(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 103488505 A
(43)申请公布日 2014.01.01

(21)申请号 201310421697.5

(22)申请日 2013.09.16

(71)申请人 杭州华为数字技术有限公司
地址 310052 浙江省杭州市滨江区滨兴路 301 号 3 幢 A 楼 301 室

(72)发明人 王工艺 李涛 李生

(74)专利代理机构 深圳市威世博知识产权代理事务所（普通合伙） 44280
代理人 何青瓦

(51)Int.Cl.
G06F 9/445(2006.01)

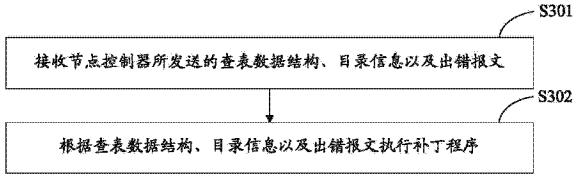
权利要求书2页 说明书9页 附图2页

(54)发明名称

补丁方法、设备及系统

(57)摘要

本发明公开了一种补丁方法、设备及系统。所述方法包括：如果节点控制器内的协议处理引擎执行 Cache 一致性协议出错，挂起出错报文及继续处理与出错报文地址不同的后续报文，则增设的补丁接收节点控制器所发送的查表数据结构、目录信息以及出错报文；增设的补丁模块根据查表数据结构、目录信息以及出错报文执行补丁程序，其中，补丁程序用于代替协议处理引擎对出错报文进行处理以生成新的报文及生成相应的新的目录信息。通过增设补丁模块，在出错时，协议处理引擎模块挂起出错报文，并对继续对后续报文进行处理，而出错报文则通过补丁模块进行处理，从而保证在协议出错时，系统依然能够正常工作。



CN 103488505 A

1. 一种补丁方法，其特征在于，如果节点控制器内的协议处理引擎执行 Cache 一致性协议出错，挂起出错报文及继续处理与所述出错报文地址不同的后续报文，则增设的补丁接收节点控制器所发送的查表数据结构、目录信息以及所述出错报文；

所述增设的补丁模块根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息。

2. 根据权利要求 1 所述的方法，其特征在于，所述出错报文包括当前出错报文及与当前出错报文具有相同地址的报文。

3. 根据权利要求 1 所述的方法，其特征在于，所述增设的补丁模块根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序的步骤之前进一步包括：所述增设的补丁模块接收节点控制器所发送的查表数据结构、目录信息、所述出错报文以及需要查询的协议表的标识。

4. 一种节点控制器，其特征在于，包括补丁模块，所述补丁模块耦接节点控制器内的协议处理引擎模块，所述补丁模块包括：接收单元以及执行单元，

所述接收单元用于在节点控制器内的协议处理引擎模块执行 Cache 一致性协议出错，挂起出错报文及继续处理与所述出错报文地址不同的后续报文时，接收节点控制器所发送的查表数据结构、目录信息以及所述出错报文，所述接收单元将所述查表数据结构、所述目录信息以及所述出错报文向所述执行单元发送；

所述执行单元用于接收所述查表数据结构、所述目录信息以及所述出错报文，根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎模块对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息。

5. 根据权利 4 所述的节点控制器，其特征在于，所述出错报文包括当前出错报文及与当前出错报文具有相同地址的报文。

6. 根据权利 4 所述的节点控制器，其特征在于，所述接收单元还用于接收节点控制器所发送的查表数据结构、目录信息、所述出错报文以及需要查询的协议表的标识。

7. 一种补丁系统，其特征在于，包括补丁模块以及节点控制器，所述补丁模块耦接节点控制器内的协议处理引擎，

所述节点控制器用于在协议处理引擎执行 Cache 一致性协议出错时，挂起出错报文及继续处理与所述出错报文地址不同的后续报文；

所述补丁模块用于接收节点控制器所发送的查表数据结构、目录信息以及所述出错报文，根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息。

8. 根据权利 7 所述的系统，其特征在于，所述出错报文包括当前出错报文及与当前出错报文具有相同地址的报文。

9. 根据权利 7 所述的系统，其特征在于，所述补丁模块还用于接收节点控制器所发送的查表数据结构、目录信息、所述出错报文以及需要查询的协议表的标识。

10. 根据权利 7 所述的系统，其特征在于，所述补丁模块包括输入随机存取存储器、补

丁处理器以及输出随机存取存储器，其中，所述输入随机存取存储器一端耦接节点控制器内的协议处理引擎的一端，另一端耦接所述补丁处理器的一端，所述补丁处理器的另一端耦接所述输出随机存取存储器的一端，所述输出随机存取存储器的另一端耦接节点控制器内的协议处理引擎的另一端，

