(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115114117 A
(43)申请公布日 2022.09.27

(21)申请号 202210881085.3

(22)申请日 2022.07.26

(71)申请人 南方科技大学
地址 518055 广东省深圳市南山区桃源街道学苑大道1088号

(72)发明人 张锋巍 宁振宇 张一鸣

(74)专利代理机构 广州嘉权专利商标事务所有限公司 44205
专利代理师 林明校

(51)Int.Cl.
G06F 11/30(2006.01)
G06F 11/07(2006.01)

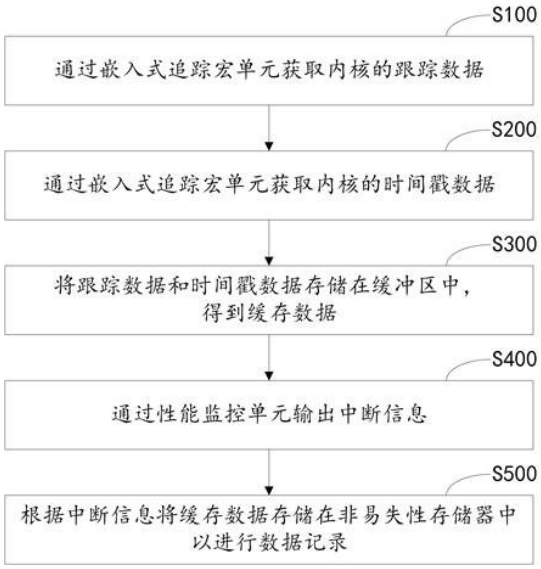
权利要求书2页 说明书9页 附图4页

(54)发明名称

数据记录方法和数据记录装置

(57)摘要

本申请提供一种数据记录方法和数据记录装置。该方法包括:通过嵌入式追踪宏单元获取内核的跟踪数据;通过嵌入式追踪宏单元获取内核的时间戳数据;将跟踪数据和时间戳数据存储在缓冲区中,得到缓存数据;通过性能监控单元输出中断信息;根据中断信息将缓存数据存储在非易失性存储器中以进行数据记录。通过使用ARM处理器内部的嵌入式追踪宏单元分别获取内核的跟踪数据和时间戳数据,并将其存储在缓冲区中,当接收到中断信息后,及时的将缓存数据存入非易失性存储器中。通过此种方式,可以完整的记录目标程序在运行时的数据流,用户通过分析非易失性存储器中存储的数据,即可以准确的恢复目标程序的数据流,以完成对目标程序的故障诊断。



CN 115114117 A

1.数据记录方法,其特征在于,应用于ARM处理器,所述方法包括:

通过嵌入式追踪宏单元获取内核的跟踪数据;

通过嵌入式追踪宏单元获取内核的时间戳数据;

将所述跟踪数据和所述时间戳数据存储在缓冲区中,得到缓存数据;

通过性能监控单元输出中断信息;

根据所述中断信息将所述缓存数据存储在非易失性存储器中以进行数据记录。

2.根据权利要求1所述的数据记录方法,其特征在于,所述通过嵌入式追踪宏单元获取内核的时间戳数据,包括:

将所述嵌入式追踪宏单元配置为输出时间戳模式;

通过倒计时计数器触发所述嵌入式追踪宏单元的跟踪单元事件;

根据所述跟踪单元事件输出所述时间戳数据。

3.根据权利要求1所述的数据记录方法,其特征在于,所述将所述跟踪数据和所述时间戳数据存储在缓冲区中,得到缓存数据,包括:

将所述跟踪数据和所述时间戳数据存储在嵌入式追踪路由器中,得到所述缓存数据。

4.根据权利要求1所述的数据记录方法,其特征在于,所述通过性能监控单元输出中断信息,包括:

通过所述性能监控单元检测所述缓冲区中所述缓存数据的数据大小;

若所述缓存数据的数据大小大于预设阈值,则输出所述中断信息。

5.根据权利要求1至4任一项所述的数据记录方法,其特征在于,所述方法还包括:

获取当前系统调用的调用类型;

根据所述调用类型选择对应的记录方式以进行数据记录。

6.根据权利要求5所述的数据记录方法,其特征在于,所述调用类型包括:读状态和写状态,所述根据所述调用类型选择对应的记录方式以进行数据记录,包括:

若所述调用类型为读状态,则对全部改变数据进行数据记录;

若所述调用类型为写状态,则仅对错误代码进行数据记录。

7.根据权利要求6所述的数据记录方法,其特征在于,所述调用类型还包括:读内容和写内容,所述根据所述调用类型选择对应的记录方式以进行数据记录,还包括:

若所述调用类型为读内容,则对部分截取数据进行数据记录;

若所述调用类型为写内容,则不进行数据记录。

8.根据权利要求5所述的数据记录方法,其特征在于,所述方法还包括:

将所述嵌入式追踪宏单元配置为跟踪异步异常模式;

获取所述嵌入式追踪宏单元检测的异步事件的事件时间;

若检测到所述异步事件的信号处理程序上的硬件断点,则进行核心转储以得到转储数据;

将所述事件时间和所述转储数据进行数据记录。

9.根据权利要求1至4任一项所述的数据记录方法,其特征在于,所述方法还包括:

若当前库函数为包装器中的目标库函数,则将所述当前库函数进行数据记录;其中,在包装器中,执行原始函数并保存执行信息。

10.数据记录装置,其特征在于,应用于ARM处理器,所述装置包括:

