(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 108984329 A
(43)申请公布日 2018.12.11

(21)申请号 201810546266.4

(22)申请日 2018.05.31

(30)优先权数据
15/610067 2017.05.31 US

(71)申请人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 S.潘达 S.贾亚库马尔 G.波瓦尔 T.伊格梭

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 黄涛 申屠伟进

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

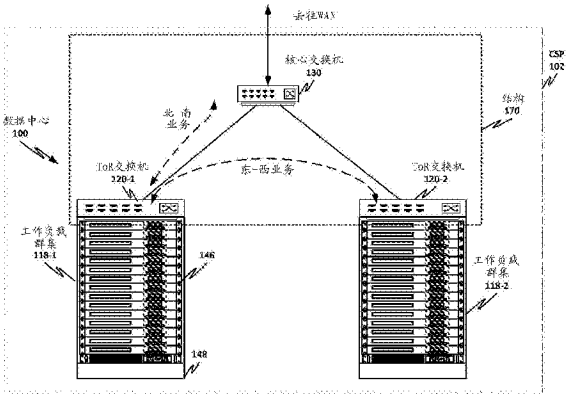
权利要求书2页 说明书16页 附图7页

(54)发明名称

延迟错误处理

(57)摘要

一种计算设备，包括：硬件平台，包括处理器和存储器；和系统管理中断(SMI)处置器；第一逻辑，被配置为经硬件平台提供第一容器和第二容器；和第二逻辑，被配置为：检测第一容器中的不可校正错误；响应于检测，产生降级系统状态；将降级状态消息提供给SMI处置器；指令第二容器寻找可恢复状态；确定第二容器已进入可恢复状态；以及开始恢复操作。



CN 108984329 A

1.一种计算设备,包括:

硬件平台,包括处理器和存储器;和

系统管理中断(SMI)处置器;

第一逻辑,被配置为经硬件平台提供第一容器和第二容器;和

第二逻辑,被配置为:

检测第一容器中的不可校正错误;

响应于检测,产生降级系统状态;

将降级状态消息提供给所述SMI处置器;

指令第二容器寻找可恢复状态;

确定第二容器已进入可恢复状态;以及

开始恢复操作。

2.如权利要求1所述的计算设备,其中所述第二逻辑还被配置为设置超时以及在超时期满之后开始所述恢复操作。

3.如权利要求1所述的计算设备,还包括结构接口;其中所述第二逻辑还将降级通知提供给控制器。

4.如权利要求3所述的计算设备,其中所述第二逻辑还请求所述控制器产生由第一容器提供的服务的新实例。

5.如权利要求1所述的计算设备,其中所述可恢复状态包括第二容器可以在最小数据丢失的情况下被迁移的状态。

6.如权利要求5所述的计算设备,其中所述第二逻辑还被配置为迁移第二容器。

7.如权利要求1-6中任一项所述的计算设备,还包括:操作系统,所述操作系统还被配置为执行第一容器的核心转储。

8.如权利要求7所述的计算设备,其中所述操作系统被配置为从所述处理器接收机器检查架构(MCA)记录信息。

9.如权利要求7所述的计算设备,其中所述第二逻辑还被配置为向所述操作系统通知所述设备的降级状态。

10.如权利要求1-6中任一项所述的计算设备,其中所述第二逻辑还包括配置接口,所述配置接口被配置为接收配置选项。

11.一个或多个有形非暂态计算机可读介质,具有存储在其上的用于提供逻辑的指令,所述逻辑用于:

提供系统管理中断(SMI)处置器;

提供第一容器和第二容器;

检测第一容器中的不可校正错误;

响应于检测,产生降级系统状态;

将降级状态消息提供给SMI处置器;

指令第二容器寻找可恢复状态;

确定第二容器已进入可恢复状态;以及

开始恢复操作。

12.如权利要求11所述的一个或多个有形非暂态计算机可读介质,其中所述逻辑还被

配置为设置超时以及在超时期满之后开始所述恢复操作。

13.如权利要求11所述的一个或多个有形非暂态计算机可读介质，其中所述逻辑还用于经结构接口将降级通知提供给控制器。

14.如权利要求13所述的一个或多个有形非暂态计算机可读介质，其中所述逻辑还用于请求所述控制器产生由第一容器提供的服务的新实例。

15.如权利要求11所述的一个或多个有形非暂态计算机可读介质，其中所述可恢复状态包括第二容器可以在最小数据丢失的情况下被迁移的状态。

16.如权利要求15所述的一个或多个有形非暂态计算机可读介质，其中所述逻辑还被配置为迁移第二容器。

17.如权利要求11-16中任一项所述的一个或多个有形非暂态计算机可读介质，其中所述逻辑还被配置为提供操作系统，所述操作系统被配置为执行第一容器的核心转储。

18.如权利要求17所述的一个或多个有形非暂态计算机可读介质，其中所述操作系统被配置为从所述处理器接收机器检查架构(MCA)记录信息。

19.如权利要求17所述的一个或多个有形非暂态计算机可读介质，其中所述逻辑还被配置为向操作系统通知所述设备的降级状态。

20.如权利要求11-16中任一项所述的一个或多个有形非暂态计算机可读介质，其中所述第二逻辑还包括配置接口，所述配置接口被配置为接收配置选项。

21.一种提供延迟错误处理的计算机实现的方法，包括：

提供系统管理中断(SMI)处置器；

提供第一容器和第二容器；

检测第一容器中的不可校正错误；

响应于检测，产生降级系统状态；

将降级状态消息提供给SMI处置器；

指令第二容器寻找可恢复状态；

确定第二容器已进入可恢复状态；以及

开始恢复操作。

22.如权利要求21所述的方法，还包括：设置超时以及在超时期满之后开始所述恢复操作。

23.如权利要求21所述的方法，还包括：经结构接口将降级通知提供给控制器。

24.如权利要求23所述的方法，还包括：请求所述控制器产生由第一容器提供的服务的新实例。

25.如权利要求21所述的方法，其中所述可恢复状态包括第二容器可以在最小数据丢失的情况下被迁移的状态。

延迟错误处理

技术领域

[0001] 本公开一般地涉及云计算的领域，并且更具体地但并不排它地，涉及一种用于延迟错误处理的系统和方法。

背景技术

[0002] 现代计算实践已放弃硬件专用计算并转向“网络即装置”。现代网络可包括数据中心，数据中心主控大量通用硬件服务器装置，所述大量通用硬件服务器装置被包含在例如服务器机架中并且由管理程序控制。每个硬件装置可运行虚拟装置(诸如工作负载服务器或虚拟桌面)的一个或多个实例。

附图说明

[0003] 当结合附图阅读时，根据下面的详细描述来最好地理解本公开。需要强调的是，根据行业中的标准实践，各种特征未必按照比例绘制，并且仅用于说明目的。在明确地或隐含地示出比例的情况下，它仅提供一个说明性示例。在其它实施例中，为了讨论的清楚，各种特征的尺寸可被任意增加或减小。

[0004] 图1是根据本说明书的一个或多个示例的云服务提供商(CSP)的网络级图。

[0005] 图2是根据本说明书的一个或多个示例的数据中心的方框图。

[0006] 图3图示根据本说明书的一个或多个示例的中央处理单元的方框图。

[0007] 图4是根据本说明书的一个或多个示例的数据中心计算架构的方框图。

[0008] 图5是图示根据本说明书的一个或多个示例的不可校正错误的恢复如何影响多个容器的方框图。

[0009] 图6a-6b是根据本说明书的一个或多个示例的执行延迟错误处理的方法的信号流程图。

具体实施方式

[0010] 下面的公开提供用于实现本公开的不同特征的许多不同实施例或示例。以下描述部件和布置的特定示例以简化本公开。这些当然仅是示例，并且不意图是限制性的。另外，本公开可在各种示例中重复参考数字和/或字母。这种重复是为了简单和清楚的目的，并且本质上并非指示讨论的各种实施例和/或配置之间的关系。不同实施例可具有不同优点，并且并不必然要求任何实施例具有特定优点。

[0011] 在现代数据中心中，可实现非常高的计算密度。例如，一批高性能计算平台可被聚集成刀片底盘(blade chassis)或计算滑车(compute sled)，并且该底盘可随后消耗机架底盘中的一个或多个槽。具有这种类型的几个高密度计算节点的机架因此可在具有例如42U或类似容量的单个机架中主控数十或数百个核心。

[0012] 软件工程技术可将采用这种架构的每个核心作为目标以在多线程过程中运行单个线程。单个应用可具有多个线程，并且因此，可消耗多个处理器核心。一个或多个另外的

核心还可专用于提供操作系统和/或其它支持软件。

[0013] 在一些情况下，为了节省为每个分立应用提供单独操作系统的开销，在也保持应用之间的某种逻辑分离的同时，单个操作系统可运行许多“容器”。这些容器可共享低级操作系统资源，但另一方面可彼此隔离。

[0014] 这种架构支持提供计算、存储、通信和加速资源，这些资源能够在经结构连接的数据中心中提供。

[0015] 如上所述的容器的优点在于：它们为基础设施即服务(IaaS)、平台即服务(PaaS)和软件即服务(SaaS)提供模块化并且灵活的机架规模实现方式。

[0016] 这种多容器化系统中的一个挑战在于：一个容器中的单个不可校正错误可能引起分级结构的故障，并且在一些情况下，可使底层操作系统发生故障，因此引起跨其它容器的数据丢失。在较低优先级容器可能遇到错误并且可能因此引起较高优先级容器发生故障的情况下，这可能尤其具有挑战性。在一些示例中，情况可能恶化，因为较低优先级容器可能具有较不强健的编程模型，而较高优先级容器可能更加“坚固”并且更加强健。因此，较不强健的较低优先级容器可能引起与更加强健的较高优先级容器的非预期交互。

[0017] 例如，一个实现方式可包括提供电子邮件服务器的一个容器和提供高可用性数据库服务器的另一容器。如果低优先级电子邮件服务器遇到不可校正错误(诸如，损坏的存储器位置)，则操作系统错误处置例程可能需要完全重新启动以确保存储器完整性。不幸的是，这种完全重新启动不仅将会影响低优先级电子邮件服务器，而且将会影响高优先级高可用性数据库服务器。另外，如果在数据库服务器正在执行关键操作(诸如，数据库写操作)的同时发生故障，则该故障可能事实上导致数据库本身中的一个或多个记录的损坏。

[0018] 尽管可通过在单个操作系统上仅提供同质容器(诸如，在单个操作系统中仅为数据库服务器提供其它数据库服务器)来部分地避免这个问题，但这个策略可能影响容器化计算的优点。另外，这种错误不能被完全避免，因为即使非常强健的应用也可能遇到错误。因此，如果多个容器各自正在运行非常强健的数据库服务器，如果那些数据库服务器之一遇到不可校正错误，则它将会使所有数据库服务器发生故障。