所述输入随机存取存储器用于存储节点控制器所发送的查表数据结构、目录信息以及所述出错报文；

所述补丁处理器用于所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息；

所述输出随机存取存储器用于存储新的报文及新的目录信息。

11. 根据权利 10 所述的系统，其特征在于，所述输入随机存取存储器以及所述输出随机存取存储器的数目都与所述协议处理引擎的数量相同，其中，一个所述输入随机存取存储器的一端分别耦接一个协议处理引擎的一端，所有输入随机存取存储器的另一端同时耦接补丁处理器，所述补丁处理器的另一端同时耦接所有输出随机存取存储器的一端，每个输出随机存取存储器的另一端分别耦接一个协议处理引擎模块的另一端。

补丁方法、设备及系统

技术领域

[0001] 本申请涉及存储领域，特别是涉及补丁方法、设备及系统。

背景技术

[0002] 随着各种终端的高速发展，各种各样的高级应用越来越丰富，对于中央处理器(Central Processing Unit，CPU)的运算速度的要求也是越来越高。但是，单个CPU的性能提高是有限的，因此，多处理器系统得到了高速的发展。如图1所示，Cache一致性协议非均匀存储器存取结构(cache coherence protocol Non-Uniform Memory Access，CC-NUMA)多处理器系统包括多个CPU110以及多个节点控制器(Node Controller，NC)120。其中，节点控制器120与CPU110之间拓扑连接，节点控制器120协调多个CPU110之间的工作。

[0003] 由于每个CPU110都至少有一个私有的高速缓存缓冲器(Cache)，而所有CPU共享一个主存储器。当不同的CPU110使用到主存储器的同一个数据时，这些CPU110会分别把这个数据放到自己私有的Cache中，并根据各自的需要对数据进行修改等操作，当其中一个CPU110修改了这个数据时，另一个CPU110并不知道已有CPU110对这个数据进行了修改，所述它还是按原来的数据进行处理，造成冲突，这就导致了Cache一致性问题。

[0004] 现有技术提供了一种Cache一致性协议，每个节点控制器120内至少设置一个协议处理引擎，当报文进入到协议处理引擎时，协议处理引擎根据报文的类型及目录信息所记载的目录状态查找到对应的协议表，其中，协议表中记载了当前报文的处理方式、对所述目录信息的操作方式及是否发生冲突等等。如果发生了冲突，则协议处理引擎会根据当前报文的处理方式及所述目录信息的操作方式进行处理，并获得新的报文以及更新的目录信息，从而避免出现Cache一致性问题。

[0005] 但是，系统规模的快速扩充、网络延时的不确定性和存储一致性模型的多样性等诸多因素，使Cache一致性协议异常复杂，协议的状态空间呈指数级增长，在现有技术条件下，Cache一致性协议并不能做到覆盖所有的情况，即所有的冲突都能解决，而一旦有冲突不能被解决，协议就会出现问题，导致协议处理引擎无法对当前报文进行处理，造成后面的报文也无法进行处理，从而导致系统无法正常工作。

发明内容

[0006] 本申请提供协议处理引擎容错处理方法、装置及系统，能够在协议出错时，保证系统正常工作。

[0007] 本申请第一方面提供一种补丁方法，如果节点控制器内的协议处理引擎执行Cache一致性协议出错，挂起出错报文及继续处理与所述出错报文地址不同的后续报文，则增设的补丁接收节点控制器所发送的查表数据结构、目录信息以及所述出错报文；所述增设的补丁模块根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息。

[0008] 结合第一方面，本申请第一方面的第一种可能的实施方式中，所述出错报文包括当前出错报文及与当前出错报文具有相同地址的报文。

[0009] 结合第一方面，本申请第一方面的第二种可能的实施方式中，所述增设的补丁模块根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序的步骤之前进一步包括：所述增设的补丁模块接收节点控制器所发送的查表数据结构、目录信息、所述出错报文以及需要查询的协议表的标识。

[0010] 本申请第二方面提供一种节点控制器，包括补丁模块，所述补丁模块耦接节点控制器内的协议处理引擎模块，所述补丁模块包括：接收单元、以及执行单元，所述接收单元用于在节点控制器内的协议处理引擎模块执行 Cache 一致性协议出错，挂起出错报文及继续处理与所述出错报文地址不同的后续报文时，接收节点控制器所发送的查表数据结构、目录信息以及所述出错报文，所述接收单元将所述查表数据结构、所述目录信息以及所述出错报文向所述执行单元发送；所述执行单元用于接收所述查表数据结构、所述目录信息以及所述出错报文，根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎模块对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息。

[0011] 结合第二方面，本申请第二方面的第一种可能的实施方式中，所述出错报文包括当前出错报文及与当前出错报文具有相同地址的报文。

[0012] 结合第二方面，本申请第二方面的第二种可能的实施方式中，所述接收单元还用于接收节点控制器所发送的查表数据结构、目录信息、所述出错报文以及需要查询的协议表的标识。