第一获取模块，所述第一获取模块用于通过嵌入式追踪宏单元获取内核的跟踪数据；

第二获取模块，所述第二获取模块用于通过嵌入式追踪宏单元获取内核的时间戳数据；

第一存储模块，所述第一存储模块用于将所述跟踪数据和所述时间戳数据存储在缓冲区中，得到缓存数据；

中断模块，所述中断模块用于通过性能监控单元输出中断信息；

第二存储模块，所述第二存储模块用于根据所述中断信息将所述缓存数据存储在非易失性存储器中以进行数据记录。

数据记录方法和数据记录装置

技术领域

[0001] 本申请涉及处理器技术领域，尤其涉及一种数据记录方法和数据记录装置。

背景技术

[0002] 实际系统中的故障诊断是很困难的，主要的障碍是，开发人员所能获得的信息是有限的，这些信息通常不足以帮助开发者修复或者定位相关的错误。此外，由于开发环境和生产环境之间的巨大差异，在开发环境中重现生产环境中的故障并不容易。

[0003] 现有的故障诊断系统，例如，REPT、Snorlax、RETracer、Credal、POMP，通常利用现代硬件(如英特尔处理器跟踪)来记录程序的控制流。例如，将控制流与程序崩溃时捕获的核心转储一起使用，以推断数据流。但是，因为涉及故障的内存和寄存器可能在核心转储被捕获之前就被覆盖了，所以依靠最终崩溃的核心转储所记录的数据并不完整。

发明内容

[0004] 本申请实施例的主要目的在于提出一种数据记录方法和数据记录装置，以提供一种应用于ARM处理器上的数据记录方案，可以完整的记录程序在运行时的控制流数据。

[0005] 为实现上述目的，本申请实施例的第一方面提出了一种数据记录方法，所述方法包括：

通过嵌入式追踪宏单元获取内核的跟踪数据；

通过嵌入式追踪宏单元获取内核的时间戳数据；

将所述跟踪数据和所述时间戳数据存储在缓冲区中，得到缓存数据；

通过性能监控单元输出中断信息；

根据所述中断信息将所述缓存数据存储在非易失性存储器中以进行数据记录。

[0006] 在一些实施例中，所述通过嵌入式追踪宏单元获取内核的时间戳数据，包括：

将所述嵌入式追踪宏单元配置为输出时间戳模式；

通过倒计时计数器触发所述嵌入式追踪宏单元的跟踪单元事件；

根据所述跟踪单元事件输出所述时间戳数据。

[0007] 在一些实施例中，所述将所述跟踪数据和所述时间戳数据存储在缓冲区中，得到缓存数据，包括：

将所述跟踪数据和所述时间戳数据存储在嵌入式追踪路由器中，得到所述缓存数据。

[0008] 在一些实施例中，所述通过性能监控单元输出中断信息，包括：

通过所述性能监控单元检测所述缓冲区中所述缓存数据的数据大小；

若所述缓存数据的数据大小大于预设阈值，则输出所述中断信息。

[0009] 在一些实施例中，数据记录方法还包括：

获取当前系统调用的调用类型；

根据所述调用类型选择对应的记录方式以进行数据记录。

[0010] 在一些实施例中，所述调用类型包括：读状态和写状态，所述根据所述调用类型选择对应的记录方式以进行数据记录，包括：

若所述调用类型为读状态，则对全部改变数据进行数据记录；

若所述调用类型为写状态，则仅对错误代码进行数据记录。

[0011] 在一些实施例中，所述调用类型还包括：读内容和写内容，所述根据所述调用类型选择对应的记录方式以进行数据记录，还包括：

若所述调用类型为读内容，则对部分截取数据进行数据记录；

若所述调用类型为写内容，则不进行数据记录。

[0012] 在一些实施例中，数据记录方法还包括：

将所述嵌入式追踪宏单元配置为跟踪异步异常模式；

获取所述嵌入式追踪宏单元检测的异步事件的事件时间；

若检测到所述异步事件的信号处理程序上的硬件断点，则进行核心转储以得到转储数据；

将所述事件时间和所述转储数据进行数据记录。

[0013] 在一些实施例中，数据记录方法还包括：

若当前库函数为包装器中的目标库函数，则将所述当前库函数进行数据记录；其中，在包装器中，执行原始函数并保存执行信息。

[0014] 为实现上述目的，本申请的第二方面提出了一种数据记录装置，应用于ARM处理器，所述装置包括：

第一获取模块，所述第一获取模块用于通过嵌入式追踪宏单元获取内核的跟踪数据；

第二获取模块，所述第二获取模块用于通过嵌入式追踪宏单元获取内核的时间戳数据；

第一存储模块，所述第一存储模块用于将所述跟踪数据和所述时间戳数据存储在缓冲区中，得到缓存数据；

中断模块，所述中断模块用于通过性能监控单元输出中断信息；

第二存储模块，所述第二存储模块用于根据所述中断信息将所述缓存数据存储在非易失性存储器中以进行数据记录。

[0015] 本申请实施例提出的数据记录方法和数据记录装置，通过使用ARM处理器内部的嵌入式追踪宏单元分别获取内核的跟踪数据和时间戳数据，并将其存储在缓冲区中，当接收到中断信息后，及时的将缓存数据存入非易失性存储器中。当目标程序启动或者一个新的线程被创建时，在相应状态下的内存信息就会被转移到缓冲区中，然后将所有记录的信息都保存到非易失性存储器上。通过此种方式，可以完整的记录目标程序在运行时的数据流，用户通过分析非易失性存储器中存储的数据，即可以准确的恢复目标程序的数据流，以完成对目标程序的故障诊断。