[0019] 这种不可校正错误的示例包括存储器子系统RAS堆栈(硬件、固件和/或软件)中的错误。从这种不可校正错误恢复的尝试可包括增强MCA产生到基础固件模型。

[0020] 因此，提供能够从不可校正错误更加从容地恢复的系统是有益的。特别地，并非当在单个容器中运行的应用遇到不可校正错误时立即前往错误处置器例程，而是替代地能够实现延迟错误处置。这种延迟错误处置是可行的，因为虽然容器可能共享底层操作系统服务，但它们通常不共享存储器页或其它资源。因此，对于一个容器而言存储器页可能损坏或无法访问的事实不应该影响另一容器。因此，替代于立即前往可能使其它容器发生故障并且引起数据丢失的灾难性错误恢复，能够实现延迟错误处置。利用延迟错误处置，可以通知其它容器寻找“可恢复状态”或者另一方面为错误处置做准备。如遍及本说明书所使用的，“可恢复状态”是节点的工作负载被完成、最小化或减少并且因此，数据丢失或数据损坏的危险也被消除、最小化或减少的状态。与容器的活动状态相比，可恢复状态可以是相对静止的状态。例如，如果容器是数据库驱动器，则可恢复状态可以是这样的状态：在该状态中它不再接受新的到来的数据库连接，并且所有未决操作已被完成和提交。在web服务器的情况下，可恢复状态可以是这样的状态：在该状态中它不再接受到来的HTTP连接，并且所有未决

事务已被合理地处置。在计算节点(例如,对于大型并行计算而言)的情况下,可恢复状态可以是这样的状态:在该状态中它不再接受到来的计算事务,并且已完成和输出已有事务。在又一示例中,可恢复状态可包括这样的状态,其中容器能够被迁移到具有最小数据丢失的新硬件平台,在这种情况下,容器可在错误恢复发生之前被迁移。

[0021] 尽管错误恢复被延期,但遇到错误的容器本身可被停止。因为它已遇到不可校正错误,所以它可能无法继续进行计算或处理。然而,在采用软件定义的联网和网络功能虚拟化的柔性数据中心中,经常可行的是,产生该服务的新实例用于处置由于发生故障的容器中的一个实例的丢失而丢失的任何另外的工作负载。

[0022] 需要注意的是,如这里所公开的,“寻找”可恢复状态不必要求“降级”(但未停止)的容器立即停止接受到来的事务。为了避免数据中心服务的崩溃,降级的容器可继续接受新的到来的连接或事务,同时数据中心处于更高负载下,并且可等待,直至负载逐渐减少以停止接受新的到来连接或事务。但是因为停止的容器正在消耗在别处不能被分配的资源,所以错误处理可能不被无限地延期。寻找可恢复状态的指令可包括超时。如果容器未在超时期满之前达到它的最佳可恢复状态,则错误处置可被以任何方式处理,使得由停止的节点消耗的资源可被带回到数据中心上的循环中。

[0023] 针对不可校正错误的最终响应可取决于系统能力。例如,该响应可以包括关机和重新启动或错误恢复。在错误恢复被禁用或不可用的一些情况下,该响应可通常是计算资源的关机,后面是错误收割(error harvesting),然后重新启动。尽管这种错误恢复从操作系统的角度看起来是透明的,但它事实上可能对当前正在执行的容器具有严重的性能影响。这是因为,为了恢复,几个硬件子单元可能必须被重新编程并且恢复,并且事务在一些情况下重新启动。这意味着:错误前状态可能丢失,并且可能必须通过软件重新执行而被重建。另外,有时高级恢复例程非常复杂并且可能需要完成几个系统管理例程。例如,镜像故障转移可能需要所有系统地址映射被更新以将发生故障的存储器反映为主存储器,这可能花费大量时间。

[0024] 在又一示例中,使用存储器备用,其中引擎将主记录中的所有以前保存的数据复制到从记录,然后将从记录标记为主记录。再次,无论系统是否已恢复,这可能需要完成几个迭代SMI流。在任何情况下,无论操作系统将错误视为不可恢复还是可恢复,其它任务(诸如,app或容器)可看见数据丢失或性能损失或二者。

[0025] 因此,下面的情况是有益的:提供延迟错误恢复机制,所述延迟错误恢复机制等待尝试错误恢复,直至其它app或容器处于更合适的(可恢复的)状态,以使得它们将不会丢失数据。

[0026] 在实施例中,系统可具有界面,该界面用于诸如从用户或协调器接收关于错误恢复选项的指令。例如,系统管理员可操作针对协调器的用户界面,她在协调器中定义错误恢复的策略,包括要采取的错误恢复动作、针对某些类型的容器的最大超时、针对容器可如何在接收到降级通知时寻找可恢复状态的参数以及协调器可如何帮助,诸如,通过引导业务离开降级的容器和/或产生由停止的容器提供的服务的新实例(自主地或根据来自主控停止的容器的硬件平台的请求)。

[0027] 本说明书的实施例支持从关键错误(诸如,存储器错误和高速缓存错误)恢复。作为非限制性示例,支持的错误类型可包括纠错码(ECC)存储器、PCIe数据错误和“有毒”数据

的其它示例。在一些实施例中，可区分可延迟错误和不应该被延迟的其它错误(诸如，缓冲器上的奇偶校验错误)。因此，本说明书的某些实施例识别这种错误，并且立即恢复它们，而不考虑延迟错误机制。

[0028] 错误抑制也可以是本说明书的实施例的考虑因素。可能希望确保：在数据错误的情况下，其它数据不被损坏。因此，可使用数据隔离。这可包括：阻止从损坏的存储器I/O(诸如，硬盘驱动器或其它外部数据源)写回。

[0029] 在延迟的错误恢复未决的同时，其它app和容器可从容地继续操作，直至它们达到良好的“停止点”。这可包括：允许其它容器完成程序执行，或者在合理时间期间继续程序执行。一些实施例可通知调度器不安排更多任务，直至错误被处置。因此，不仅容器被允许保持运行，而且可采取措施确保另外的工作负载从它去除，使得它可以达到可恢复状态。

[0030] 在某些实施例中，对于容器运行比系统的预期时间长的情况，还可指定超时。换句话说，可请求不受影响的容器开始减少工作负载并且达到可恢复状态。这可由操作系统或固件发出。一旦该请求被发出，操作系统或固件就可设置超时，在该超时之后，如果容器尚未通知它已达到从容的恢复状态，则它可被强制关闭或关机。这可类似于维护模式关机请求。因此，尽管应用被给予达到可恢复状态的机会，但它未被给予无限时间来这样做，因为那样可能影响最终错误恢复发生的能力。

[0031] 操作系统也被给予暂停违规应用的能力，使得不可使用来自该应用的进一步数据。因此，操作系统调度器有能力确保不再存在违规应用的调用。违规应用的状态可保持可用于稍后的调试和参考。

[0032] 在某些实施例中，系统代理固件还可通知系统管理员或其它管理任务：系统正在降级模式中运行。这可以采用例如这样的形式：向数据中心协调器发出通知。这可确保：数据中心协调器开始引导流量离开违规平台，从而使得它可以达到可恢复状态。例如，协调器可指令负载平衡器不将降级的容器分派给任何新的负载平衡桶，并且还可指令负载平衡器逐渐地将桶从降级的容器重新引导至其它容器。这可允许降级的容器逐渐地停机，从而使得它可以达到可恢复状态。

[0033] 有益地，当遇到与存储器和其它数据路径和子系统相关的另一容器中的关键错误时，延迟错误处理允许其它容器避免数据丢失或数据损坏。实施例还可创建默认app或容器的架构转储，所述架构转储可以随后在恢复模式或软件中被重放以允许可能的故障分析和稍后的调试。

[0034] 现在将更具体地参照附图描述用于延迟错误处理的系统和方法。应该注意的是，遍及各附图，某些参考数字可被重复以指示：特定装置或块在各附图间完全或基本上一致。然而，这并不意图暗示公开的各种实施例之间的任何特定关系。在某些示例中，一类元件可由特定参考数字表示(“小插件10”)，而所述类的个体物种或示例可由带有连字符号的数字表示(“第一特定小插件10-1”和“第二特定小插件10-2”)。

[0035] 图1是根据本说明书的一个或多个示例的云服务提供商(CSP) 102的网络100的网络级图。作为非限制性示例，CSP 102可以是传统企业数据中心、企业“私有云”或“公共云”，提供诸如基础设施即服务(IaaS)、平台即服务(PaaS)或软件即服务(SaaS)之类的服务。

[0036] CSP 102可提供某个数量的工作负载群集118，工作负载群集118可以是个体服务器、刀片服务器、机架安装服务器或任何其它合适的服务器拓扑的群集。在这个说明性示例

中，两个工作负载群集118-1和118-2被示出，每个工作负载群集在底盘148中提供机架安装服务器146。

[0037] 每个服务器146可主控独立操作系统，并且提供服务器功能，或者服务器可被虚拟化，在这种情况下，它们可处于虚拟机管理器(VMM)、管理程序和/或协调器的控制下，并且可主控一个或多个虚拟机、虚拟服务器或虚拟器具。这些服务器机架可共同位于单个数据中心中，或者可位于不同地理数据中心。根据契约协议，一些服务器146可具体专用于某些企业客户或租户，而其它服务器146可被共享。

[0038] 数据中心中的各种装置可经交换结构170彼此连接，交换结构170可包括一个或多个高速路由和/或交换装置。交换结构170可既提供“北-南”业务(例如，去往和来自广域网(WAN) (诸如，互联网)的业务)，又提供“东-西”业务(例如，横跨数据中心的业务)。在历史上，北-南业务占网络业务的大部分，但随着web服务变得更加复杂和分布，东-西业务量已上升。在许多数据中心中，东-西业务现在占业务的大多数。

[0039] 另外，当每个服务器146的能力增加时，业务量可进一步增加。例如，每个服务器146可提供多个处理器槽，每个槽容纳具有四至八个核心的处理器以及用于所述核心的足够的存储器。因此，每个服务器可主控许多VM，每个VM产生它自己的业务。

[0040] 为了适应数据中心中的大量业务，可提供高性能交换结构170。交换结构170在这个示例中被图示出为“扁平”网络，其中每个服务器146可具有与架顶式(ToR)交换机120的直接连接(例如，“星形”配置)，并且每个ToR交换机120可耦合到核心交换机130。仅作为说明性示例示出这种两层扁平网络架构。在其它示例中，可使用其它架构，作为非限制性示例，诸如基于“Clos”架构的三层星形或叶刺(也被称为“胖树”拓扑)、轴辐式拓扑、网状拓扑、环形拓扑或3-D网状拓扑。

[0041] 可通过任何合适的互连来提供所述结构本身。例如，每个服务器146可包括结构接口，诸如Intel® 主机结构接口(HFI)、网络接口卡(NIC)或其它主机接口。主机接口本身可经互连件或总线(诸如，PCI、PCIe或类似物)耦合到一个或多个处理器，并且在一些情况下，这种互连总线可被视为结构170的一部分。

[0042] 互连技术可由单个互连件或混合互连件提供，在这种混合互连件中PCIe提供片上通信，1Gb或10Gb铜以太网提供相对较短的与ToR交换机120的连接，并且光缆提供相对较长的与核心交换机130的连接。作为非限制性示例，互连技术包括Intel® OmniPath™、TrueScale™、超级路径互连(UPI)(以前称为QPI或KTI)、STL、光纤通道、以太网、以太网光纤通道(FCoE)、InfiniBand、PCI、PCIe或光纤等等。这些互连技术中的一些互连技术将会比其它互连技术更适合某些部署或功能，并且为即时应用选择合适的结构是普通技术人员的工作。