[0013] 本申请第三方面提供一种补丁系统，包括补丁模块以及节点控制器，所述补丁模块耦接节点控制器内的协议处理引擎模块，所述节点控制器用于在协议处理引擎执行 Cache 一致性协议出错时，挂起出错报文及继续处理与所述出错报文地址不同的后续报文；所述补丁模块用于接收节点控制器所发送的查表数据结构、目录信息以及所述出错报文，根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息。

[0014] 结合第三方面，本申请第三方面的第一种可能的实施方式中，所述出错报文包括当前出错报文及与当前出错报文具有相同地址的报文。

[0015] 结合第三方面，本申请第三方面的第二种可能的实施方式中，所述补丁模块还用于接收节点控制器所发送的查表数据结构、目录信息、所述出错报文以及需要查询的协议表的标识。

[0016] 结合第三方面，本申请第三方面的第三种可能的实施方式中，所述补丁模块包括输入随机存取存储器、补丁处理器以及输出随机存取存储器，其中，所述输入随机存取存储器一端耦接节点控制器内的协议处理引擎模块的一端，另一端耦接所述补丁处理器的一端，所述补丁处理器的另一端耦接所述输出随机存取存储器的一端，所述输出随机存取存储器的另一端耦接节点控制器内的协议处理引擎的另一端，所述输入随机存取存储器用于存储节点控制器所发送的查表数据结构、目录信息以及所述出错报文；所述补丁处理器用于所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序

用于代替所述协议处理引擎对所述出错报文进行处理以生成新的报文及生成相应的新的目录信息；所述输出随机存取存储器用于存储新的报文及新的目录信息。

[0017] 结合第三方面的第三种可能的实施方式，本申请第三方面的第四种可能的实施方式中，所述输入随机存取存储器以及所述输出随机存取存储器的数目都与所述协议处理引擎的数量相同，其中，一个所述输入随机存取存储器的一端分别耦接一个协议处理引擎的一端，所有输入随机存取存储器的另一端同时耦接补丁处理器，所述补丁处理器的另一端同时耦接所有输出随机存取存储器的一端，每个输出随机存取存储器的另一端分别耦接一个协议处理引擎模块的另一端。

[0018] 上述申请通过增设补丁模块，在协议处理引擎出错时，协议处理引擎挂起出错报文，并对继续对与出错报文地址不同的后续报文进行处理，而出错报文则通过补丁模块进行处理，从而保证在协议出错时，系统依然能够正常工作。

附图说明

[0019] 图 1 是现有技术多处理系统一实施方式的结构示意图；

[0020] 图 2 是本申请补丁系统一实施方式的结构示意图；

[0021] 图 3 是本申请补丁方法一实施方式的流程图；

[0022] 图 4 是本申请节点控制器一实施方式的结构示意图；

[0023] 图 5 是本申请节点控制器另一实施方式的结构示意图。

具体实施方式

[0024] 以下描述中，为了说明而不是为了限定，提出了诸如特定系统结构、接口、技术之类的具体细节，以便透彻理解本申请。然而，本领域的技术人员应当清楚，在没有这些具体细节的其它实施方式中也可以实现本申请。在其它情况中，省略对众所周知的装置、电路以及方法的详细说明，以免不必要的细节妨碍本申请的描述。

[0025] 参阅图 2，图 2 是本申请补丁系统一实施方式的结构示意图。本实施方式的补丁系统包括：补丁模块 210 以及节点控制器 220。补丁模块 210 以及节点控制器 220 共同对多个 CPU 进行调度。其中，补丁模块 210 包括多个输入随机存取存储器（input ram）211、一个补丁处理器 213 以及多个输出随机存取存储器（output ram）215。节点控制器 220 包括多个协议处理引擎 221 以及多个协议表（P-Table）存储器 223。而且，输入随机存取存储器 211、输出随机存取存储器 215、协议处理引擎 221 以及协议表存储器 223 的数量均相等。

[0026] 一个输入随机存取存储器 211 的一端耦接一个协议处理引擎 221 的一端，所有输入随机存取存储器 211 的另一端同时耦接补丁处理器 213，补丁处理器 213 的另一端同时耦接所有输出随机存取存储器 215 的一端，每个输出随机存取存储器 215 的另一端分别耦接一个协议处理引擎 221 的另一端。每个协议处理引擎 221 分别耦接一个协议表存储器 223。

[0027] 报文通过协议处理引擎 221 时，协议处理引擎 221 会根据报文的类型以及当前的目录状态读取协议表存储器 223 中的协议，其中，协议表存储器 223 中的协议包括报文的处理方式、对目录的操作方式、内部表项的操作方式以及协议是否出错等等。如果协议设计不完备，导致报文按照协议处理后，产生错误的结果，协议没法对报文进行处理，则协议出错，而没法处理的报文为出错报文。如果协议没有出错，根据协议对报文进行 Cache 一致性协