附图说明

[0016] 图1是本申请一实施例提供的数据记录方法的流程图；

图2是图1中步骤S200的流程图；

图3是图1中步骤S400的流程图；

图4是本申请另一实施例提供的数据记录方法的流程图；

图5是本申请又一实施例提供的数据记录方法的流程图；

图6是本申请一实施例提供的数据记录方法的示意图；

图7是本申请实施例提供的数据记录装置的示意图。

具体实施方式

[0017] 为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实施例仅用以解释本申请，并不用于限定本申请。

[0018] 需要说明的是，虽然在装置示意图中进行了功能模块划分，在流程图中示出了逻辑顺序，但是在某些情况下，可以以不同于装置中的模块划分，或流程图中的顺序执行所示出或描述的步骤。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。

[0019] 除非另有定义，本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同。本文中所使用的术语只是为了描述本申请实施例的目的，不是旨在限制本申请。

[0020] 此外，所描述的特征、结构或特性可以以任何合适的方式结合在一个或更多实施例中。在下面的描述中，提供许多具体细节从而给出对本公开的实施例的充分理解。然而，本领域技术人员将意识到，可以实践本公开的技术方案而没有特定细节中的一个或更多，或者可以采用其它的方法、组元、装置、步骤等。在其它情况下，不详细示出或描述公知方法、装置、实现或者操作以避免模糊本公开的各方面。

[0021] 附图中所示的方框图仅仅是功能实体，不一定必须与物理上独立的实体相对应。即，可以采用软件形式来实现这些功能实体，或在一个或多个硬件模块或集成电路中实现这些功能实体，或在不同网络和/或处理器装置和/或微控制器装置中实现这些功能实体。

[0022] 附图中所示的流程图仅是示例性说明，不是必须包括所有的内容和操作/步骤，也不是必须按所描述的顺序执行。例如，有的操作/步骤还可以分解，而有的操作/步骤可以合并或部分合并，因此实际执行的顺序有可能根据实际情况改变。

[0023] 首先，对本申请中涉及的若干名词进行解析：

随机存取存储器(Random Access Memory，RAM)：也叫主存，是与中央处理器直接交换数据的内部存储器，通常作为操作系统或其他正在运行中的程序的临时数据存储介质。RAM中存储的信息在断电后均会丢失，是易失性存储器。

[0024] 非易失性存储器(Non-Volatile Memory，NVM)：是指当电流关掉后，所存储的数据不会消失的电脑存储器。依存储器内的数据是否能在使用时随时改写为标准，可分为二大类产品，即只读存储器(Read-Only Memory，ROM)和快闪存储器(Flash Memory)。

[0025] 嵌入式追踪宏单元(Embedded Trace Macrocell，ETM)：用于获取处理器核的跟踪数据。

[0026] 嵌入式追踪缓冲区(Embedded Trace Buffer，ETB)：是一个跟踪接收器，它使用一定大小的RAM为跟踪数据提供芯片上存储。

[0027] 嵌入式追踪路由器(Embedded Trace Router,ETR):功能类似于ETB,但允许用户使用设备中的任何RAM来存储跟踪数据。

[0028] 性能监控单元(Performance Monitor Unit,PMU):运行时可以收集关于处理器和内存的各种统计信息。

[0029] 本申请实施例提供数据记录方法和数据记录装置,具体通过如下实施例进行说明,首先描述本申请实施例中的数据记录方法。

[0030] 图1是本申请实施例提供的数据记录方法的一个可选的流程图,应用于ARM处理器中,图1中的方法可以包括但不限于包括步骤S100至步骤S500。

[0031] S100,通过嵌入式追踪宏单元获取内核的跟踪数据;

S200,通过嵌入式追踪宏单元获取内核的时间戳数据;

S300,将跟踪数据和时间戳数据存储在缓冲区中,得到缓存数据;

S400,通过性能监控单元输出中断信息;

S500,根据中断信息将缓存数据存储在非易失性存储器中以进行数据记录。

[0032] 本申请实施例提出的数据记录方法,通过使用ARM处理器内部的嵌入式追踪宏单元分别获取内核的跟踪数据和时间戳数据,并将其存储在缓冲区中,当接收到中断信息后,及时的将缓存数据存入非易失性存储器中。当目标程序启动或者一个新的线程被创建时,在相应状态下的内存信息就会被转移到缓冲区中,然后将所有记录的信息都保存到非易失性存储器上。通过此种方式,可以完整的记录目标程序在运行时的数据流,用户通过分析非易失性存储器中存储的数据,即可以准确的恢复目标程序的数据流,以完成对目标程序的故障诊断。

[0033] 步骤S100中,通过ARM处理器中的嵌入式追踪宏单元获取内核的跟踪数据。可以理解的是,为了获取ARM处理器的完整且准确的控制流程,嵌入式追踪宏单元需要始终处于开启状态,以便持续追踪,获取到足够的跟踪数据。其中,跟踪数据中包括了通过指令跟踪和数据跟踪获取的数据信息。

[0034] 可以理解的是,在多核系统中,每一个内核都具有自己的嵌入式追踪宏单元,并且每一个嵌入式追踪宏单元只能追踪自己所属的内核执行的指令,即只能获取自己所属内核的跟踪数据。因此,在记录的数据中可以很容易的获得单个内核中指令的执行顺序,却无法得到多个内核之间指令执行的先后,导致无法检测出并行错误。例如,无法检测出由竞争条件引起的故障。因此,通过步骤S200,使嵌入式追踪宏单元在获取跟踪数据的同时,获取时间戳数据。

[0035] 在一些实施例中,参照图2,在步骤S200中,通过嵌入式追踪宏单元获取内核的时间戳数据,包括:

S210,将嵌入式追踪宏单元配置为输出时间戳模式;

