



(12) 发明专利

(10) 授权公告号 CN 103226496 B

(45) 授权公告日 2016. 05. 25

(21) 申请号 201310082439. 9

CN 101390079 A, 2009. 03. 18,

(22) 申请日 2013. 03. 15

EP 1574965 A1, 2005. 09. 14,

US 2010172257 A1, 2010. 07. 08,

(73) 专利权人 杭州华三通信技术有限公司

审查员 梁艳

地址 310053 浙江省杭州市高新技术产业开发区之江科技工业园六和路 310 号华为杭州生产基地

(72) 发明人 杨逸

(74) 专利代理机构 北京德琦知识产权代理有限公司 11018

代理人 杨春香 宋志强

(51) Int. Cl.

G06F 9/52(2006. 01)

(56) 对比文件

CN 101442513 A, 2009. 05. 27,

CN 102279771 A, 2011. 12. 14,

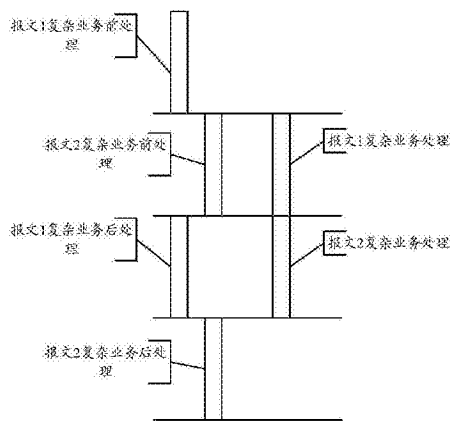
权利要求书3页 说明书8页 附图5页

(54) 发明名称

业务同步处理方法和多核设备

(57) 摘要

本申请提供了业务同步处理方法和多核设备。本方法中,通过将数据转发核的虚拟化为N个不同的VCPU,N个VCPU对应不同的栈空间、且同一时刻只有一个VCPU完全占用所述数据转发核的资源处于工作状态,而其他多个VCPU处于非工作状态,处于工作状态的VCPU在当前待处理的报文所涉及的业务为本VCPU不能独自执行的复杂业务,则将该报文传递给用于执行所述报文所涉及的业务的业务处理模块,释放本VCPU占用的所述数据转发核的资源,选取处于非工作状态的VCPU继续执行报文的处理,这能提高S worker的资源利用效率,避免S worker因为等待业务处理模块返回的处理结果而导致的资源浪费。



1. 一种业务同步处理方法,该方法应用于多核处理器中的数据转发核,其特征在于,将所述数据转发核虚拟化为N个不同的VCPU,该N个VCPU对应不同的栈空间、且同一时刻只有一个VCPU占用所述数据转发核的资源而处于工作状态,其它VCPU处于非工作状态;该方法包括:

A,任一VCPU在处于工作状态时,如果识别出当前待处理的报文所涉及的业务为本VCPU不能独自执行的复杂业务,则对所述报文进行复杂业务之前的简单业务处理,将处理后的报文传递给用于执行所述复杂业务的业务处理模块,保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间,释放本VCPU当前占用的数据转发核资源,从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源进入工作状态,同时将本VCPU的状态切换为非工作状态。

2. 根据权利要求1所述的方法,其特征在于,步骤A进一步包括:设置本VCPU的复杂业务处理标识为表示业务未完成的第一值;

步骤A中,所述从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源并进入工作状态包括:

检测每一处于非工作状态的VCPU的复杂业务处理标识;

当检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值时,选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源并进入工作状态,其中,任一VCPU对应的复杂业务处理标识为第二值,是指该VCPU传递至所述业务处理模块的报文已被所述业务处理模块进行复杂业务处理且返回至该VCPU;

当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU、且有至少一个VCPU不存在复杂业务处理标识时,按照设定的VCPU执行顺序选择下一个不存在复杂业务处理标识的VCPU占用该被释放的数据转发核资源并进入工作状态;

当检测到每一处于非工作状态的VCPU存在复杂业务处理标识、且该复杂业务处理标识为第一值时,执行同步等待,直至检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值,选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源进入工作状态。

3. 根据权利要求1所述的方法,其特征在于,步骤A中,如果VCPU识别出当前待处理的报文所涉及的业务为本VCPU独自执行的业务,则进一步包括:

对所述报文进行对应的业务处理;

在完成所述报文的处理后,检测每一处于非工作状态的VCPU对应的复杂业务处理标识;

当检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值时,保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间,释放本VCPU当前占用的数据转发核资源,将本VCPU的状态从当前的工作状态切换为非工作状态,并选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源进入工作状态;

当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU时,继续执行下一个报文的处理。

4. 根据权利要求2或3所述的方法,其特征在于,复杂业务处理标识为第二值的VCPU在被选择进入工作状态时,进一步包括:

获取所述业务处理模块返回的经过复杂业务处理的报文；

依据本VCPU对应的栈空间记录的运行轨迹处理所述业务处理模块返回的报文；

在完成报文的处理后删除本VCPU对应的栈空间记录的所述运行轨迹、以及删除VCPU的复杂业务处理标识,并检测每一处于非工作状态的VCPU对应的复杂业务处理标识,

当检测到至少一个处于非工作状态的VCPU对应的复杂业务处理标识为第二值时,保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间,释放本VCPU当前占用的数据转发核资源,将本VCPU的状态从当前的工作状态切换为非工作状态,并选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源并进入工作状态;

当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU时,继续执行下一个报文的处理。

5. 根据权利要求2所述的方法,其特征在于,不存在复杂业务处理标识的VCPU在被选择进入工作状态时,进一步包括:

从所述数据转发核的缓存中获取当前待处理的报文,依据本VCPU对应的栈空间记录的运行轨迹处理获取的报文。

6. 一种多核设备,其特征在于,所述多核设备包括至少一数据转发核,所述数据转发核包括:通过采用虚拟化技术从所述数据转发核虚拟化出的N个VCPU;其中,N个VCPU对应不同的栈空间、且同一时刻只有一个VCPU占用所述数据转发核的资源而处于工作状态,其它VCPU处于非工作状态;

每一VCPU包括:

识别单元,用于在自身VCPU占用所述数据转发核资源处于工作状态时识别出当前处理的报文涉及的业务是否为自身VCPU不能独自执行的业务;

传递单元,用于在所述识别单元的识别结果为是时,对所述报文进行复杂业务之前的简单业务处理,将处理后的报文传递给用于执行所述复杂业务的业务处理模块,并发送处理通知给处理单元;

处理单元,用于接收所述处理通知后,保存自身VCPU当前在所述数据转发核的运行轨迹至自身VCPU对应的栈空间,释放自身VCPU当前占用的数据转发核资源,从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源进入工作状态,同时将自身VCPU的状态切换为非工作状态。

7. 根据权利要求6所述的多核设备,其特征在于,所述处理单元进一步设置自身VCPU的复杂业务处理标识为表示业务未完成的第一值;

所述处理单元从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源并进入工作状态包括:

检测每一处于非工作状态的VCPU的复杂业务处理标识;

当检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值时,选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源并进入工作状态,其中,任一VCPU对应的复杂业务处理标识为第二值,是指该VCPU传递至所述业务处理模块的报文已被所述业务处理模块进行复杂业务处理且返回至该VCPU;

当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU、且有至少一个VCPU不存在复杂业务处理标识时,按照设定的VCPU执行顺序选择下一个不存在复杂

业务处理标识的VCPU占用该被释放的数据转发核资源并进入工作状态；

当检测到每一处于非工作状态的VCPU存在复杂业务处理标识、且该复杂业务处理标识为第一值时，执行同步等待，直至检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值，选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源进入工作状态。

8. 根据权利要求6所述的多核设备，其特征在于，所述处理单元进一步在所述识别单元的识别结果为否时，对所述报文进行对应的业务处理，在完成处理后，检测每一处于非工作状态的VCPU对应的复杂业务处理标识；

当检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值时，保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间，释放本VCPU当前占用的数据转发核资源，将本VCPU的状态从当前的工作状态切换为非工作状态，并选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源进入工作状态；

当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU时，继续执行下一个报文的处理。

9. 根据权利要求7或8所述的多核设备，其特征在于，每一VCPU进一步包括：业务单元，所述业务单元用于在自身VCPU的复杂业务处理标识为第二值且被选择进入工作状态时，执行以下操作：