议处理，从而获得新的报文及生成新的目录，并将新的报文加入流水线中。

[0028] 当某个协议处理引擎 221 执行 Cache 一致性协议出错时，协议处理引擎模块 221 挂起出错报文以暂停处理报文，包括当前出错报文以及与当前出错报文地址相同的报文，由于每个报文都有相应的地址，所以很容易查找到上述报文，其中地址是指 CPU 的系统地址，并将查表数据结构、目录信息以及出错报文发送给与该协议处理引擎 221 对应的输入随机存取存储器 211。查表数据结构用于记录包括出错报文地址、是否处于冲突处理阶段、是否处于写操作阶段以及是否因为冲突而挂起出错报文在内的环境数据。目录信息包括指针和状态，所述指针用于指向处理器中的地址空间，所述状态用于指示指针所指向的处理器的地址空间的具体状态，所述具体状态包括独占、共享和无效。例如，在出错前，正在访问某个控制器的某个地址空间，则指针指向该地址空间，并且状态项记录了这个地址空间的具体状态。而查表数据结构则记录访问这个地址空间时的环境数据。

[0029] 在另一种实施方式中，查表数据结构包括多个字段定义，所述字段定义用于记录事务处理的状态，包括报文的地址、目标 ID 号、是否处于冲突处理阶段、是否处于写操作阶段、是否因为冲突而挂起报文。当需要运行补丁时，事务处理处于中间状态，补丁知道中间状态以实现正确完成报文处理。同时，补丁处理完毕后，也需要把中间状态输出，保存下来，以便下次同样事务处理时使用。当然，最终事务处理完毕后，数据结构是会被自动清除掉的。

[0030] 如果处理器有足够的信息处理报文，则不需继续查询协议表，如果处理器没有足够的信息处理报文，则需要继续查询协议表。如果还需要继续查询协议表，则可以将需要查询的协议表的标识连同查表数据结构、目录信息以及出错报文一起发送给与该协议处理引擎模块 221 对应的输入随机存取存储器 211。协议处理引擎 221 继续处理与所述出错报文地址不同的后续报文，从而避免堵塞。

[0031] 补丁处理器 213 检测出输入随机存取存储器 211 中有需要处理的出错报文时，从输入随机存取存储器 211 中读取查表数据结构、目录信息以及出错报文，并根据查表数据结构、目录信息以及出错报文执行补丁程序，其中，补丁程序用于代替协议处理引擎 221 对出错报文按照协议处理引擎原来的处理方式进行处理以生成新的报文及生成相应的新的目录信息。如果还需要继续查询协议表，则补丁处理器 213 根据需要查询的协议表的标识查询协议表，然后根据查表数据结构、目录信息以及出错报文执行补丁程序以生成新的报文及生成相应的新的目录信息。例如，补丁程序可根据查表数据结构恢复对上述地址空间的操作，又从目录信息中获得该地址空间的状态，所以，补丁程序可代替协议处理引擎 221 对出错报文按照协议处理引擎原来的处理方式进行处理，并生成新的报文，处理完毕后，该地址空间的状态可能发生改变，于是更新目录信息从而产生新的目录信息。

[0032] 补丁处理器 213 将新的报文及新的目录信息写入到与该协议处理引擎模块 221 对应的输出随机存取存储器 215，其中，每个协议处理引擎模块 221 与一个输出随机存取存储器 215 连接。该协议处理引擎 221 监测到对应的输出随机存取存储器 215 中有新的报文后，将新的报文插入到流水线后半段。其中，报文顺序排列构成了流水线。

[0033] 如果其它的协议处理引擎 221 在执行协议时出错了，同样按照上述的方法进行处理。

[0034] 在本实施方式中，每个协议处理引擎 221 分别对应设置一个输入随机存取存储器

211 以及一个输出随机存取存储器 215，并且，多个输入随机存取存储器 211 以及多个输出随机存取存储器 215 共用一个补丁处理器 213。在协议处理引擎 221 在执行协议时出错率比较低的情况下，也可以多个协议处理引擎 221 共用一个输入随机存取存储器 211、一个输出随机存取存储器 215 以及一个补丁处理器 213。相反地，如果在要求比较高的情况下，也可以一个协议处理引擎 221 使用一个输入随机存取存储器 211、一个输出随机存取存储器 215 以及一个补丁处理器 213。

[0035] 参阅图 3，图 3 是本申请补丁方法一实施方式的流程图。本实施方式的补丁方法增包括如下步骤：

[0036] S301：如果节点控制器内的协议处理引擎执行 Cache 一致性协议出错，挂起出错报文及继续处理与出错报文地址不同的后续报文，则增设的补丁接收节点控制器所发送的查表数据结构、目录信息以及出错报文。