S220,通过倒计时计数器触发嵌入式追踪宏单元的跟踪单元事件;

S230,根据跟踪单元事件输出时间戳数据。

[0036] 首先通过步骤S210,将嵌入式追踪宏单元配置为额外输出时间戳的模式,时间戳中包含了CPU的时钟计时信息。由于嵌入式追踪宏单元输出时间戳的默认生成频率较低,会相隔若干个跟踪包后才输出一次时间戳,此种方式在一些比较密集的竞争条件发生时是不足以判断指令先后的。而每当有特殊的跟踪单元事件发生时,嵌入式追踪宏单元都可以生

成时间戳。因此通过步骤S220，利用嵌入式追踪宏单元内置的倒计时计数器作为外部来源，来触发跟踪单元事件，嵌入式追踪宏单元通过步骤S230，检测跟踪单元事件，即可输出对应的时间戳数据。

[0037] 具体示例，倒计时计数器可以在计数器值减少到0时触发跟踪单元事件，因此，可以将倒计时计数器的初始值和重载值均配置为0，使跟踪单元事件可以一直发生，从而使嵌入式追踪宏单元以最大的频率生成时间戳。在一些其他实施例中，也可以根据目标程序的特点，灵活选择生成时间戳的频率。

[0038] 本申请实施例的数据记录方法，通过同时获取时间戳数据和跟踪数据，在后续的故障分析过程中，可以判断出多个内核中执行指令的先后关系，方便后续的故障分析。

[0039] 可以理解的是，嵌入式追踪宏单元的默认片上缓冲区为嵌入式追踪缓冲区，用于对时间戳数据和跟踪数据提供芯片上存储。嵌入式追踪缓冲区的容量通常是有限的(例如ARM Juno开发板上为64KB)，由于CPU的执行速度非常快，嵌入式追踪缓冲区在几秒钟之内就被填满，且由于嵌入式追踪缓冲区在填满后并不能引发中断，因此在嵌入式追踪缓冲区填满的情况下，其中存储的时间戳数据和跟踪数据会被新数据覆盖，导致数据丢失。

[0040] 在一些实施例中，步骤S300中，将跟踪数据和时间戳数据存储在缓冲区中，得到缓存数据，包括：将跟踪数据和时间戳数据存储在嵌入式追踪路由器中，得到缓存数据。本申请实施例将嵌入式追踪路由器作为嵌入式追踪宏单元的缓冲区，嵌入式追踪路由器允许用户使用设备中的任何RAM来存储跟踪数据和时间戳数据。例如，可以向嵌入式追踪路由器分配一个高达4GB物理内存的缓冲区来提供给跟踪数据和时间戳数据进行临时存放，可以大大减小缓存数据向非易失性存储器中转移数据的频率。

[0041] 在一些实施例中，参照图3，步骤S400中，通过性能监控单元输出中断信息，包括：

S410，通过性能监控单元检测缓冲区中缓存数据的数据大小；

S420，若缓存数据的数据大小大于预设阈值，则输出中断信息。

[0042] 由于缓冲区被填满后并不会引发中断，因此通过ARM处理器中的性能监控单元来输出中断信息。性能监控单元在运行时可以收集关于处理器和内存的各种统计信息，因此通过步骤S410，使性能监控单元实时检测缓冲区中缓存数据的数据大小。由于缓冲区的存储空间是一定的，因此通过步骤S420，当检测到缓存数据的数据大小大于预设阈值后，说明缓冲区即将填满，此时即控制性能监控单元输出中断信息，指示缓存数据转移。其中，预设阈值的大小可以根据缓冲区的存储空间的大小进行设置。

[0043] 通过步骤S500，当检测到中断信息时，即需要将缓存数据转移至非易失性存储器中进行保存，通过此种数据记录方式，即可完整的记录目标程序在运行过程中的追踪数据和时间戳数据，便于后续对指令执行过程进行分析。

[0044] 为了记录非确定性事件的影响，可以为用户空间和内核空间都配置嵌入式追踪宏单元，但此种方式会引入无法接受的巨量开销(在几秒钟内产生GB级的跟踪数据)，因此对于非确定事件而言，本申请实施例考虑两种主要类别的非确定性事件：系统调用和异步事件。

[0045] 在一些实施例中，对于系统调用，参照图4，本申请的数据记录方法还包括：

S600，获取当前系统调用的调用类型；

S700，根据调用类型选择对应的记录方式以进行数据记录。

[0046] 首先通过步骤S600,判断当前系统调用的调用类型,然后通过步骤S700,根据判断得到的调用类型来选择与调用类型对应的记录方式来进行数据记录。可以理解的是,本申请实施例的数据记录为直接将数据记录在非易失性存储器中。

[0047] 在一些实施例中,调用类型包括:读状态和写状态,根据调用类型选择对应的记录方式以进行数据记录,包括:

若调用类型为读状态,则对全部改变数据进行数据记录;

若调用类型为写状态,则仅对错误代码进行数据记录。

[0048] 当检测得到的调用类型为读状态(Reading Status)时,例如,函数为getpid等。RS-Type系统调用读取与系统状态有关的信息,这些系统调用的结果可以通过返回值转移。因此需要直接记录改变的内存或寄存器,即对全部的改变数据进行数据记录。

[0049] 当检测得到的调用类型为写状态(Writing Status)时,例如,函数为epoll_create等。WS型系统调用改变系统的状态,但不直接改变程序的内存和寄存器。因此,在正常情况下忽略他们,除非调用失败并返回了一个错误代码,此时,仅对错误代码进行数据记录。

[0050] 在一些实施例中,调用类型还包括:读内容和写内容,根据调用类型选择对应的记录方式以进行数据记录,还包括:

若调用类型为读内容,则对部分截取数据进行数据记录;

若调用类型为写内容,则不进行数据记录。

[0051] 当检测得到的调用类型为读内容(Reading Content)时,例如,函数为read等。RC型系统调用从外部输入读取内容。由于RC型系统调用的内容通常较多,出于性能考虑,本申请实施例在记录时,仅会对部分截取数据进行数据记录。例如,在记录此类系统调用时截断内容,只记录前256字节。可以理解的是,截取的数据大小可以任意设置。

[0052] 当检测得到的调用类型为写内容(Writing Content)时,例如,函数为write等。WC型系统调用将内容写入外部源。此种情况下,不会影响目标程序的执行,因此忽略他们,不进行数据记录。

[0053] 通过对调用类型进行分类和检测,可以针对不同类型的系统调用采取不同的记录方式,从而大大减少捕获开销。

[0054] 在一些实施例中,对于异步事件,参照图5,本申请的数据记录方法还包括:

S810,将嵌入式追踪宏单元配置为跟踪异步异常模式;

S820,获取嵌入式追踪宏单元检测的异步事件的事件时间;

S830,若检测到异步事件的信号处理程序上的硬件断点,则进行核心转储以得到转储数据;

S840,将事件时间和转储数据进行数据记录。

[0055] 为了更好的处理异步事件(如中断事件),需要在分析阶段确定异步事件的时间。因此,通过步骤S810,将嵌入式追踪宏单元配置为跟踪异步异常模式(IRQ中断模式和FIQ快速中断模式)。在此种模式下,当发生异步事件时,可以通过步骤S820,获取到异步事件的事件时间。

[0056] 为了捕捉目标程序的信号处理程序的效果,首先从二进制中静态分析signal函数的用法,以预先知道哪些函数属于信号处理程序,然后在信号处理程序的地址上设置一个

硬件断点。当通过步骤S830,程序碰到断点时,将信号处理程序的堆栈框架和内核设置的寄存器的内容记录为核心转储,以得到转储数据。最后通过步骤S840,将事件时间和转储数据进行数据记录。在后续的分析阶段,可以使用记录的信号处理程序的内容来帮助恢复数据流。

[0057] 在一些实施例中,本申请的数据记录方法还包括:若当前库函数为包装器中的目标库函数,则将当前库函数进行数据记录;其中,在包装器中,执行原始函数并保存执行信息。

[0058] 在数据记录时,记录所有的库函数是没有必要的。库函数可以分为两类:确定性的函数和非确定性的函数。确定性函数(例如,math.sqrt)不需要记录,因为它们的效果可以在分析阶段重现。一些非确定性函数(例如,一些可移植操作系统接口)本质上只是系统调用的包装,因此需要忽略它们以避免重复记录。关于其他非确定性函数或开发者指定的库函数,则需要进行数据记录。因此,通过判断当前库函数是否为包装器中的目标库函数,来判断当前库函数是否需要进行数据记录。可以理解的是,目标库函数为用户原先设置的一些需要记录的函数。

[0059] 本申请实施例中,通过制作一个工具以自动对每个目标库函数生成包装器,在包装器中,执行原始函数并保存执行信息。在进行数据记录时,需要使用库函数钩子来收集它们对内存和寄存器的影响。具体的,可以通过修改系统中的动态链接器的重定位过程来挂钩库函数。因为动态链接器在加载程序时是通过搜索和重定位有关库函数的地址来完成重定位的,通过改变这个过程,可以把动态链接器重定位到的钩子函数上。本申请实施例通过自动化的钩子过程,可以减少大量的手工工程,并使库函数钩子更加通用化。

[0060] 参照图6,下面以一个具体的实施例来详细描述本申请的数据记录方法。本申请实施例的记录阶段主要包含三个单元,嵌入式追踪宏单元管理器、非确定性事件捕捉器和库函数钩子。嵌入式追踪宏单元管理器是一个内核模块,用于控制硬件功能(嵌入式追踪宏单元和嵌入式追踪路由器),嵌入式追踪宏单元管理器通过执行步骤S100、步骤S200和步骤S300,来得到包括跟踪数据和时间戳数据的缓存数据,并在接收到性能监控单元输出的中断信息时,通过执行步骤S500,将缓存数据转移至非易失性存储器中。非确定性事件捕捉器用于执行步骤S600、步骤S700、步骤S810至步骤S840,以记录与目标程序有关的非决定性事件(系统调用和异步事件)并将记录的事件数据存储至非易失性存储器中。库函数钩子检查由目标程序调用的库函数并提取相关信息,每当目标程序启动,或者一个新的线程被创建,它在相应状态下的内存信息就会进行核心转储,这个核心转储被保存作为初始快照。通过上述三个单元记录得到的信息都被保存到非易失性存储器中,并在故障发生时转移到离线服务器上,随后在这个服务器上执行分析阶段。

[0061] 分析阶段可以采用现有的数据分析方法。控制流构造器利用嵌入式追踪宏单元的跟踪数据、时间戳数据和程序的二进制来重建控制流。嵌入式追踪宏单元只记录关键(如分支和条件)指令的地址。因此,需要将嵌入式追踪宏单元d的跟踪数据与程序的二进制结合起来,以恢复控制流。具体来说,从程序的二进制文件中获得嵌入式追踪宏单元跟踪的每两个连续地址之间的指令,然后重建整个控制流。对于在多个处理器上运行的程序,相对于细粒度时间戳的准确性,与数据竞争相关的指令执行得并不紧密。也就是说,这些指令大多数时候位于不同的块中,而细粒度的时间戳可以确定这些块的访问顺序。因此通过结合时间