[0043] 然而，需要注意的是，尽管这里作为说明提供高端结构(诸如，OmniPath™)，但更一般地讲，结构170可以是用于特定应用的任何合适的互连件或总线。在一些情况下，这可以包括传统互连件，比如局域网(LAN)、令牌环网络、同步光网络(SONET)、异步传送模式(ATM)网络、无线网络(诸如，WiFi和蓝牙)、“普通老式电话系统”(POTS)互连件等。还明确地预期：在未来，新的网络技术将会出现以补充或替换这里列出的那些网络技术中的一些网络技术，并且任何这种未来网络拓扑和技术可以是结构170的一部分或形成结构170的一部分。

[0044] 在某些实施例中，如OSI七层网络模型中最初所概述的，结构170可在各个“层”上提供通信服务。在当代实践中，OSI模型未被严格地遵循。一般而言，层1和2经常被称为“以太网”层(但在大型数据中心中，以太网已经常被更新的技术取代)。层3和4经常被称为传输控制协议/互联网协议(TCP/IP)层(其可被进一步细分为TCP和IP层)。层5-7可被称为“应用层”。这些层定义作为有用的框架被公开，但意图是非限制性的。

[0045] 图2是根据本说明书的一个或多个示例的数据中心200的方框图。在各种实施例中，数据中心200可以是与图1的数据中心100相同的数据中心，或者可以是不同数据中心。另外的视图被提供在图2中以图示数据中心200的不同方面。

[0046] 在这个示例中，提供结构270以将数据中心200的各个方面互连。结构270可与图1的结构170相同，或者可以是不同结构。如上所述，可通过任何合适的互连技术来提供结构270。在这个示例中，Intel® OmniPath™被用作说明性且非限制性示例。

[0047] 如所图示的，数据中心200包括形成多个节点的许多逻辑元件。应该理解，每个节点可由物理服务器、一组服务器或其它硬件提供。每个服务器可正在运行适合其应用的一个或多个虚拟机。

[0048] 节点0 208是包括处理器插座0和处理器插座1的处理节点。处理器可以是例如具有多个核心(诸如，4或8个核心)的Intel® Xeon™处理器。节点0 208可被配置为诸如通过主控多个虚拟机或虚拟器具来提供网络或工作负载功能。

[0049] 处理器插座0和处理器插座1之间的板上通信可由板上上行链路278提供。这可在两个处理器插座之间提供超高速、短长度互连，使得在节点0 208上运行的虚拟机可以以超高速彼此通信。为了促进这种通信，虚拟交换机(vSwitch)可被提供在节点0 208上，虚拟交换机(vSwitch)可被视为结构270的一部分。

[0050] 节点0 208经结构接口272连接到结构270。结构接口272可以是如上所述的任何合适的结构接口，并且在这个特定说明性示例中，可以是用于连接到Intel® OmniPath™结构的Intel®主机结构接口。在一些示例中，诸如通过经由OmniPath™ 提供UPI隧道效应，可为与结构270的通信开辟通道。

[0051] 因为数据中心200可按照分布式方式提供在以前各代中在板上提供的许多功能，所以可提供高性能结构接口272。结构接口272可按照每秒几千兆比特的速度进行操作，并且在一些情况下，可与节点0 208紧密耦合。例如，在一些实施例中，用于结构接口272的逻辑直接与片上系统上的处理器集成在一起。这在结构接口272和处理器插座之间提供超高速通信，而不需要中间总线装置，所述中间总线装置可能将另外的延时引入到所述结构中。然而，这并不暗示在传统总线上提供结构接口272的实施例将被排除。相反地，明确地预期：在一些示例中，结构接口272可被提供在诸如PCIe总线之类的总线上，PCIe总线是PCI的串行化版本，提供比传统PCI更高的速度。遍及数据中心200，各个节点可提供不同类型的结构接口272，诸如板上结构接口和插入式结构接口。还应该注意的是，片上系统中的某些块可被提供作为知识产权(IP)块，所述IP块可以被“投入”到集成电路中作为模块化单元。因此，可在一些情况下从这种IP块得出结构接口272。

[0052] 要注意的是，以“网络即装置”方式，节点0 208可提供有限的板上存储器或存储装置，或者不提供板上存储器或存储装置。相反地，节点0 208可主要依赖于分布式服务，诸如存储器服务器和联网存储服务器。在板上，节点0 208可仅提供足够的存储器和存储装置以

引导所述装置并且使其与结构270通信。因为现代数据中心的超高速，这种分布式架构是可能的，并且可以是有益的，因为不需要为每个节点过度提供资源。相反地，可在许多节点之间动态地提供大量高速或专用存储器，使得每个节点可访问大量资源，但当该特定节点不需要那些资源时，那些资源不会空闲。

[0053] 在这个示例中，节点1存储器服务器204和节点2存储服务器210提供节点0 208的操作存储器和存储能力。例如，存储器服务器节点1 204可提供远程直接存储器访问(RDMA)，由此节点0 208可按照DMA方式经结构270访问节点1 204上的存储器资源，类似于它将会访问它自己的板上存储器的方式。由存储器服务器204提供的存储器可以是传统存储器，诸如易失性双倍数据速率类型3(DDR3)动态随机存取存储器(DRAM)，或者可以是更独特类型的存储器，诸如永久快速存储器(PFM)，比如Intel® 3D Crosspoint™ (3DXP)，其以如DRAM一样的速度操作但是非易失性的。

[0054] 类似地，替代于为节点0 208提供板上硬盘，可提供存储服务器节点2 210。存储服务器210可提供联网磁盘簇(NBOD)、PFM、独立磁盘冗余阵列(RAID)、独立节点冗余阵列(RAIN)、网络附接存储器(NAS)、光学存储装置、磁带驱动器或其它非易失性存储器解决方案。

[0055] 因此，在执行其指定功能时，节点0 208可访问来自存储器服务器204的存储器，并且将结果存储在由存储服务器210提供的存储装置上。这些装置中的每个装置经结构接口272耦合到结构270，结构接口272提供快速通信，所述快速通信使这些技术成为可能。

[0056] 作为另外的说明，还描绘了节点3 206。节点3 206也包括结构接口272以及由上行链路在内部连接的两个处理器插座。然而，与节点0 208不同，节点3 206包括它自己的板上存储器222和存储装置250。因此，节点3 206可被配置为主要在板上执行它的功能，并且可以不需要依赖于存储器服务器204和存储服务器210。然而，在合适的情况下，类似于节点0 208，节点3 206可利用分布式资源补充它自己的板上存储器222和存储装置250。

[0057] 这里公开的各种部件的基本构件可被称为“逻辑元件”。逻辑元件可包括硬件(包括例如软件可编程处理器、ASIC或FPGA)、外部硬件(数字、模拟或混合信号)、软件、交互软件、服务、驱动器、接口、部件、模块、算法、传感器、部件、固件、微码、可编程逻辑或可以协作以实现逻辑操作的对象。另外，一些逻辑元件由有形非暂态计算机可读介质提供，所述有形非暂态计算机可读介质具有存储在其上的用于指令处理器执行某个任务的可执行指令。作为非限制性示例，这种非暂态介质可以包括例如硬盘、固态存储器或盘、只读存储器(ROM)、永久快速存储器(PFM)(例如、Intel® 3D Crosspoint™)、外部存储装置、独立磁盘冗余阵列(RAID)、独立节点冗余阵列(RAIN)、网络附接存储装置(NAS)、光学存储装置、磁带驱动器、备用系统、云存储装置或前述各项的任何组合。这种介质还可以包括编写到FPGA中或编码在ASIC或处理器上的硬件中的指令。

[0058] 图3图示根据某些实施例的中央处理单元(CPU) 312的方框图。虽然CPU 312描绘特定配置，但CPU 312的核心和其它部件可被按照任何合适的方式布置。CPU 312可包括任何处理器或处理装置，诸如微处理器、嵌入式处理器、数字信号处理器(DSP)、网络处理器、应用处理器、协处理器、片上系统(SOC)或用于执行代码的其它装置。在描绘的实施例中，CPU 312包括四个处理元件(描述的实施例中的核心330)，所述四个处理元件可包括不对称处理元件或对称处理元件。然而，CPU 312可包括任何数量的处理元件，所述任何数量的处

理元件可以是对称的或不对称的。

[0059] 硬件处理元件的示例包括:线程单元、线程槽、线程、过程单元、情境、情境单元、逻辑处理器、硬件线程、核心和/或能够保存处理器的状态(诸如,执行状态或架构状态)的任何其它元件。换句话说,在一个实施例中,处理元件表示能够独立地与代码(诸如,软件线程、操作系统、应用或其它代码)关联的任何硬件。物理处理器(或处理器插座)通常表示集成电路,所述集成电路可能包括任何数量的其它处理元件,诸如核心或硬件线程。

[0060] 核心可表示位于集成电路上的能够保持独立架构状态的逻辑,其中每个独立保持的架构状态与至少一些专用执行资源关联。硬件线程可表示位于集成电路上的能够保持独立架构状态的任何逻辑,其中独立保持的架构状态共享对执行资源的访问。物理CPU可包括任何合适数量的核心。在各种实施例中,核心可包括一个或多个无序处理器核心或者一个或多个有序处理器核心。然而,可从任何类型的核心(诸如原生核心、软件管理核心、适应于执行原生指令集架构(ISA)的核心、适应于执行转换的ISA的核心、合作设计的核心或其它已知核心)个别地选择各个核心。在异质核心环境(即,不对称核心)中,某种形式的转换(诸如,二进制转换)可被用于安排或执行在一个或两个核心上的代码。

[0061] 在描绘的实施例中,核心330A包括无序处理器,无序处理器具有前端单元370,前端单元370用于取出到来的指令,执行各种处理(例如,高速缓存、解码、分支预测等)并且将指令/操作向前传递给无序(000)引擎。000引擎对解码的指令执行进一步处理。

[0062] 前端370可包括解码模块,解码模块耦合到取出逻辑以对取出的元素进行解码。在一个实施例中,取出逻辑包括与核心330的线程槽关联的个体定序器。通常,核心330与第一ISA关联,第一ISA定义/指定可在核心330上执行的指令。作为第一ISA的一部分的机器代码指令经常包括参考/指定待执行指令或操作的指令的一部分(被称为操作码)。解码模块可包括这样的电路:该电路从其操作码识别这些指令,并且在流水线中传递解码的指令以用于如第一ISA所定义的处理。在一个实施例中,核心330的解码器识别相同的ISA (或其子集)。替代地,在异质核心环境中,一个或多个核心(例如,核心330B)的解码器可识别第二ISA (第一ISA的子集或不同的ISA)。

[0063] 在描绘的实施例中,无序引擎包括分配单元382用于从前端单元370接收解码的指令(所述解码的指令可具有一个或多个微指令或*u*ops的形式),并且将它们分配给合适的资源(诸如,寄存器等)。接下来,指令被提供给保留站384,保留站384保留资源并且安排它们在多个执行单元386A-386N之一上执行。可存在各种类型的执行单元,包括例如算术逻辑单元(ALU)、加载和存储单元、向量处理单元(VPU)、浮点执行单元等。来自这些不同执行单元的结果被提供给重新排序缓冲器(ROB) 388,ROB 388获取无序结果并且使它们恢复到正确程序次序。