获取所述业务处理模块返回的经过复杂业务处理的报文；

依据本VCPU对应的栈空间记录的运行轨迹处理所述业务处理模块返回的报文；

在完成报文的处理后删除本VCPU对应的栈空间记录的所述运行轨迹、以及删除VCPU的复杂业务处理标识，并检测每一处于非工作状态的VCPU对应的复杂业务处理标识，

当检测到至少一个处于非工作状态的VCPU对应的复杂业务处理标识为第二值时，保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间，释放本VCPU当前占用的数据转发核资源，将本VCPU的状态从当前的工作状态切换为非工作状态，并选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源并进入工作状态；

当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU时，继续执行下一个报文的处理。

10. 根据权利要求7所述的多核设备，其特征在于，每一VCPU进一步包括：获取单元；

所述获取单元用于在自身VCPU不存在复杂业务处理标识且被选择进入工作状态时，从所述数据转发核的缓存中获取当前待处理的报文，并依据本VCPU对应的栈空间记录的运行轨迹处理获取的报文。

业务同步处理方法和多核设备

技术领域

[0001] 本申请涉及网络通信技术,特别涉及业务同步处理方法和多核设备。

背景技术

[0002] 对基于CPU做报文转发的软转发系统而言,在CPU主频提升空间有限的情况下,通过多核来提高处理器整体转发性能成为一种趋势。同时,为了减少运行操作系统(OS: Operation System)的开销,目前,处理器的多核一般采用非对称多业务处理(ASMP: Asymmetric MultiProcessing)结构。该ASMP架构下,其中一个核运行OS,作为控制核(controller),负责内存等CPU资源的分配,在转发业务的处理上,负责路由的计算和生成并下刷路由表项给其它核。而其余的核作为数据转发核(worker),只负责从controller接收转发表项,做比较单一的报文查表转发处理,具体如图1所示。

[0003] 在实际的组网环境中,存在着各种各样的报文:有的报文处理起来比较简单,在一个worker核上即可完成处理;有的报文比较复杂,需要多个worker核相互协作才能完成处理。所以,对于worker,又细分成两种:一种是处理简单业务,称之为Simple worker(简称S worker),彼此之间并行处理。一种是处理复杂业务,称之为Complicated worker(简称C worker)。当一个需要复杂业务处理的报文到达S worker后,该S worker会先对该报文进行简单的初期处理,将处理后的“半成品”报文发送至C worker,由C worker进行“再加工”,这个时候,S worker和C worker之间就存在着一种同步的关系,不能并行处理。

[0004] 如图2所示,当S worker3通过业务接口接收报文时,识别该报文的处理是属于简单业务的处还是复杂业务的处理,如果是简单业务处理,则直接对该报文进行简单业务处理并查找相关转发表项进行转发即可,如果是复杂业务处理,则先对该报文进行简单的业务处理(该处理称为前期处理),将经过简单处理的报文递交给特定的负责复杂业务处理的C worker,C worker完成复杂业务处理后再把报文返回给S worker3,由该S worker3继续对该报文简单的业务处理(该处理称为后期处理),如图2所示S worker3与C worker之间的双向箭头所示。需要说明的是,实际应用中,图2中所有S worker都与C worker之间存在双向箭头,只不过这里以S worker3为例所以仅在图2中明显示出S worker3与C worker之间的双向箭头,而其他S worker与C worker之间的双向箭头未示出而已。

[0005] 另外,在实际应用中,在多核处理器之外还可能存在着专门处理某种特定复杂业务的硬件加速单元,比如专门处理报文加解密的ASIC芯片或者FPGA。如图3所示,多核处理器与处理特定业务的硬件加速单元通过高速数据总线相连。这个硬件加速单元可以是一个ASIC芯片(比如专门处理报文加解密的加密引擎),也可以是一个FPGA。当S worker通过业务接口接收报文时,发现该报文涉及到可以硬件加速的业务,则先对该报文进行简单的初期处理,之后通过高速数据总线把报文递交给硬件加速单元处理。等到硬件加速单元完成对该报文的处理并通过高速数据总线返回后,该S worker再继续对该报文的后期处理。