戳数据，即可以确定数据竞争的顺序。

[0062] 数据流构造器综合数据记录和控制流信息。数据流提供了每条指令后内存和寄存器的状态，这对开发人员确定故障的根本原因至关重要。相关的系统通过前向和后向分析恢复数据流。然而，后向分析给数据流引入了不确定因素，因为有些指令是不可逆的。例如，指令EOR r0,r0,r0清除了寄存器r0，因此不能用后向分析来推断存储在r0的原始值。后向分析在长时间运行的应用中也是往往不足的，因为数据恢复的准确性会随着追踪数据的增加而下降。因此，使用记录阶段收集的信息(如非决定性事件、库函数、核心转储)进行前向分析，以重建数据流，同时由于在记录阶段为长期跟踪，使得本申请实施例的数据记录方法能够支持长期运行的应用程序。

[0063] 对于常见的指令类型，通过研究其语义信息，并推断出每条指令执行后的内存和寄存器的状态。对于执行过程中的非确定性事件，通过解析记录的信息(例如，系统调用和信号处理程序)来恢复内存和寄存器的状态。同时通过检查传递的参数是否与数据流中恢复的值一致，并应用记录的寄存器和内存变化。

[0064] 程序故障根本原因探测器用于探测程序故障的根本原因。故障的根本原因定义为满足以下要求的一组指令和输入：参与了故障的执行，并且与故障的发生最相关；包含必要的指令，改变这些指令可以修复这个错误。具体来说，该探测器首先对重建的控制流进行类似于先前工作的静态别名分析，以获得一个点到集，这在内存地址之间建立了一个传递关系。其次，通过点到集和重建数据流中被访问的确切内存地址，过滤出一小部分与崩溃的内存有关的候选指令。对于并行程序，探测器在所有的可能指令中提取特征，当有更多的正常执行时，通过预测消除无用的特征，并在离线设备中重新执行逐一测试其余的特征，以其中一种能够触发错误确定为这个错误的根本原因。对顺序程序的检测与多线程程序类似，但探测器不必提取和采用特征，而是从程序崩溃的位置向前回溯损坏的数据值和相应的控制流(如输入源、导致数据损坏的指令块、崩溃的位置)，并将其识别为根本原因。

[0065] 本申请实施例的数据记录方法通过利用ARM处理器的硬件特征，有效地重建了与故障有关的精确控制和数据流，且本方法支持ARM平台上的未经修改的二进制文件，并无需修改硬件。在数据记录时可以追踪在生产环境中长时间运行的应用程序，为开发人员提供足够的信息进行根本原因分析。

[0066] 在一些实施例中，参照图7，本申请还提出一种数据记录装置，应用于ARM处理器，数据记录装置包括：

第一获取模块，第一获取模块用于通过嵌入式追踪宏单元获取内核的跟踪数据；

第二获取模块，第二获取模块用于通过嵌入式追踪宏单元获取内核的时间戳数据；

第一存储模块，第一存储模块用于将跟踪数据和时间戳数据存储在缓冲区中，得到缓存数据；

中断模块，中断模块用于通过性能监控单元输出中断信息；

第二存储模块，第二存储模块用于根据中断信息将缓存数据存储在非易失性存储器中以进行数据记录。

[0067] 本申请实施例提出的数据记录装置，通过使用ARM处理器内部的嵌入式追踪宏单元分别获取内核的跟踪数据和时间戳数据，并将其存储在缓冲区中，当接收到中断信息后，

及时的将缓存数据存入非易失性存储器中。当目标程序启动或者一个新的线程被创建时，在相应状态下的内存信息就会被转移到缓冲区中，然后将所有记录的信息都保存到非易失性存储器上。用户通过分析非易失性存储器中存储的数据，即可以准确的恢复目标程序的数据流，从而完成对目标程序的故障诊断。本申请实施例的数据记录装置用于执行上述实施例中的数据记录方法，其具体的执行步骤与上述实施例中的数据记录方法相同，此处不再一一赘述。

[0068] 本申请实施例描述的实施例是为了更加清楚的说明本申请实施例的技术方案，并不构成对于本申请实施例提供的技术方案的限定，本领域技术人员可知，随着技术的演变和新应用场景的出现，本申请实施例提供的技术方案对于类似的技术问题，同样适用。

[0069] 本领域技术人员可以理解的是，图中示出的技术方案并不构成对本申请实施例的限定，可以包括比图示更多或更少的步骤，或者组合某些步骤，或者不同的步骤。

[0070] 本领域普通技术人员可以理解，上文中所公开方法中的全部或某些步骤、系统、设备中的功能模块/单元可以被实施为软件、固件、硬件及其适当的组合。

[0071] 应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0072] 上面结合附图对本申请实施例作了详细说明，但是本申请不限于上述实施例，在所属技术领域普通技术人员所具备的知识范围内，还可以在不脱离本申请宗旨的前提下作出各种变化。此外，在不冲突的情况下，本申请的实施例及实施例中的特征可以相互组合。

