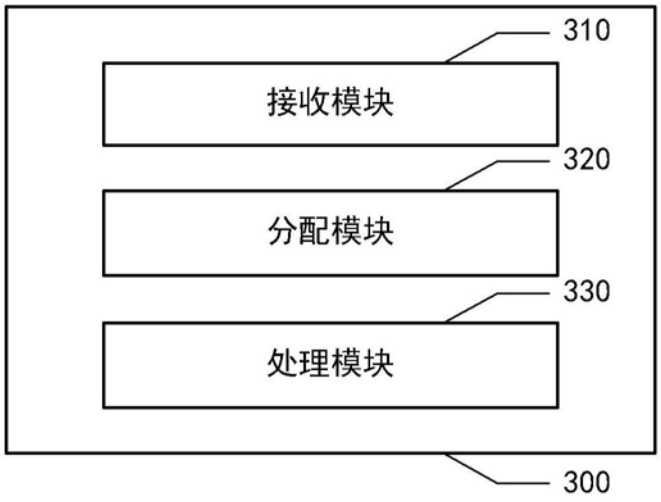


图3

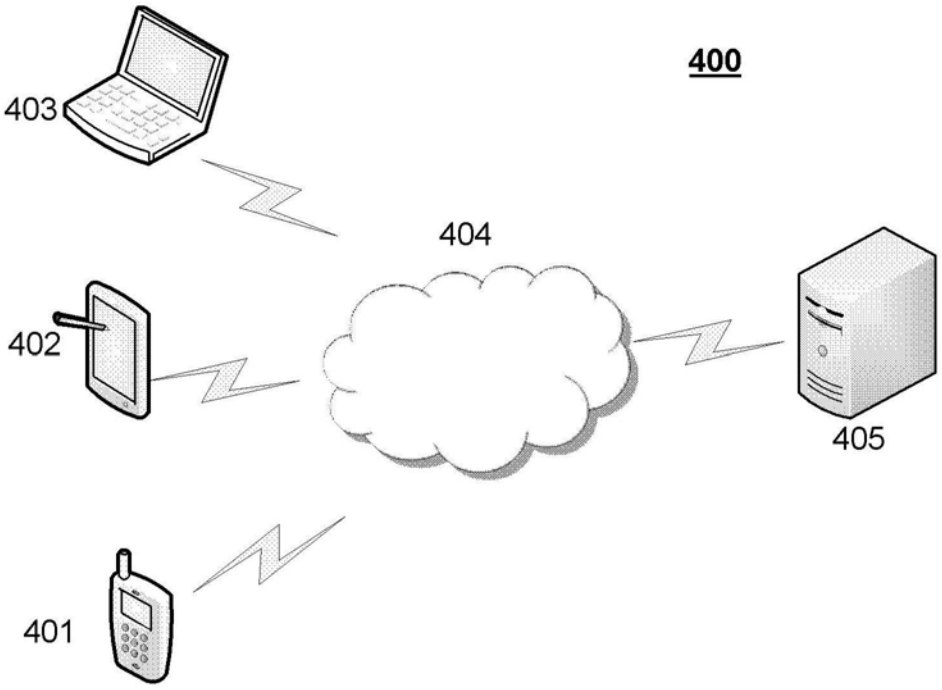


图4

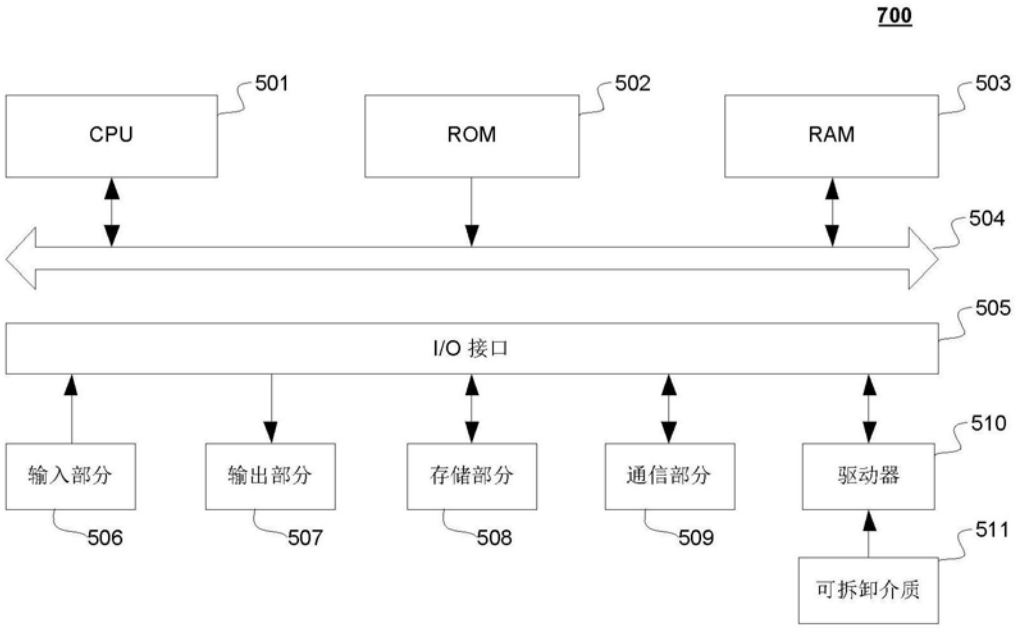

700
501
CPU
502
ROM
503
RAM
504
505
I/O 接口
输入部分
506
输出部分
507
存储部分
508
通信部分
509
510
驱动器
511
可拆卸介质

图5