[0006] 从图2、图3可以看出,S worker与C worker或者硬件加速单元之间存在复杂业务的同步关系,对这种有同步关系的复杂业务,S worker在把报文递交给C worker或者外挂

的硬件加速单元处理的时候,本身只能处于等待状态,等到C worker或者硬件加速单元对该报文的复杂处理返回,并对该报文的复杂业务处理完成后才能再继续对报文进行处理,具体如图4所示。如果复杂业务处理的时间比较长,则S worker等待的时间就比较长,这对S worker的CPU浪费是相当可观的。

发明内容

[0007] 本申请提供了业务同步处理方法和多核设备,以避免S worker的资源浪费,提高S worker的资源利用效率。

[0008] 一种业务同步处理方法,该方法应用于多核处理器中的数据转发核,其中,将所述数据转发核虚拟化为N个不同的VCPU,该N个VCPU对应不同的栈空间、且同一时刻只有一个VCPU占用所述数据转发核的资源处于工作状态,而其它VCPU处于非工作状态;该方法包括:

[0009] A,任一VCPU在处于工作状态时,如果识别出当前待处理的报文所涉及的业务为本VCPU不能独自执行的复杂业务,则对所述报文进行复杂业务之前的简单业务处理,将处理后的报文传递给用于执行所述复杂业务的业务处理模块,保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间,释放本VCPU当前占用的数据转发核资源,从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源进入工作状态,同时将本VCPU的状态切换为非工作状态。

[0010] 一种多核设备,其关键在于,所述多核设备包括至少一数据转发核,所述数据转发核包括:通过采用虚拟化技术从所述数据转发核虚拟化出的N个VCPU;其中,N个VCPU对应不同的栈空间、且同一时刻只有一个VCPU占用所述数据转发核的资源处于工作状态,而其它VCPU处于非工作状态;

[0011] 每一VCPU包括:

[0012] 识别单元,用于在自身VCPU占用所述数据转发核资源处于工作状态时识别出当前处理的报文涉及的业务是否为自身VCPU不能独自执行的业务;

[0013] 传递单元,用于在所述识别单元的识别结果为是时,对所述报文进行复杂业务之前的简单业务处理,将处理后的报文传递给用于执行所述复杂业务的业务处理模块,并发送处理通知给处理单元;

[0014] 处理单元,用于接收所述处理通知后,保存自身VCPU当前在所述数据转发核的运行轨迹至自身VCPU对应的栈空间,释放自身VCPU当前占用的数据转发核资源,从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源进入工作状态,同时将自身VCPU的状态切换为非工作状态。

[0015] 由以上技术方案可以看出,本发明中,VCPU在识别出当前报文涉及的业务为本VCPU不能独自执行的业务,则先对所述报文进行复杂业务之前的简单业务处理,将处理后的报文传递给用于执行所述复杂业务的业务处理模块,也即,此时VCPU和业务处理模块之间存在同步的业务,为避免VCPU占用的数据转发核S worker的资源在该同步的业务等待时间内被闲置,则释放本VCPU当前占用的数据转发核资源,从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源进入工作状态,同时将本VCPU的状态切换为非工作状态。这样,该被选择处于工作状态的VCPU可以继续占用S worker的资源执行报文的处理,避免了数据转发核资源的资源浪费,提高数据转发核资源的利用效率。

附图说明

- [0016] 图1为ASMP架构下的多核处理器结构示意图；
- [0017] 图2为S worker与C worker之间的交互示意图；
- [0018] 图3为S worker与硬件加速单元之间的交互示意图；
- [0019] 图4为具有同步关系的复杂业务的处理示意图；
- [0020] 图5为本发明实施例提供的方法流程图；
- [0021] 图6为本发明实施例提供的数据核结构示意图；
- [0022] 图7为本发明实施例提供的详细方法流程图；
- [0023] 图8为本发明实施例提供的具有同步关系的复杂业务的处理示意图；
- [0024] 图9为本发明实施例提供的VCPU结构图。

具体实施方式

[0025] 为了使本发明的目的、技术方案和优点更加清楚,下面结合附图和具体实施例对本发明进行详细描述。

[0026] 本发明中,通过对S worker进行软件虚拟化,使得S worker在与C worker或者硬件加速单元同步等待的时间继续处理后需的报文,提高报文处理效率,也避免数据转发核的资源浪费。