[0037] 当节点控制器内的某个协议处理引擎模块执行 Cache 一致性协议出错时，协议处理引擎模块挂起出错报文以暂停处理报文，包括当前出错报文以及与当前出错报文地址相同的报文，由于每个报文都有相应的地址，所以很容易查找到上述报文，其中地址是指 CPU 的系统地址，并将查表数据结构、目录信息以及出错报文发送给补丁。查表数据结构用于记录包括出错报文地址、是否处于冲突处理阶段、是否处于写操作阶段以及是否因为冲突而挂起出错报文在内的环境数据。目录信息包括指针和状态，所述指针用于指向处理器中的地址空间，所述状态用于指示指针所指向的处理器的地址空间的具体状态，所述具体状态包括独占、共享和无效。例如，在出错前，正在访问某个控制器的某个地址空间，则指针指向该地址空间，并且状态项记录了这个地址空间的具体状态。而查表数据结构则记录访问这个地址空间时的环境数据。

[0038] 在另一种实施方式中，查表数据结构包括多个字段定义，所述字段定义用于记录事务处理的状态，包括报文的地址、目标 ID 号、是否处于冲突处理阶段、是否处于写操作阶段、是否因为冲突而挂起报文。当需要运行补丁时，事务处理处于中间状态，补丁知道中间状态以实现正确完成报文处理。同时，补丁处理完毕后，也需要把中间状态输出，保存下来，以便下次同样事务处理时使用。当然，最终事务处理完毕后，数据结构会被自动清除掉。

[0039] 如果处理器有足够的信息处理报文，则不需继续查询协议表，如果处理器没有足够的信息处理报文，则需要继续查询协议表。如果还需要继续查询协议表，则可以将需要查询的协议表的标识连同查表数据结构、目录信息以及出错报文一起发送给补丁模块。协议处理引擎继续处理与所述出错报文地址不同的后续报文，从而避免堵塞。

[0040] 增设的补丁相应接收节点控制器所发送的查表数据结构、目录信息以及出错报文。如果需要继续查询协议表，则补丁模块相应接收查表数据结构、目录信息、出错报文以及需要查询的协议表的标识。

[0041] S302：增设的补丁模块根据所述查表数据结构、所述目录信息以及所述出错报文执行补丁程序，其中，所述补丁程序用于代替所述协议处理引擎对出错报文进行处理以生成新的报文及生成相应的新的目录信息。

[0042] 增设的补丁模块检测出有需要进行处理的出错报文时，输入查表数据结构、目录信息以及出错报文，并根据查表数据结构、目录信息以及出错报文执行补丁程序，其中，补丁程序用于代替协议处理引擎对出错报文按照协议处理引擎原来的处理方式进行处理以

生成新的报文及生成相应的新的目录信息。例如，补丁程序可根据查表数据结构恢复对上述地址空间的操作，又从目录信息中获得该地址空间的状态，所以，补丁程序可代替协议处理引擎 221 对出错报文按照协议处理引擎原来的处理方式进行处理，并生成新的报文，处理完毕后，该地址空间的状态可能发生改变，于是更新目录信息从而产生新的目录信息。

[0043] 如果还需要继续查询协议表，则增设的补丁根据需要查询的协议表的标识查询协议表，然后根据查表数据结构、目录信息以及出错报文执行补丁程序以生成新的报文及生成相应的新的目录信息。增设的补丁将新的报文及新的目录信息插入到流水线后半段。其中，报文顺序排列构成了流水线。

[0044] 本实施方式中所述的方法中的所有内容可以应用于上述的补丁系统及下述的节点控制器中。

[0045] 参阅图 4，图 4 是本申请节点控制器一实施方式的结构示意图。本实施方式的节点控制器包括补丁模块，所述补丁模块耦接节点控制器内的协议处理引擎模块，所述补丁模块包括 :接收单元 410 以及执行单元 420。

[0046] 接收单元 410 用于在节点控制器内的协议处理引擎模块执行 Cache 一致性协议出错，挂起出错报文及继续处理与出错报文地址不同的后续报文时，接收节点控制器所发送的查表数据结构、目录信息以及出错报文。比如，当节点控制器内的某个协议处理引擎模块执行 Cache 一致性协议出错时，协议处理引擎模块挂起出错报文以暂停处理报文，包括当前出错报文以及与当前出错报文地址相同的报文，由于每个报文都有相应的地址，所以很容易查找到上述报文，其中地址是指 CPU 的系统地址，并将查表数据结构、目录信息以及出错报文发送给接收单元 410。查表数据结构用于记录包括出错报文地址、是否处于冲突处理阶段、是否处于写操作阶段以及是否因为冲突而挂起出错报文在内的环境数据。目录信息包括指针和状态，所述指针用于指向处理器中的地址空间，所述状态用于指示指针所指向的处理器的地址空间的具体状态，所述具体状态包括独占、共享和无效。例如，在出错前，正在访问某个控制器的某个地址空间，则指针指向该地址空间，并且状态项记录了这个地址空间的具体状态。而查表数据结构则记录访问这个地址空间时的环境数据。