[0064] 在描绘的实施例中,前端单元370和无序引擎380都被耦合到存储器分级体系的不同级。具体地示出的是指令级高速缓存372,指令级高速缓存372继而耦合到中级高速缓存376,中级高速缓存376继而耦合到末级高速缓存395。在一个实施例中,末级高速缓存395被实现在片上(有时被称为非核心)单元390中。非核心390可与系统存储器399通信,在图示的实施例中,经嵌入式DRAM (eDRAM)实现系统存储器399。000引擎380内的各种执行单元686与第一级高速缓存374通信,第一级高速缓存374也与中级高速缓存376通信。另外的核心330B-330D也可耦合到末级高速缓存395。

[0065] 在特定实施例中，非核心390可位于与核心的电压域和/或频域分开的电压域和/或频域中。也就是说，非核心390可由与用于为核心供电的供给电压不同的供给电压供电，和/或可在与核心的操作频率不同的频率操作。

[0066] CPU 312还可包括功率控制单元(PCU) 340。在各种实施例中，PCU 340可控制施加于每个核心(在每个核心基础上)和施加于非核心的供给电压和操作频率。PCU 340还可在不执行工作负载时指令核心或非核心进入空闲状态(其中不供给电压和时钟)。

[0067] 在各种实施例中，PCU 340可检测硬件资源(诸如，核心和非核心)的一个或多个应力特性。应力特性可包括施加于硬件资源上的应力的量的指示。作为示例，应力特性可以是施加于硬件资源的电压或频率；在硬件资源处感测到的功率水平、电流水平或电压水平；在硬件资源处感测到的温度；或其它合适的测量值。在各种实施例中，当在特定时刻感测应力特性时，可执行特定应力特性的多次测量(例如，在不同位置)。在各种实施例中，PCU 340可按照任何合适的间隔检测应力特性。

[0068] 在各种实施例中，PCU 340是与核心330分立的部件。在特定实施例中，PCU 340按照与由核心630使用的时钟频率不同的时钟频率运行。在PCU是微控制器的一些实施例中，PCU 340根据与核心330所使用的ISA不同的ISA执行指令。

[0069] 在各种实施例中，CPU 312还可包括非易失性存储器350用于存储与核心330或非核心390关联的应力信息(诸如，应力特性、递增应力值、积累的应力值、应力积累速率或其它应力信息)，使得当失去电力时保持应力信息。

[0070] 图4是根据本说明书的一个或多个示例的数据中心计算架构的方框图。架构400图示数据中心中的各个部件的相关性，以使得错误处理可以影响超过一个容器。

[0071] 在这个示例中，刀片底盘404包括刀片404-1至404-n。刀片底盘404可被实现为多u模块408，多u模块408包括计算模块408-1至408-n。计算模块408可被装入抽屉412内。抽屉412可安装到机架416的一个或多个槽中。并且机架416可以是可替换数据中心外壳420的一部分。

[0072] 刀片404-1至404-n可提供多个资源，诸如处理器424、存储器428、结构接口432和存储容器436。结构接口432可将计算节点耦合到结构模块440。结构模块440可提供交换机端口448并且连接到VLAN 444。VLAN 444可包括许多VLAN端口452。

[0073] 存储容器436可包括iSCSI目标438和一个或多个逻辑驱动器456，每个逻辑驱动器456在物理驱动器460上被主控。

[0074] 如以上所讨论的，在操作系统实例上运行的单个计算节点中的各种容器可彼此隔离这些资源中的许多资源。

[0075] 然而，这种系统中的错误恢复范例可基于机器检查架构。在这种情况下，在错误消耗时，机器检查异常被立即处置。对于可恢复错误，相应的任务由操作系统终止，并且OS可随后将存储器与另外的使用隔离以避免未来错误。在不可恢复错误的情况下，利用冷重新启动来使整个机器停机以恢复标称系统操作。

[0076] 在这种不可恢复错误的情况下，当使整个机器停机时，可应用固件优先模型以减轻一些问题，但固件优先未必解决数据丢失。另外，当任务被终止或者机器被重新启动时，可能存在最小硬件错误收割。这可提供对应用级故障分析的最小洞察。例如，命名为Windows Hardware Error Architecture (WHEA)_uncorrectable_error的Microsoft错误

检查0x124不提供除WHEA记录之外的信息。

[0077] 相比之下，利用延迟错误处置，对于可恢复和不可恢复错误情况，都可立即隔离故障存储器区域。通过为操作人员提供规划服务的选项，这优化操作费用。通过改进任务调度控制(诸如，机器开发和操作(DevOp))，它还提高可管理性。另外，它可以提供存储器类型或制造的预测故障分析。

[0078] 关于数据丢失，这里描述的系统去除受到错误影响的容器，并且保存机器检查情境以用于未来服务。这允许剩余容器继续执行而不立即中断，因此减少数据丢失并且增加系统可用时间和可用性。关于故障分析，除了错误记录之外，目前的解决方案也可将任务专有记录或情境保存在存储器中。这允许另外的后处理或事后故障分析。

[0079] 图5是图示根据本说明书的一个或多个示例的不可校正错误的恢复如何影响多个容器的方框图。在图5的示例中，刀片系统502主控计算平台503，计算平台503为数据中心过程提供硬件。在刀片系统502上运行的是操作系统512，操作系统512主控多个任务或应用508，并且还提供两个容器，即容器504-1和容器504-2。

[0080] 硬件平台503可包括各种部件用于为系统的软件部件提供硬件服务。这可包括多个核心522-1至522-6。核心522可访问一个或多个DRAM模块，诸如DRAM 516-1和DRAM 516-2。存储器控制器526-1和526-2可为DRAM模块516提供硬件控制。还可提供一级或多级高速缓存530。输入输出控制器(IOCTL) 534可提供输入和输出操作。系统代理538可提供例如用于检测错误和从错误恢复以及其它系统服务的固件。系统代理538可与一个或多个I/O模块546对接，如在I/O模块546-2的情况下那样直接对接，或者经平台控制器集线器(PCH) 542对接。

[0081] 应该理解，刀片系统502仅被提供作为可提供计算服务的硬件平台503的非限制性的并且说明性的示例。许多其它配置是可能的，并且本说明书并不意图局限于刀片系统502的示例或任何其它特定硬件平台。

[0082] 例如，当容器504-1正在访问DRAM 516-1内的存储器的块时，错误可能发生。存储器的所述块可被具体地分割并且专用于容器504-1，并且因此可能无法被容器504-2访问。在执行它的操作并且访问DRAM 516-1的同时，容器504-1可能遇到不可校正错误。这可能是例如导致不可校正软件错误的错误编程的结果，或者它可能是DRAM 516-1内的硬件故障(诸如，坏的或受损存储块)的结果。

[0083] 继而，容器504-2可能正在访问完全分开的存储器块，该存储器块可以位于DRAM 516-1上，或者可以位于完全分开的DRAM(诸如，DRAM 516-2)中。因为容器504-1和容器504-2具有分开地分割的存储器块，所以容器504-1遇到的存储器错误不直接影响容器504-2。然而，因为容器504-1上的错误是不可校正错误，所以可能无法在不重新启动刀片系统502情况下恢复容器504-1的功能。当刀片系统502被重新启动时，可以检查存储器，并且如果该错误是硬件错误的结果，则在一些情况下，坏存储器块可以被从循环去除，使得它不可寻址，并且刀片系统502可以随后继续正常地工作。

[0084] 然而，如果刀片系统502在容器504-1遇到不可校正错误时立即被系统代理538停机，则容器504-2也被立即停机。因此，虽然容器504-1所遇到的存储器错误不直接影响容器504-2，但恢复容器504-1所需的冷重新启动确实影响容器504-2。另外，在没有对容器504-2的任何警告的情况下直接冷重新启动可导致容器504-2丢失数据和/或损坏数据。

[0085] 如果容器504-2正在提供比容器504-1高优先级的功能，则这也意味着：作为相对较低优先级功能的错误的结果，高优先级功能被终止或停止。如以上所讨论，容器504-1可以是电子邮件服务器，而容器504-2可以是高可用性数据库服务器。在这种情况下，在没有对容器504-2的警告的情况下使刀片系统502停机可能在容器504-2中引起数据丢失或数据库损坏。

[0086] 因此，在一些实施例中，下面的情况是有益的：替代于立即重新启动刀片系统502，向容器504-2通知冷重新启动将会是必要的，并且然后在使刀片系统502停机之前等待容器504-2到达良好的“停止地点”。如这里所述，这个通知指令容器504-2寻找可恢复状态。要注意的是，作为非限制性示例，提供针对容器504-2的通知。在其它实施例中，系统代理538可简单地监视容器504-2，并且自主地确定容器504-2何时已达到可恢复状态。还要注意的是，其它网络元件(诸如，协调器或控制器)可诸如通过指令负载平衡器将更少的业务引导至容器504-2来帮助容器504-2寻找可恢复状态。

[0087] 一旦刀片系统502达到如下状态：所有容器处于针对重新启动的合适状态，那么系统代理538就可执行它的正常错误处置，直至并且包括刀片系统502的冷重新启动。同时，容器504-1可被停止并且不可用，但通过简单地产生由容器504-1提供的功能的新实例，容器504-1的丢失经常可以在数据中心(尤其是提供软件定义的联网和协调的功能的数据中心)中被改善。在一些情况下，如果容器504-2提供非常不能容忍中断的功能，则在刀片系统502被重新启动之前，可产生容器504-2的功能的新实例，并且可在所述两个实例之间完成移交，使得重新启动相对无缝。

[0088] 图6A-6B是根据本说明书的一个或多个示例的执行延迟错误处理的方法的信号流程图。图6A-6B的示例图示操作系统608、可提供系统代理的系统固件606、CPU硬件604和CPU微码602之间的交互。作为实施例的说明，这里表示的逻辑被划分成分开的块。然而，应该理解，这里示出的划分仅是说明性的，并且在一个块中提供的功能在许多实例中可以被移动至另一个块。特别地，CPU中的微码602和硬件604之间的划分经常是设计优化的问题，并且在不同实施例中在一方或另一方之间移动功能经常是可行的。另外，由系统固件606提供的功能可经常被移动到微码602，或被提供在其它地方。

[0089] 在图示的示例中，在图6B中的操作系统608中在块662开始，存储器位置对存储器页执行存储器访问。这个访问从离页连接符流动至微码602，在微码602中在块610，数据收集器单元(DCU)接收数据访问请求并且确定这是“有毒”访问请求。换句话说，这个位置的存储器被损坏或者另一方面无法访问，并且存储器访问不能继续。

[0090] 在块614中，响应于接收到有毒数据已被接收的指示，微码可采取合适的行动，诸如触发微码中的“存储器损坏”事件。

[0091] 在块618中，响应于存储器损坏事件，微码602可发出错误信号。在一个示例中，错误信号是具有错误代码0x20或某种其它“无效”指示符的页错误。在另一示例中，页错误可具有空页错误代码，或者可使用某种特定指定的有毒代码。这个特定代码的目的可以是触发操作系统上的分段错误或用于根据需要再次重放容器的架构状态转储。这类似于在SMM或真实模式到受保护模式转变期间切换的情境。