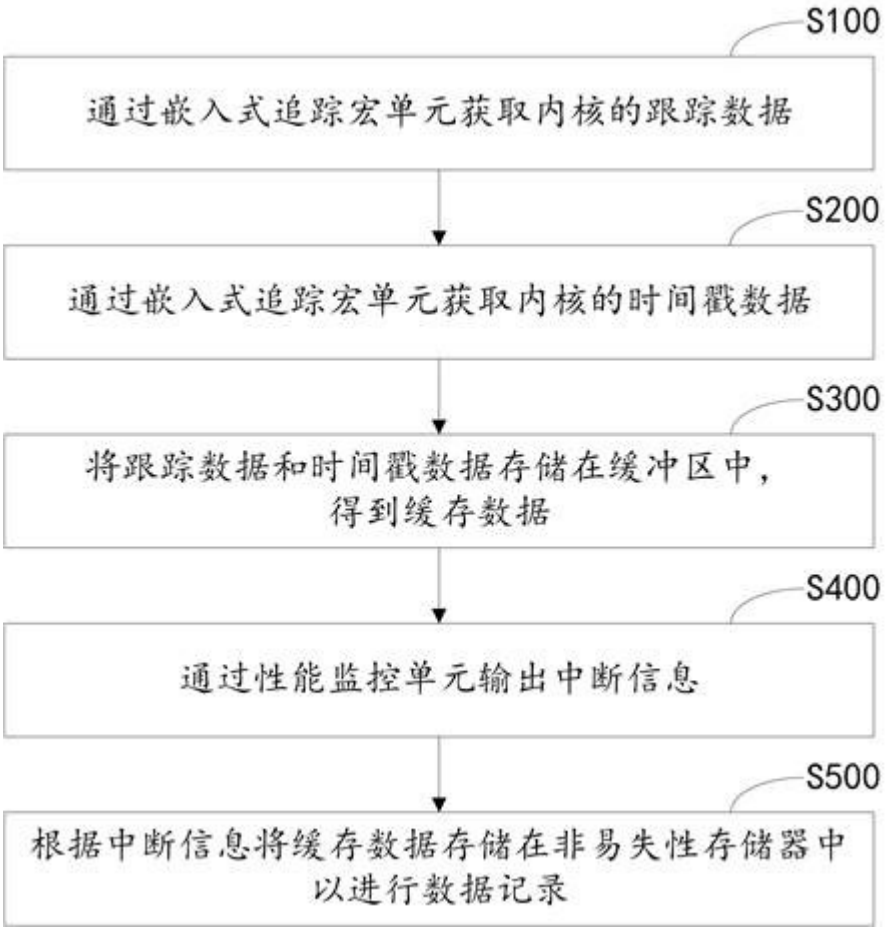

S100
通过嵌入式追踪宏单元获取内核的跟踪数据
S200
通过嵌入式追踪宏单元获取内核的时间戳数据
S300
将跟踪数据和时间戳数据存储在缓冲区中，得到缓存数据
S400
通过性能监控单元输出中断信息
S500
根据中断信息将缓存数据存储在非易失性存储器中以进行数据记录