[0047] 在另一种实施方式中，查表数据结构包括多个字段定义，所述字段定义用于记录事务处理的状态，包括报文的地址、目标 ID 号、是否处于冲突处理阶段、是否处于写操作阶段、是否因为冲突而挂起报文。当需要运行补丁时，事务处理处于中间状态，补丁知道中间状态以实现正确完成报文处理。同时，补丁处理完毕后，也需要把中间状态输出，保存下来，以便下次同样事务处理时使用。当然，最终事务处理完毕后，数据结构会被自动清除掉。

[0048] 如果处理器有足够的信息处理报文，则不需继续查询协议表，如果处理器没有足够的信息处理报文，则需要继续查询协议表。如果还需要继续查询协议表，则可以将需要查询的协议表的标识连同查表数据结构、目录信息以及出错报文一起发送给接收单元 410。协议处理引擎模块继续处理与所述出错报文地址不同的后续报文，从而避免堵塞。

[0049] 接收单元 410 相应接收节点控制器所发送的查表数据结构、目录信息以及出错报文。如果需要继续查询协议表，则接收单元 410 相应接收查表数据结构、目录信息、出错报文以及需要查询的协议表的标识。

[0050] 接收单元 410 将查表数据结构、目录信息以及出错报文向执行单元发送 420。

[0051] 执行单元 420 用于接收所述查表数据结构、目录信息以及出错报文，根据查表数

据结构、目录信息以及出错报文执行补丁程序，其中，补丁程序用于代替协议处理引擎模块对出错报文按照协议处理引擎原来的处理方式进行处理以生成新的报文及生成相应的新的目录信息。

[0052] 执行单元 420 检测出有需要进行处理的出错报文时，接收查表数据结构、目录信息以及出错报文，并根据查表数据结构、目录信息以及出错报文执行补丁程序，其中，补丁程序用于代替协议处理引擎模块对出错报文进行处理以生成新的报文及生成相应的新的目录信息。如果还需要继续查询协议表，则执行单元 420 根据需要查询的协议表的标识查询协议表，然后根据查表数据结构、目录信息以及出错报文执行补丁程序以生成新的报文及生成相应的新的目录信息。例如，补丁程序可根据查表数据结构恢复对上述地址空间的操作，又从目录信息中获得该地址空间的状态，所以，补丁程序可代替协议处理引擎 221 对出错报文按照协议处理引擎原来的处理方式进行处理，并生成新的报文，处理完毕后，该地址空间的状态可能发生改变，于是更新目录信息从而产生新的目录信息。

[0053] 执行单元 420 将新的报文及新的目录信息插入到流水线后半段。其中，报文顺序排列构成了流水线。

[0054] 参阅图 5，图 5 是本申请节点控制器另一实施方式的结构示意图。本实施方式的协议处理引擎容错处理装置包括补丁模块，补丁模块包括输入随机存取存储器 510、补丁处理器 520 以及输出随机存取存储器 530。输入随机存取存储器 510 的一端用于耦接节点控制器内的协议处理引擎模块，输入随机存取存储器 510 的另一端耦接补丁处理器 520 的一端，补丁处理器 520 的另一端耦接输出随机存取存储器 530 的一端，输出随机存取存储器 530 的另一端用于耦接节点控制器内的协议处理引擎模块。

[0055] 输入随机存取存储器 510 用于在节点控制器内的协议处理引擎模块执行 Cache 一致性协议出错，挂起出错报文及继续处理与出错报文地址不同的后续报文时，接收节点控制器所发送的查表数据结构、目录信息以及出错报文。比如，当节点控制器内的某个协议处理引擎执行 Cache 一致性协议出错时，协议处理引擎挂起出错报文以暂停处理报文，包括当前出错报文以及与当前出错报文地址相同的报文，由于每个报文都有相应的地址，所以很容易查找到上述报文，其中地址是指 CPU 的系统地址，并将查表数据结构、目录信息以及出错报文发送给输入随机存取存储器 510。查表数据结构用于记录包括出错报文地址、是否处于冲突处理阶段、是否处于写操作阶段以及是否因为冲突而挂起出错报文在内的环境数据。目录信息包括指针和状态，所述指针用于指向处理器中的地址空间，所述状态用于指示指针所指向的处理器的地址空间的具体状态，所述具体状态包括独占、共享和无效。例如，在出错前，正在访问某个控制器的某个地址空间，则指针指向该地址空间，并且状态项记录了这个地址空间的具体状态。而查表数据结构则记录访问这个地址空间时的环境数据。

[0056] 在另一种实施方式中，查表数据结构包括多个字段定义，所述字段定义用于记录事务处理的状态，包括报文的地址、目标 ID 号、是否处于冲突处理阶段、是否处于写操作阶段、是否因为冲突而挂起报文。当需要运行补丁时，事务处理处于中间状态，补丁知道中间状态以实现正确完成报文处理。同时，补丁处理完毕后，也需要把中间状态输出，保存下来，以便下次同样事务处理时使用。当然，最终事务处理完毕后，数据结构会被自动清除掉。