[0092] 页错误可经离页连接符B被发送给操作系统608。在块664中，操作系统608可触发分段错误。核心转储(core dump)可发生，并且应用可被从执行去除。架构状态可被保存以

用于未来重放。有益地，利用核心转储，调试器可被用于分析程序流，以确定什么引起了存储器错误并且避免未来存储器错误。

[0093] 在块668中，在操作系统608上，硬件平台上的其它容器可继续执行。

[0094] 在块694中，可执行去往其它容器的线程的任务切换。

[0095] 返回到块618，在微码602内，降级状态被发出给系统代理。

[0096] 在块622中，微码602可针对固件606的系统代理触发“降级”状态。降级状态可被用于向平台本身上的资源和数据中心中的其它资源通知：这个硬件平台上的容器现在应该被视为“降级”。

[0097] 在一个示例中，利用新降级状态寄存器触发降级状态。降级状态寄存器的示例性实现方式可包括下面内容：

定时器[63:07]	计数器[6:3]	延迟处置完成[2:2]	EN/MCE/DHC[1:1]	MCE触发[0:0]
用于触发事件的偏移。	延迟状态可以被保留，直至计数器已达到最大值	固件已完成延迟错误处置：设置这个位触发下一个序列(例如，错误处置可继续进行)。	在延迟的处置完成时启用机器检查异常(MCE)	现在触发MCE (与调试情况相关)。

[0098] 在块626中，微码602可向系统代理通知降级状态。

[0099] 例如，在硬件604内，在块630，系统代理接收降级状态信号，并且可触发用于初始降级响应处置的SMI。SMI可经离页连接符C被发出给固件中的SMI处置器。

[0100] 在硬件604内，系统代理可随后启动计数器以用于下一SMI产生。这个计数器可被用于确保：没有其它容器花费太长时间达到可恢复状态以及不必要地延迟硬件平台的恢复。

[0101] 因此，在判定块638中，硬件604连续地监视超时以查看它是否已期满。只要超时未期满，系统代理就继续等待容器达到从容的离开点。

[0102] 如果定时器在其它容器未达到合适的离开点的情况下期满，则流程继续前进至离页连接符D。

[0103] 返回到图6B，从离页连接符C，系统固件606中的SMI处置器接收SMI，SMI向它通知降级状态。

[0104] 在块646中，SMI处置器评估所述系统是否处于降级状态。如果系统已被识别为处于降级状态，则它可向操作系统通知规划的关机或维护周期。操作系统可随后向其它容器通知：它们应该开始朝着可重置点工作。操作系统还可通知其它数据中心部件(诸如，数据中心协调器)，使得协调器可以采取措施以减少降级的容器上的工作负载。例如，如果电子邮件容器遇到错误并且数据库驱动器正在相同硬件平台上的分开的容器上运行，则当操作系统向协调器通知数据库驱动器正在降级状态下运行时，协调器可开始尽可能多地引导业务离开该数据库驱动器实例。例如，如果足够的资源可用，则协调器可以产生数据库驱动器的新实例。协调器还可以指令负载平衡器不将任何新的业务筒分派给数据库驱动器的降级实例。为了进一步减少数据库驱动器上的负载，协调器可以逐渐地指令负载平衡器将已有桶重新分派给其它实例。这可允许数据库驱动器的降级实例达到这样的状态：在该状态中它不接收任何新的到来业务。它可以随后从容地完成任何未决请求的处理，并且一旦它没有未决请求，它就处于它可以从容地关机的状态，并且可向系统代理通知该情况。

[0105] 在块650中，如果机器检查架构(MCA)或增强MCA (EMCA)被启用，则SMI处置器可收

集合适的日志;SMI处置器还可向例如架顶式交换机或远程监视和管理(RMM)通知降级系统状态。这可以是由操作系统608提供的通知的附加或替代。

[0106] 在判定块654中,系统固件606从硬件604接收硬超时限制。在循环中,SMI处置器可连续地检查其它容器是否已达到可恢复状态,诸如系统代理或系统固件是否已接收到其它容器不再处置新的到来业务的通知。替代地,可能发生硬超时。

[0107] 这些事件中的任一事件最终触发SMI处置器606设置延迟处置完成位。

[0108] 一旦延迟处置完成位被设置,控制就通过离页连接符E流回至图6A。在块642中,延迟处置完成位触发由SMI或MCE执行的最后错误处理。

[0109] 一旦最后错误处理已被触发,在块698中,就根据如这里所述的已有错误处置方法来处置该错误。

[0110] 前面内容概述了几个实施例的特征,使得本领域技术人员可更好地理解本公开的各个方面。本领域技术人员应该理解,他们可容易地使用本公开作为设计或修改用于执行相同目的和/或实现这里引入的实施例的相同优点的其它过程和结构的基础。本领域技术人员还应该意识到,这种等同构造不脱离本公开的精神和范围,并且他们可在不脱离本公开的精神和范围的情况下在这里做出各种改变、替换和变更。

[0111] 这里公开的任何硬件元件的全部或一部分可容易地被提供在包括中央处理单元(CPU)封装的片上系统(SoC)中。SoC代表将计算机或其它电子系统的部件集成到单个芯片中的集成电路(IC)。因此,例如,客户端装置或服务器装置可被整体地或部分地提供在SoC中。SoC可包含数字、模拟、混合信号和射频功能,所有这些功能可被提供在单个芯片基底上。其它实施例可包括多芯片模块(MCM),其中多个芯片位于单个电子封装内并且被配置为通过电子封装紧密地彼此交互。

[0112] 还要注意的是,在某些实施例中,一些部件可被省略或合并。在一般意义上,附图中描绘的布置可在它们的表示中是更加逻辑性的,而物理架构可包括这些元件的各种排列、组合和/或混合。必须注意的是,无数可能的设计配置可以被用于实现这里概述的操作目的。因此,关联的基础设施具有无数的替换布置、设计选择、装置可能性、硬件配置、软件实现方式和装备选项。

[0113] 在一般意义上,任何合适地配置的处理器可以执行与数据关联的任何类型的指令以实现这里详述的操作。这里公开的任何处理器可以将元件或物品(例如,数据)从一个状态或事物变换为另一状态或事物。在操作中,存储装置可将信息存储在任何合适类型的有形非暂态存储介质(例如,随机存取存储器(RAM)、只读存储器(ROM)、现场可编程门阵列(FPGA)、可擦除可编程只读存储器(EPROM)、电可擦除可编程ROM (EEPROM)等)、软件、硬件(例如,处理器指令或微码)中,或者在合适的情况下并且基于特定需要将信息存储在任何其它合适的部件、装置、元件或物体中。另外,基于特定需要和实现方式,在处理器中跟踪、发送、接收或存储的信息可以被提供在任何数据库、寄存器、表、高速缓存、队列、控制列表或存储结构中,可以在任何合适的时间帧中参考所有这些数据库、寄存器、表、高速缓存、队列、控制列表或存储结构。这里公开的任何存储器或存储元件应该视情况被解释为被包括在广义术语“存储器”和“存储装置”内。这里的非暂态存储介质明确地意图包括被配置为提供所公开的操作或使处理器执行所公开的操作的任何非暂态专用或可编程硬件。

[0114] 实现这里描述的全部或一部分功能的计算机程序逻辑被以各种形式实现,所述各

种形式包括但绝不限于源代码形式、计算机可执行形式、机器指令或微码、可编程硬件和各种中间形式(例如,由汇编器、编译器、链接器或定位器产生的形式)。在示例中,源代码包括一系列计算机程序指令,该一系列计算机程序指令以各种编程语言或以硬件描述语言实现,所述各种编程语言诸如是供各种操作系统或操作环境使用的目标代码、汇编语言或高级语言(诸如,OpenCL、FORTRAN、C、C++、JAVA或HTML),所述硬件描述语言诸如是Spice、Verilog和VHDL。源代码可定义并且使用各种数据结构和通信消息。源代码可具有计算机可执行形式(例如,经解释器),或者源代码可被转换(例如,经转换器、汇编器或编译器)成计算机可执行形式,或者转换成中间形式(诸如,字节码)。在合适的情况下,任何前述内容可被用于建立或描述合适的分立或集成电路,无论是顺序、组合、状态机还是其它。

[0115] 在一个示例性实施例中,附图的任何数量的电路可被实现在关联的电子装置的板上。所述板可以是一般电路板,该一般电路板可以容纳电子装置的内部电子系统的各种部件,并且进一步提供用于其它外围设备的连接器。基于特定配置需要、处理需求和计算设计,任何合适的处理器和存储器可以被合适地耦合到所述板。要注意的是,利用这里提供的许多示例,可根据两个、三个、四个或更多个电气部件描述交互。然而,仅为了清楚和示例的目的而进行这种描述。应该理解,该系统可以被以任何合适的方式合并或重新配置。根据类似的设计替代方案,附图的任何图示的部件、模块和元件可被按照各种可能的配置组合,全部配置落在本说明书的宽广范围内。

[0116] 本领域技术人员可确定许多其它改变、替换、变化、变更和修改,并且本公开意图包括落在所附权利要求的范围内的所有这种改变、替换、变化、变更和修改。为了帮助美国专利商标局(USPTO)并且另外帮助在本申请上公布的任何专利的任何阅读者解释这里所附权利要求,申请人希望指出,申请人:(a)不意图任何所附权利要求援引美国法典第35卷112节第六款(6) (pre-AIA)或同一节的 (f)款 (post-AIA),因为它在本文的申请日存在,除非在特定权利要求中明确地使用词语“用于...的装置”或“用于...的步骤”;和(b)不意图说明书中的任何语句以原本未在所附权利要求中明确地反映的任何方式限制本公开。

[0117] 示例性实现方式

作为说明提供下面的示例。

[0118] 示例1包括一种计算设备,包括:硬件平台,包括处理器和存储器;和系统管理中断(SMI)处置器;第一逻辑,被配置为经硬件平台提供第一容器和第二容器;和第二逻辑,被配置为:检测第一容器中的不可校正错误;响应于检测,产生降级系统状态;将降级状态消息提供给SMI处置器;指令第二容器寻找可恢复状态;确定第二容器已进入可恢复状态;以及开始恢复操作。

[0119] 示例2包括示例1的计算设备,其中所述第二逻辑还被配置为设置超时并且在超时期满之后开始所述恢复操作。

[0120] 示例3包括示例1的计算设备,还包括结构接口;其中所述第二逻辑还将降级通知提供给控制器。

[0121] 示例4包括示例3的计算设备,其中所述第二逻辑还请求控制器产生由第一容器提供的服务的新实例。

[0122] 示例5包括示例1的计算设备,其中所述可恢复状态包括第二容器可以在最小数据丢失的情况下被迁移的状态。

[0123] 示例6包括示例5的计算设备，其中所述第二逻辑还被配置为迁移第二容器。

[0124] 示例7包括示例1-6中任一项的计算设备，还包括操作系统，所述操作系统还被配置为执行第一容器的核心转储。

[0125] 示例8包括示例7的计算设备，其中所述操作系统被配置为从处理器接收机器检查架构(MCA)记录信息。

[0126] 示例9包括示例7的计算设备，其中所述第二逻辑还被配置为向操作系统通知所述设备的降级状态。

[0127] 示例10包括示例1-6中任一项的计算设备，其中所述第二逻辑还包括配置接口，所述配置接口被配置为接收配置选项。

[0128] 示例11包括一个或多个有形非暂态计算机可读介质，所述一个或多个有形非暂态计算机可读介质具有存储在其上的用于提供逻辑的指令，所述逻辑用于：提供系统管理中断(SMI)处置器；提供第一容器和第二容器；检测第一容器中的不可校正错误；响应于检测，产生降级系统状态；将降级状态消息提供给SMI处置器；指令第二容器寻找可恢复状态；确定第二容器已进入可恢复状态；以及开始恢复操作。