[0027] 下面对本发明提供的方法进行描述:

[0028] 图5示出了本发明提供的方法流程图。如图5所示,该流程可包括以下步骤:

[0029] 步骤501,将S worker虚拟化为N个不同的VCPU。

[0030] 本步骤501中,可采用虚拟化技术将S worker虚拟化为N个不同的VCPU。

[0031] 其中,N可根据经验设置。

[0032] 本发明中,N个VCPU具有不同的标识,且分别对应不同的栈空间。另外,为便于描述该N个VCPU,本发明还从设定值比如1开始按照递增顺对这N个VCPU进行编号。图6示出了从1开始按照递增顺对这N个VCPU进行编号的实例。

[0033] 在本发明中,N个VCPU在同一时刻只有一个VCPU完全占用S worker的资源处于工作状态,而其他VCPU处于非工作状态,不可同时出现两个以上的VCPU共享占用S worker的资源处于工作状态。在初始,可预先假设编号为1的VCPU(比如VCPU1)被默认为处于工作状态进行报文的处理,而其他VCPU处于非工作状态。

[0034] 步骤502,处于工作状态的VCPU在识别出当前待处理的报文所涉及的业务为本VCPU不能独自执行的复杂业务,则先对所述报文进行复杂业务之前的简单业务处理,将处理后的报文传递给用于执行所述复杂业务的业务处理模块,保存本VCPU当前在S worker的运行轨迹至本VCPU对应的栈空间,释放本VCPU当前占用S worker资源,从处于非工作状态的VCPU中选择一个VCPU占用该被释放的S worker资源进入工作状态,同时将本VCPU的状态切换为非工作状态。

[0035] 以VCPU1处于工作状态为例,则VCPU1在处理发往S worker的报文时,识别当前报文涉及的业务,如果该识别出的业务为本VCPU1不能独自执行的复杂业务,则先对所述报文进行简单业务处理,这里,该简单业务处理实质为该报文进行复杂业务之前的前期处理。之

后,将处理后的报文传递给用于执行所述业务的业务处理模块。

[0036] 这样,VCPU1和业务处理模块之间就存在同步的业务,为避免VCPU1占用的S worker的资源在该同步业务时间内被闲置,则VCPU1保存本VCPU1当前在S worker的运行轨迹至本VCPU1对应的栈空间,释放本VCPU1占用的S worker资源,从处于非工作状态的VCPU中选择一个VCPU(以VCPU2为例)占用该被释放的S worker资源处于工作状态,同时将本VCPU1的状态切换为非工作状态。这样,该被选择处于工作状态的VCPU2可以在VCPU1和业务处理模块之间存在业务同步期间继续占用S worker资源执行报文的处理,避免了S worker的资源浪费,提高S worker的资源利用效率。

[0037] 作为本发明的一个实施例,VCPU1当前在S worker的运行轨迹可包括:S worker的CPU的寄存器信息、VCPU1当前运行的最后一条指令地址等。这里VCPU1之所以保存当前在S worker的运行轨迹,目的是便于在后续收到业务处理模块返回的报文(其具体为本VCPU1之前传递的报文执行完复杂业务处理后的报文)时,直接按照该记录的运行轨迹恢复对应的状态,以便在该恢复的状态下处理该收到的报文,实现业务的同步。

[0038] 作为本发明的一个实施例,上述的业务处理模块可为执行复杂业务处理的C worker或者硬件加速单元。

[0039] 下面以一个具体实施例对图5所示流程进行详细描述:

[0040] 参见图7,图7为本发明实施例提供的方法流程图。如图7所示,该流程可包括以下步骤:

[0041] 步骤701,将S worker虚拟化为N个VCPU。

[0042] 步骤702,在初始,默认VCPU1完全占用S worker的资源处于工作状态,而其他VCPU处于非工作状态。

[0043] 步骤703,VCPU1从S worker的缓存中获取一报文。

[0044] 本发明中,当S worker通过业务口接收到报文时,其会先暂时存放该接收的报文至缓存中,以便处于工作状态的VCPU来获取。

[0045] 步骤704,VCPU1识别该报文所涉及的业务,如果该业务为本VCPU1独自执行的业务,则执行步骤705,如果该业务为本VCPU1不能独自执行的业务,则执行步骤706。