[0057] 如果处理器有足够的信息处理报文，则不需继续查询协议表，如果处理器没有足够的信息处理报文，则需要继续查询协议表。如果还需要继续查询协议表，则可以将需要查

询的协议表的标识连同查表数据结构、目录信息以及出错报文一起发送给输入随机存取存储器 510。协议处理引擎继续处理与所述出错报文地址不同的后续报文，从而避免堵塞。

[0058] 输入随机存取存储器 510 相应接收节点控制器所发送的查表数据结构、目录信息以及出错报文。如果需要继续查询协议表，则输入随机存取存储器 510 相应接收查表数据结构、目录信息、出错报文以及需要查询的协议表的标识。

[0059] 补丁处理器 520 用于根据查表数据结构、目录信息以及出错报文执行补丁程序，其中，补丁程序用于代替协议处理引擎对出错报文进行处理以生成新的报文及生成相应的新的目录信息。比如，补丁处理器 520 检测出输入随机存取存储器 510 有需要进行处理的出错报文时，接收查表数据结构、目录信息以及出错报文，并根据查表数据结构、目录信息以及出错报文执行补丁程序，其中，补丁程序用于代替协议处理引擎对出错报文按照协议处理引擎原来的处理方式进行处理以生成新的报文及生成相应的新的目录信息。如果还需要继续查询协议表，则补丁处理器 520 根据需要查询的协议表的标识查询协议表，然后根据查表数据结构、目录信息以及出错报文执行补丁程序以生成新的报文及生成相应的新的目录信息。例如，补丁程序可根据查表数据结构恢复对上述地址空间的操作，又从目录信息中获得该地址空间的状态，所以，补丁程序可代替协议处理引擎 221 对出错报文按照协议处理引擎原来的处理方式进行处理，并生成新的报文，处理完毕后，该地址空间的状态可能发生改变，于是更新目录信息从而产生新的目录信息。补丁处理器 520 将暂存于补丁处理器 520 中的新的报文及新的目录信息插入到流水线后半段。其中，报文顺序排列构成了流水线。

[0060] 输出随机存取存储器 730 用于暂存补丁处理器 720 输出的新的报文及新的目录信息。

[0061] 本申请还提供了一种协议处理引擎容错处理系统，包括补丁模块，补丁模块耦接节点控制器内的协议处理引擎模块。在一种具体的实施方式可参阅图 2 及相关的描述，此处不重复一一赘述。

[0062] 上述申请通过增设补丁模块，在协议处理引擎出错时，协议处理引擎挂起出错报文，并对继续对与出错报文地址不同的后续报文进行处理，而出错报文则通过补丁模块进行处理，从而保证在协议出错时，系统依然能够正常工作。

[0063] 在本申请所提供的几个实施方式中，应该理解到，所揭露的系统，装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施方式仅仅是示意性的，例如，所述模块或单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

[0064] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施方式方案的目的。

[0065] 另外，在本申请各个实施方式中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的

单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

[0066] 所述集成的单元如果以软件功能单元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取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或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)或处理器(processor)执行本申请各个实施方式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。