[0129] 示例12包括示例11的所述一个或多个有形非暂态计算机可读介质，其中所述逻辑还被配置为设置超时以及在超时期满之后开始所述恢复操作。

[0130] 示例13包括示例11的所述一个或多个有形非暂态计算机可读介质，其中所述逻辑还用于经结构接口将降级通知提供给控制器。

[0131] 示例14包括示例13的所述一个或多个有形非暂态计算机可读介质，其中所述逻辑还用于请求控制器产生由第一容器提供的服务的新实例。

[0132] 示例15包括示例11的所述一个或多个有形非暂态计算机可读介质，其中所述可恢复状态包括第二容器可以在最小数据丢失的情况下被迁移的状态。

[0133] 示例16包括示例15的所述一个或多个有形非暂态计算机可读介质，其中所述逻辑还被配置为迁移第二容器。

[0134] 示例17包括示例11-16中任一项的所述一个或多个有形非暂态计算机可读介质，其中所述逻辑还被配置为提供操作系统，所述操作系统被配置为执行第一容器的核心转储。

[0135] 示例18包括示例17的所述一个或多个有形非暂态计算机可读介质，其中所述操作系统被配置为从处理器接收机器检查架构(MCA)记录信息。

[0136] 示例19包括示例17的所述一个或多个有形非暂态计算机可读介质，其中所述逻辑还被配置为向操作系统通知所述设备的降级状态。

[0137] 示例20包括示例11-16中任一项的所述一个或多个有形非暂态计算机可读介质，其中所述第二逻辑还包括配置接口，所述配置接口被配置为接收配置选项。

[0138] 示例21包括一种提供延迟错误处理的计算机实现的方法，所述方法包括：提供系统管理中断(SMI)处置器；提供第一容器和第二容器；检测第一容器中的不可校正错误；响应于检测，产生降级系统状态；将降级状态消息提供给SMI处置器；指令第二容器寻找可恢复状态；确定第二容器已进入可恢复状态；以及开始恢复操作。

[0139] 示例22包括示例21的方法，还包括：设置超时以及在超时期满之后开始所述恢复操作。

[0140] 示例23包括示例21的方法，还包括：经结构接口将降级通知提供给控制器。

[0141] 示例24包括示例23的方法，还包括：请求控制器产生由第一容器提供的服务的新实例。

[0142] 示例25包括示例21的方法，其中所述可恢复状态包括第二容器可以在最小数据丢失的情况下被迁移的状态。

[0143] 示例26包括示例26的方法，还包括：迁移第二容器。

[0144] 示例27包括示例21-26中任一项的方法，还包括：提供操作系统，所述操作系统被配置为执行第一容器的核心转储。

[0145] 示例28包括示例27的方法，其中所述操作系统被配置为从处理器接收机器检查架构(MCA)记录信息。

[0146] 示例29包括示例27的方法，还包括：向操作系统通知所述设备的降级状态。

[0147] 示例30包括示例21-26中任一项的方法，还包括：提供配置接口，所述配置接口被配置为接收配置选项。

[0148] 示例31包括一种设备，所述设备包括用于执行示例21-30中任一项的方法的装置。

[0149] 示例32包括示例31的设备，其中用于执行所述方法的所述装置包括处理器和存储器。

[0150] 示例33包括示例32的设备，其中所述存储器包括机器可读指令，当被执行时，所述机器可读指令使所述设备执行示例21-30中任一项的方法。

[0151] 示例34包括示例31-33中任一项的设备，其中所述设备是计算系统。

[0152] 示例35包括至少一个有形非暂态计算机可读介质，所述至少一个有形非暂态计算机可读介质包括指令，当被执行时，所述指令实现如示例21-34中任一项所示的方法或设备。

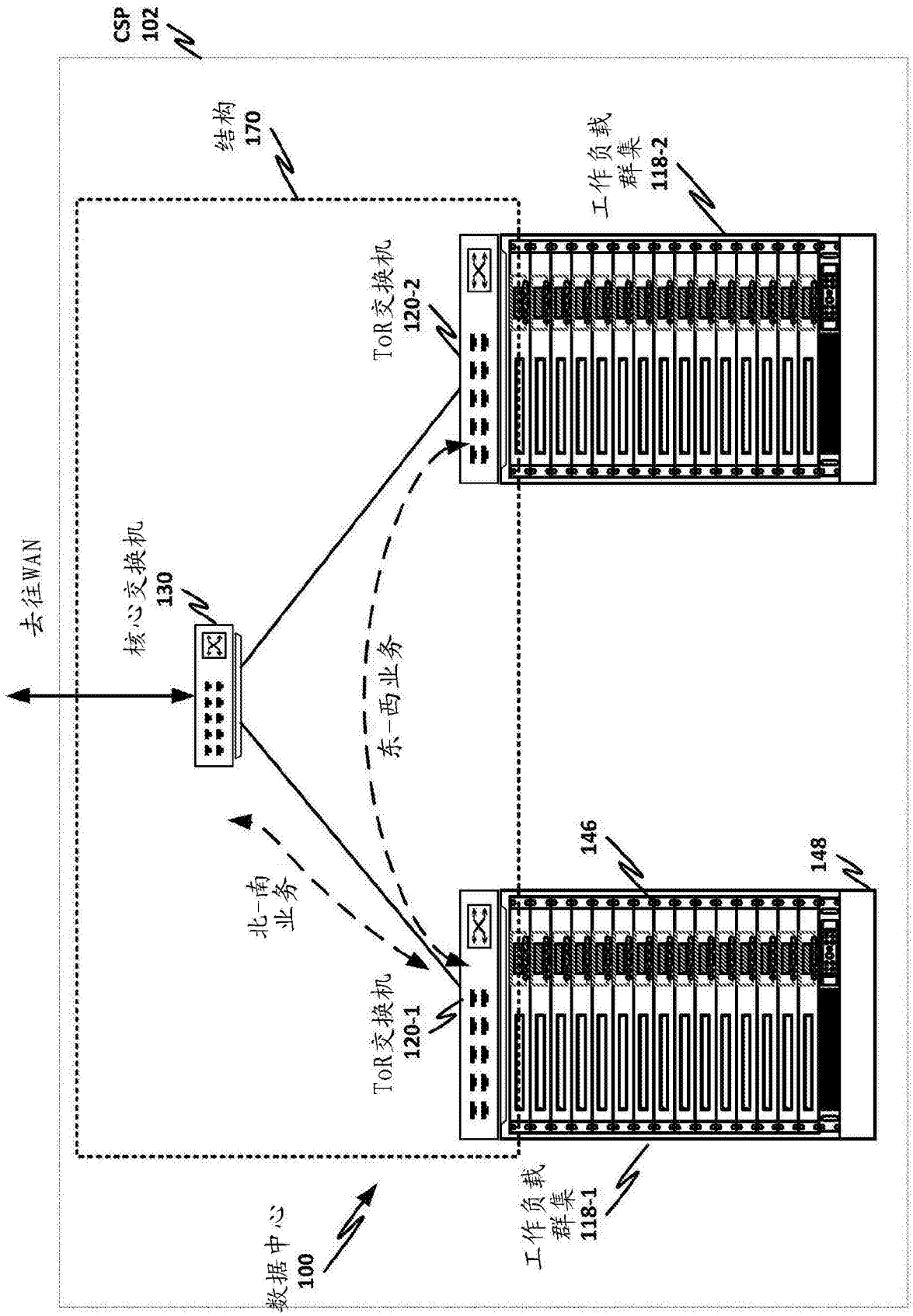

CSP
102
结构
170
工作负载
群集
118-2
ToR交换机
120-2
去往WAN
核心交换机
130
东-西业务
北-南
业务
146
148
ToR交换机
120-1
工作负载
群集
118-1
数据中心
100

图 1

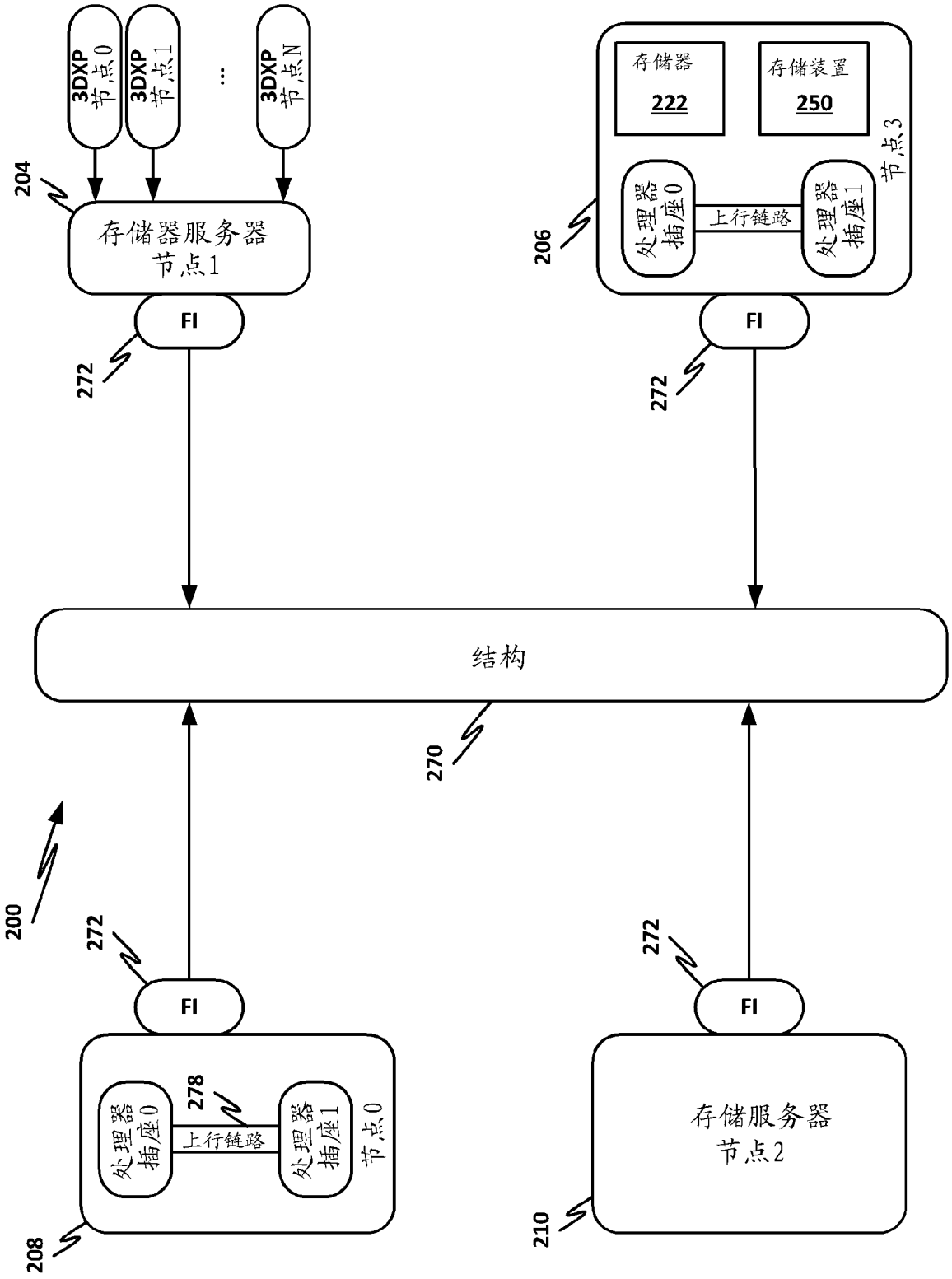

3DXP 节点0
3DXP 节点1
...
3DXP 节点N
204
存储器服务器 节点1
FI
272
存储器
222
存储装置
250
处理器插座0
上行链路
处理器插座1
节点3
206
FI
272
结构
270
200
272
FI
处理器插座0
278
上行链路
处理器插座1
节点0
208
272
FI
存储服务器 节点2
210