图1

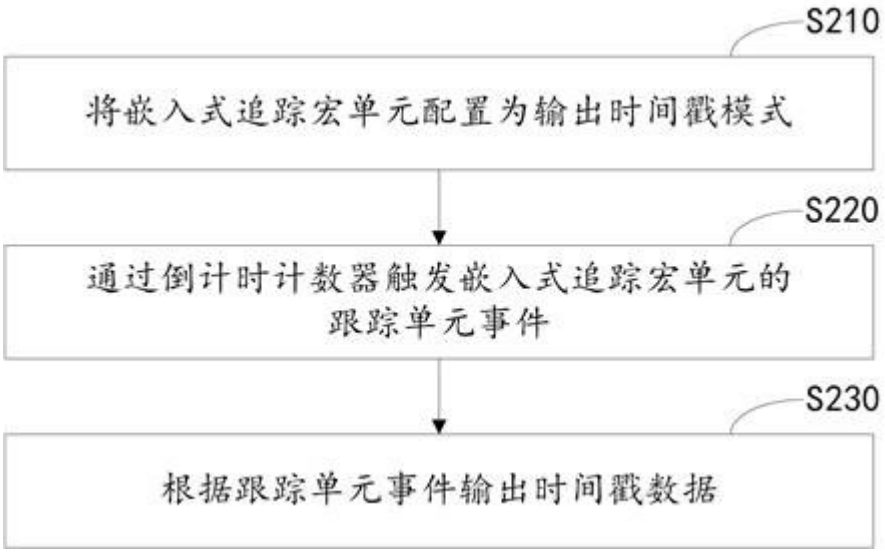

S210
将嵌入式追踪宏单元配置为输出时间戳模式
S220
通过倒计时计数器触发嵌入式追踪宏单元的跟踪单元事件
S230
根据跟踪单元事件输出时间戳数据

图2


通过性能监控单元检测缓冲区中缓存数据的数据大小
S410
若缓存数据的数据大小大于预设阈值，则输出中断信息
S420

图3


S600
获取当前系统调用的调用类型
S700
根据调用类型选择对应的记录方式以进行数据记录

图4

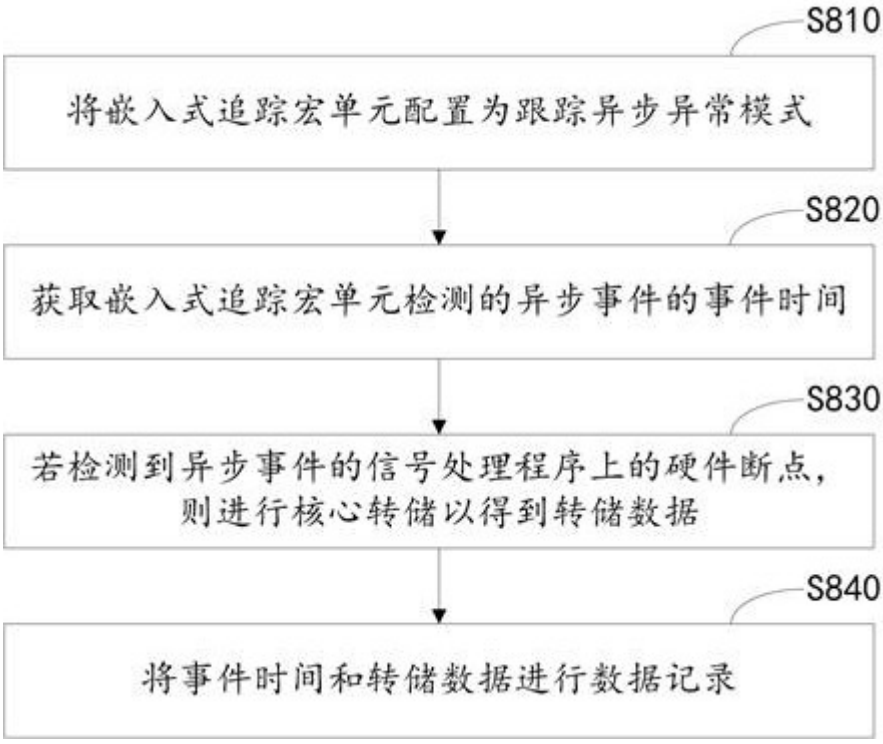

S810
将嵌入式追踪宏单元配置为跟踪异步异常模式
S820
获取嵌入式追踪宏单元检测的异步事件的事件时间
S830
若检测到异步事件的信号处理程序上的硬件断点，
则进行核心转储以得到转储数据
S840
将事件时间和转储数据进行数据记录

图5

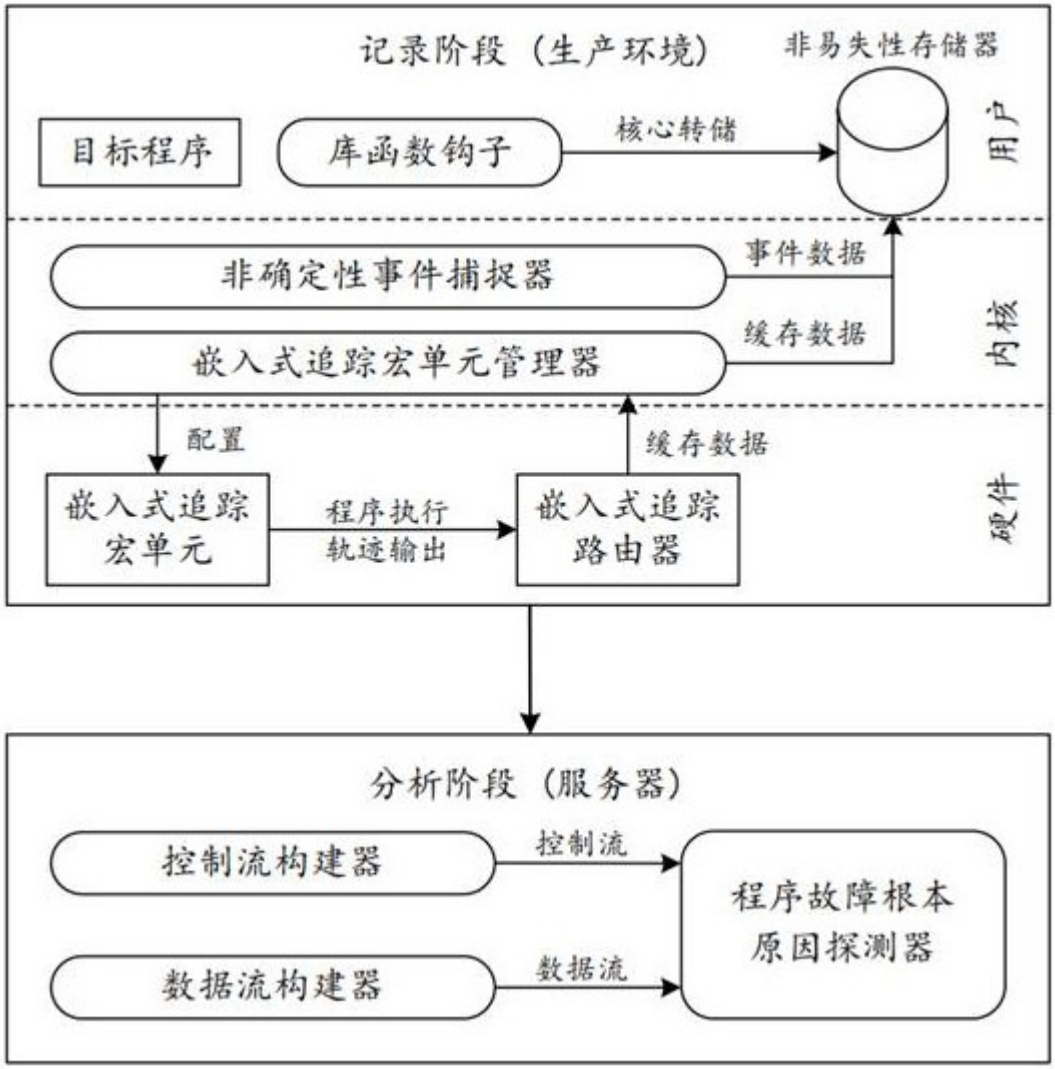

记录阶段（生产环境）
非易失性存储器
目标程序
库函数钩子
核心转储
用户
非确定性事件捕捉器
事件数据
缓存数据
嵌入式追踪宏单元管理器
内核
配置
缓存数据
嵌入式追踪宏单元
程序执行轨迹输出
嵌入式追踪路由器
硬件
分析阶段（服务器）
控制流构建器
控制流
程序故障根本原因探测器
数据流构建器
数据流

图6

图7