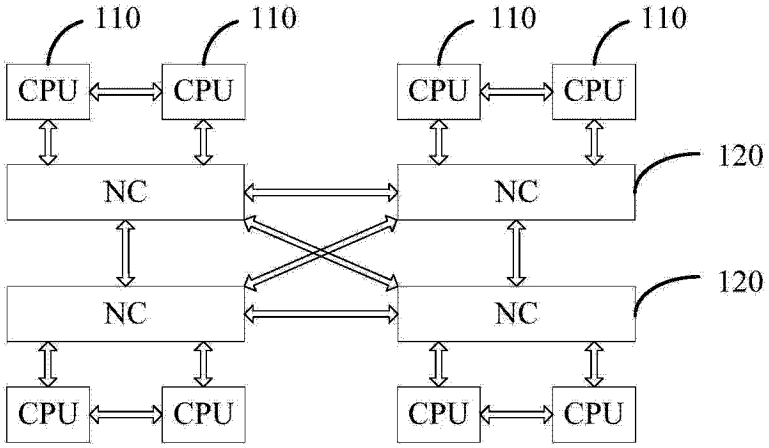

110
110
110
110
CPU
CPU
CPU
CPU
120
NC
NC
120
NC
NC
CPU
CPU
CPU
CPU

图 1

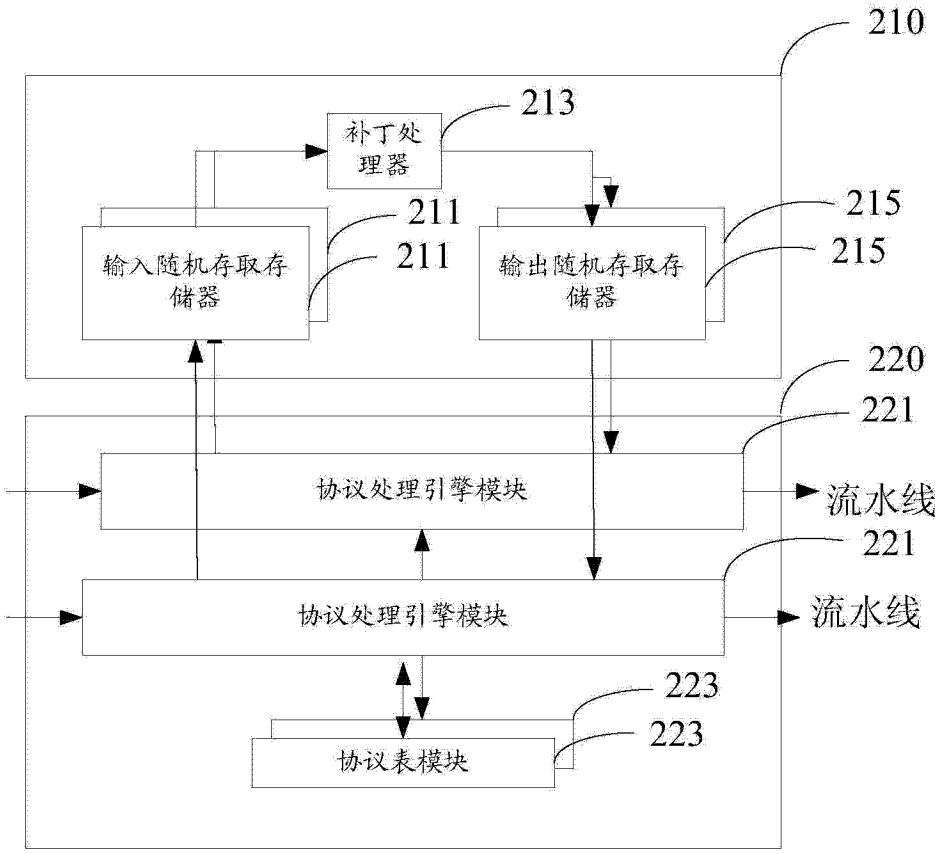

210
213
补丁处理器
211
211
215
215
输入随机存取存储器
输出随机存取存储器
220
221
协议处理引擎模块
流水线
221
协议处理引擎模块
流水线
223
223
协议表模块

图 2

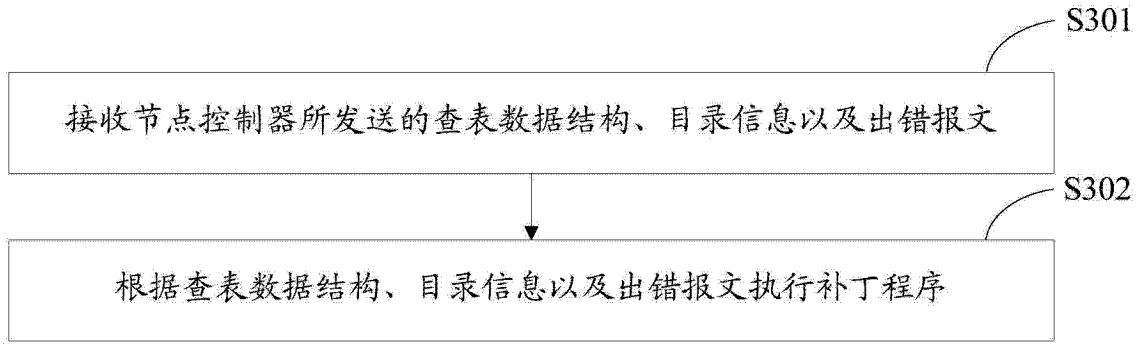

S301
接收节点控制器所发送的查表数据结构、目录信息以及出错报文
S302
根据查表数据结构、目录信息以及出错报文执行补丁程序

图 3

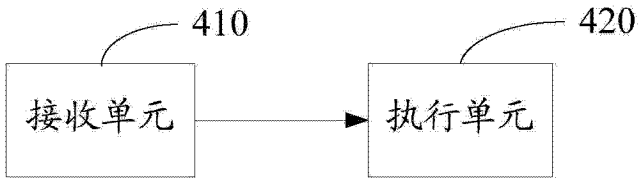

410
接收单元
420
执行单元

图 4

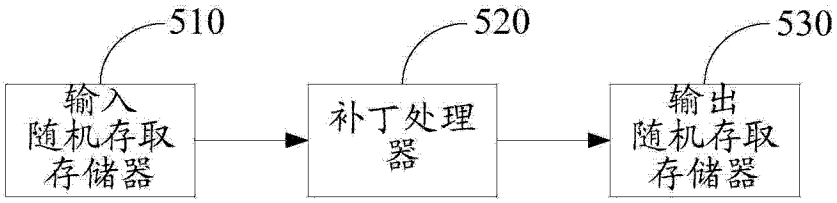

510
输入
随机存取
存储器
520
补丁处理
器
530
输出
随机存取
存储器

图 5