图 2

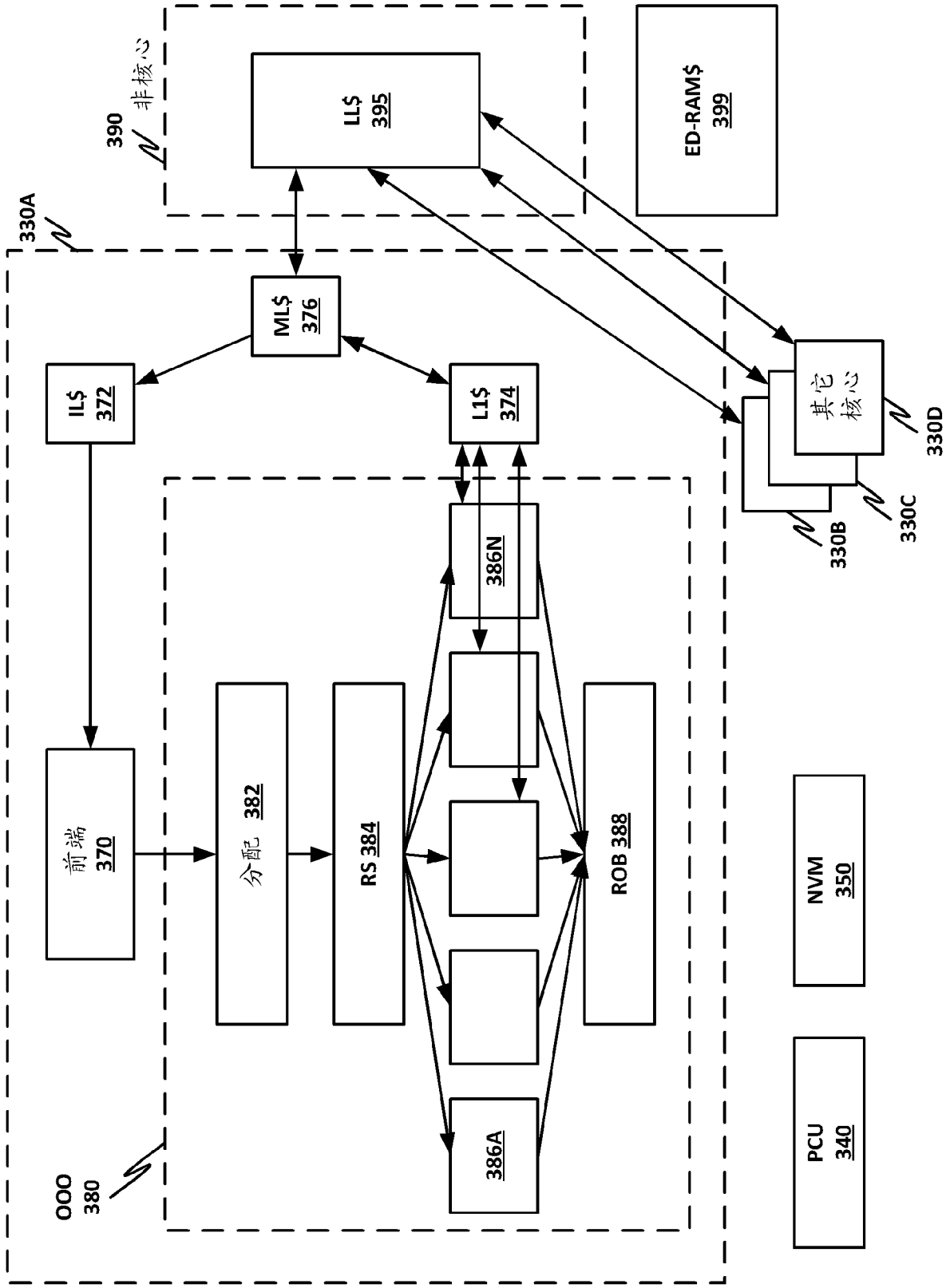

390
非核心
LL$
395
ED-RAM$
399
330A
ML$
376
IL$
372
L1$
374
其它
核心
330D
330C
330B
386N
前端
370
分配 382
RS 384
ROB 388
NVM
350
PCU
340
386A
OOO
380

图 3

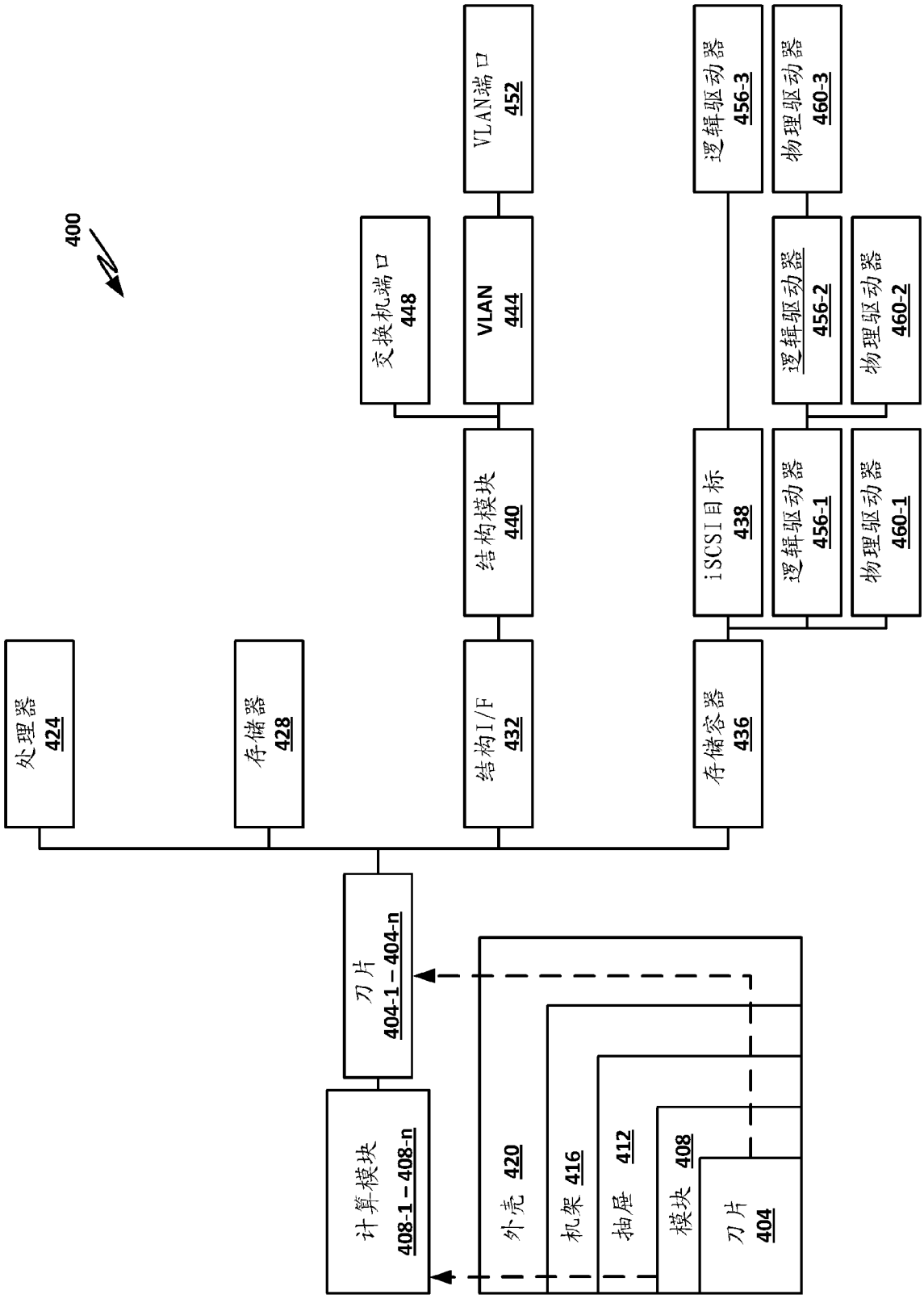

400
处理器 424
存储器 428
结构I/F 432
存储容器 436
结构模块 440
VLAN 444
交换机端口 448
VLAN端口 452
iSCSI目标 438
逻辑驱动器 456-1
逻辑驱动器 456-2
逻辑驱动器 456-3
物理驱动器 460-1
物理驱动器 460-2
物理驱动器 460-3
刀片 404-1 – 404-n
计算模块 408-1 – 408-n
外壳 420
机架 416
抽屉 412
模块 408
刀片 404

图 4

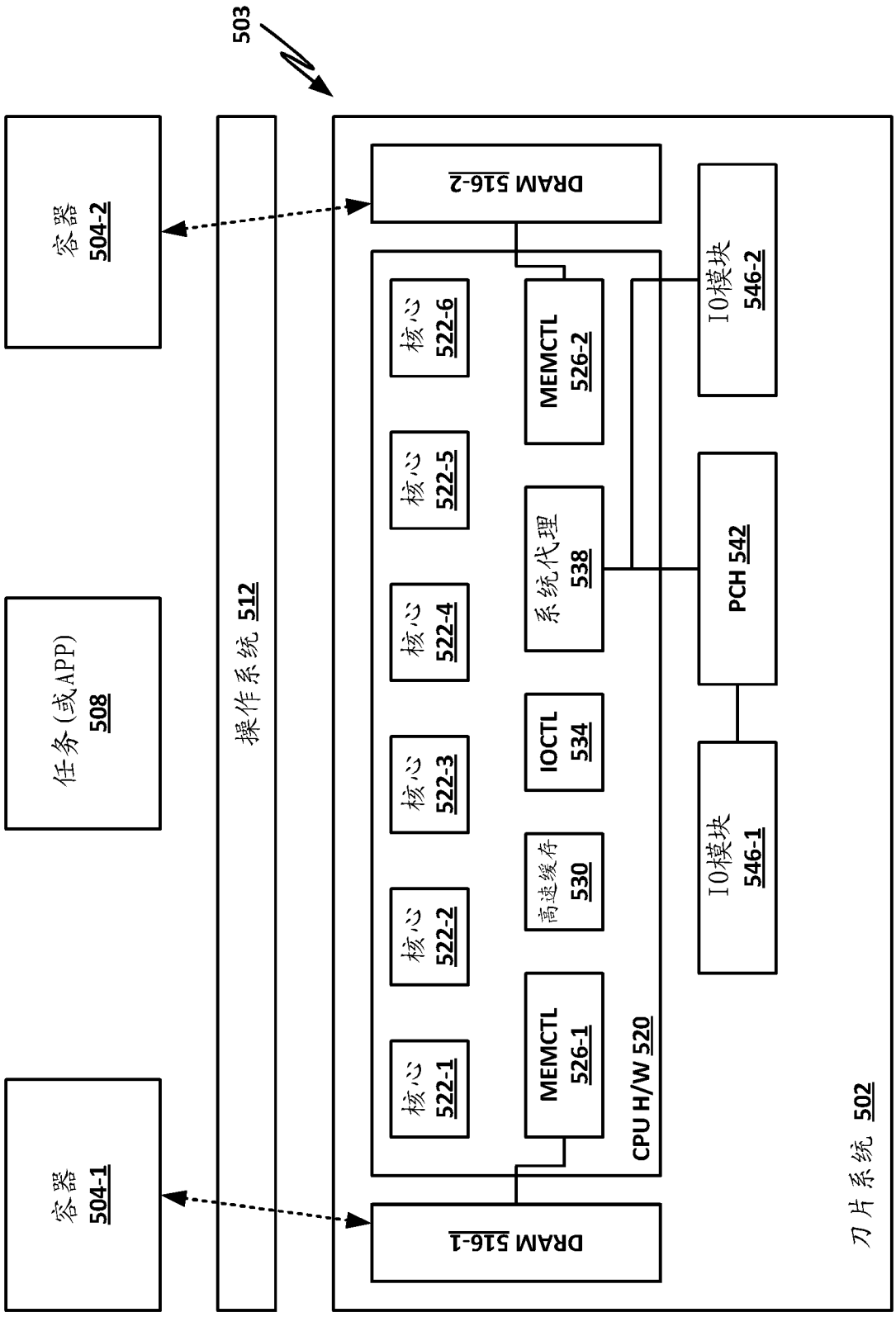

503
容器 504-2
任务(或APP) 508
容器 504-1
操作系统 512
DRAM 516-2
DRAM 516-1
核心 522-6
核心 522-5
核心 522-4
核心 522-3
核心 522-2
核心 522-1
MEMCTL 526-2
MEMCTL 526-1
系统代理 538
高速缓存 530
IOCTL 534
CPU H/W 520
PCH 542
IO模块 546-2
IO模块 546-1
刀片系统 502

图 5

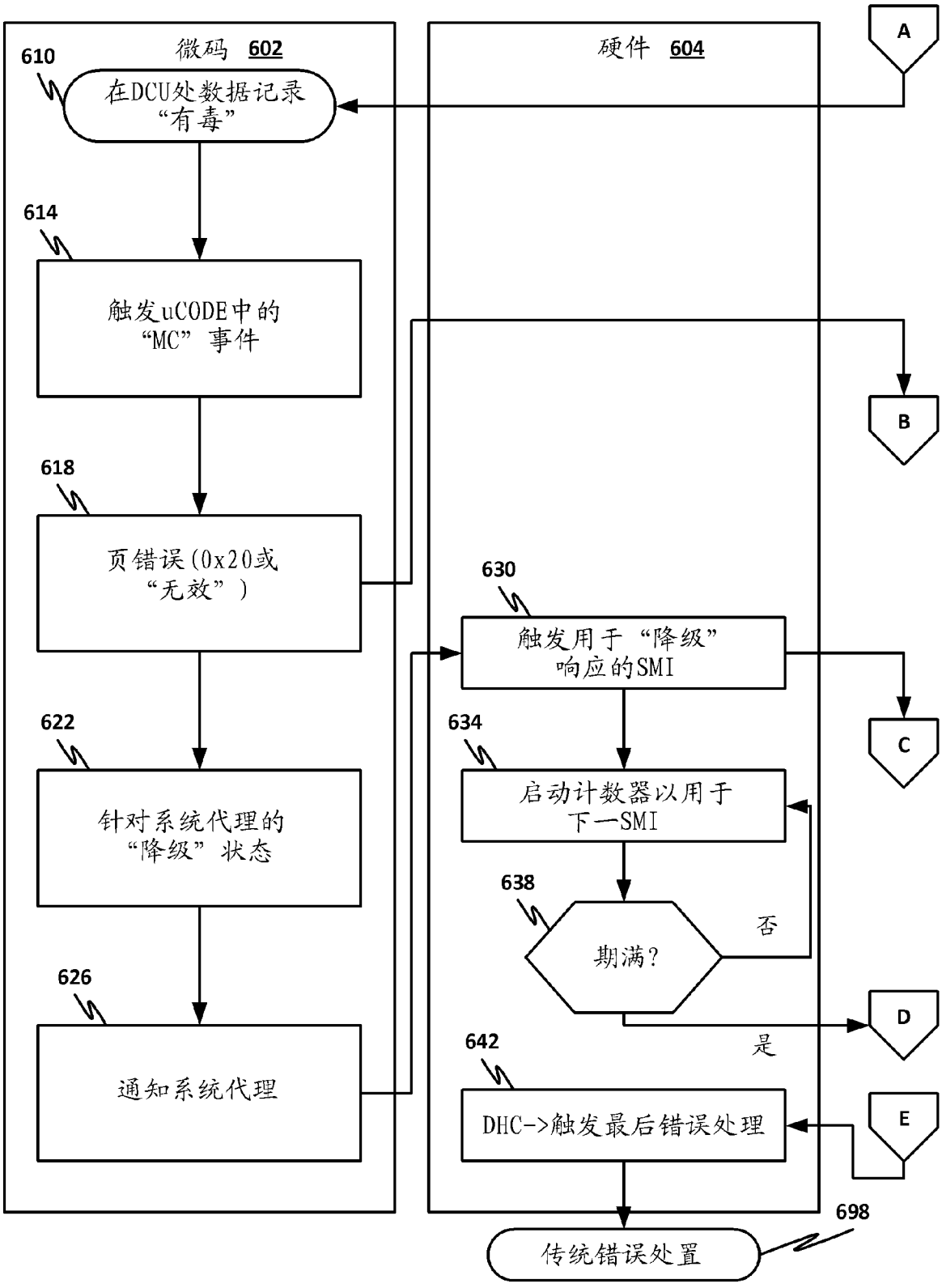

微码 602
硬件 604
610
在DCU处数据记录“有毒”
614
触发uCODE中的“MC”事件
618
页错误(0x20或“无效”)
622
针对系统代理的“降级”状态
626
通知系统代理
630
触发用于“降级”响应的SMI
634
启动计数器以用于下一SMI
638
期满?
否
是
642
DHC->触发最后错误处理
698
传统错误处置
A
B
C
D
E

图 6a

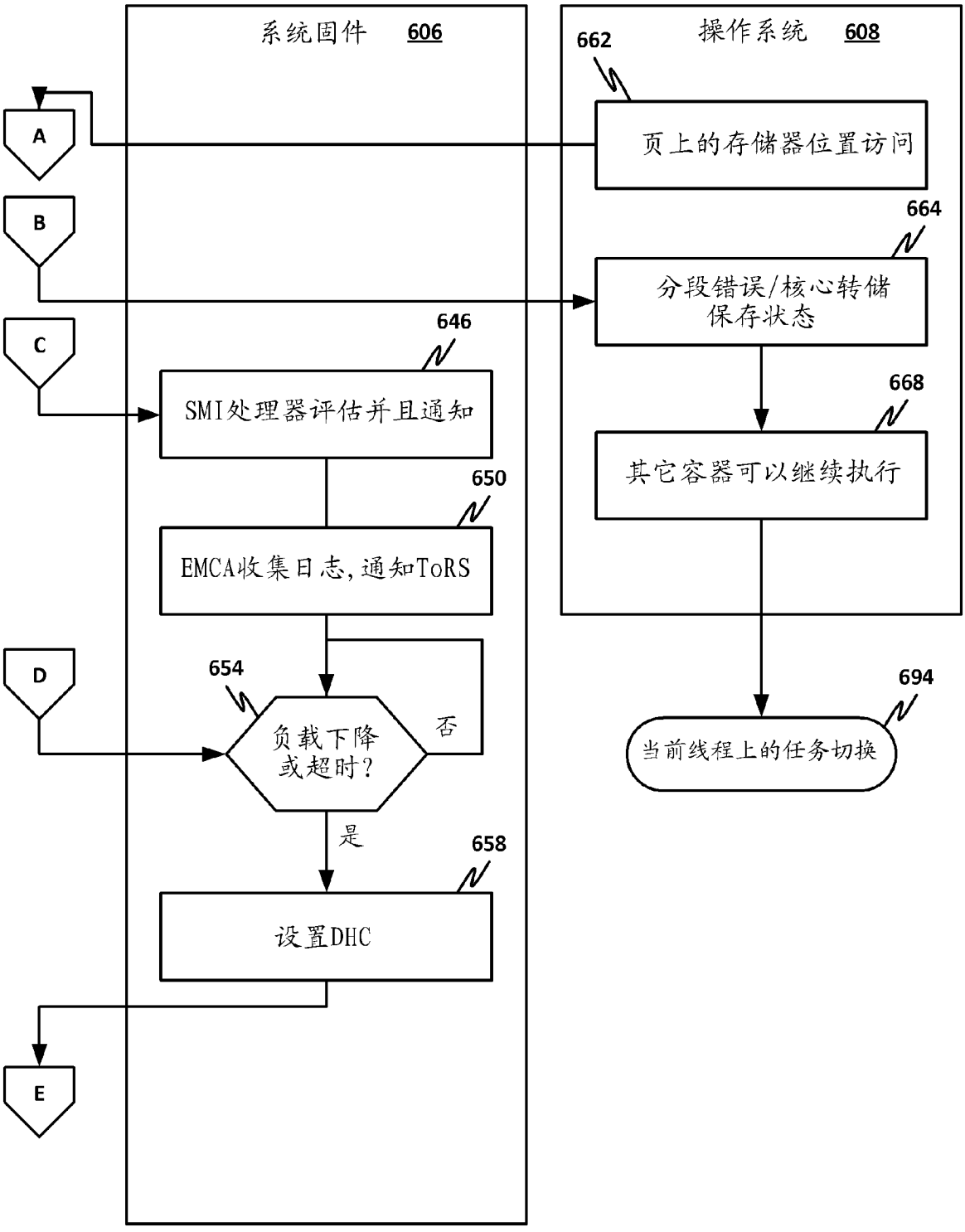

系统固件 606
操作系统 608
662
页上的存储器位置访问
664
分段错误/核心转储
保存状态
668
其它容器可以继续执行
694
当前线程上的任务切换
A
B
C
D
E
646
SMI处理器评估并且通知
650
EMCA收集日志,通知ToRS
654
负载下降
或超时?
否
是
658
设置DHC

图 6b