[0046] 为便于识别报文所涉及的业务,本发明会预先建立一个访问控制列表(ACL),该ACL会记载三层IP地址和业务之间的对应关系,基于此,当VCPU1获取到报文时,其会依据该报文的三层IP地址在ACL中查找该三层IP地址对应的业务,将该查找到的业务确定为该报文所涉及的业务。

[0047] 本发明实施例中,本VCPU1不能独自执行的业务,意味着该业务比较复杂,其需要借助复杂的业务处理模块比如C worker或者硬件加速单元执行,而如果该业务为本VCPU1能独自执行的业务,则意味着该业务比较简单,本VCPU1可按照现有业务处理方式处理该报文比如转发报文,具体见步骤705。

[0048] 步骤705,VCPU1按照现有业务处理方式处理该报文,在完成处理后,继续从S worker的缓存中获取下一个报文,返回步骤704。

[0049] 本步骤705是在S worker起始工作时的步骤,此时VCPU1还未与其他VCPU发生工作状态切换,基于下文对复杂业务处理标识的描述,可以知道此时VCPU1不会与业务处理模块比如C worker或者硬件加速单元存在同步关系的业务,只能是继续对报文处理,以使得在S

worker只有简单的业务、且VCPUI还未与其他VCPUI进行工作状态的切换之前,始终是由VCPUI完全占用CPU进行报文的处理。

[0050] 步骤706,VCPUI对该报文进行对应的简单处理,将本VCPUI的标识携带该处理后的报文中并传递给业务处理模块执行复杂业务处理,设置本VCPUI的复杂业务处理标识为表示未完成的第一值比如0,保存本VCPUI当前在S worker的运行轨迹至本VCPUI对应的栈空间,释放本VCPUI占用的所述S worker的资源,由处于非工作状态的其中一个VCPUI(记为VCPUm)占用所述S worker的资源并切换为工作状态,而将本VCPUI的状态切换为非工作状态。之后执行步骤707。

[0051] 这里,VCPUI将报文传递给业务处理模块比如C worker或者硬件加速单元执行复杂业务处理,则意味着该VCPUI与业务处理模块比如C worker或者硬件加速单元存在同步关系的业务。为了保证该同步关系的业务顺利执行,则VCPUI释放本VCPUI占用的所述S worker的资源,由处于非工作状态的VCPUm占用所述S worker的资源并切换为工作状态,而将本VCPUI的状态切换为非工作状态,以由VCPUm继续执行工作,具体见步骤707。这相比于现有S worker盲目等待业务处理模块返回的处理结果,大大节约了S worker资源的浪费时间,提高S worker的资源利用效率。

[0052] 本发明中,当业务处理模块比如C worker或者硬件加速单元接收到来自S worker中任一VCPUI发送的报文时,其会识别出该报文携带的VCPUI标识,并对该报文进行相应的业务处理,在完成处理后,将处理后的报文发送至具有该VCPUI标识的VCPUI中,而当S worker检测到其一VCPUI收到业务处理模块比如C worker或者硬件加速单元返回的报文时,则意味着该VCPUI传递至业务处理模块的报文被业务处理模块处理完成,该S worker就会将该VCPUI的复杂业务处理标识设置为表示复杂业务处理完成的第二值比如1。

[0053] 本发明中,S worker的任一VCPUI与复杂业务处理模块比如C worker或者硬件加速单元之间报文的传递可通过共享内存、中断、消息或者特有的高速数据总线等执行,本发明并不具体限定。

[0054] 上述步骤702至步骤706为在S worker起始工作时该S worker上被默认处于工作状态的VCPUI执行的操作。而此时其他VCPUI还未处于工作状态过,基于此,本发明中,在N个VCPUI从1开始编号的前提下,m可为2。

[0055] 步骤707,VCPUm的状态被切换为工作状态后,从S worker的缓存获取报文。

[0056] 本步骤707在VCPUm首次处于工作状态的前提下执行,类似步骤703。

[0057] 步骤708,VCPUm识别该报文所涉及的业务,如果该业务为本VCPUm能独自执行的业务,则执行步骤709,如果该业务为本VCPUm不能独自执行的业务,则执行步骤713。

[0058] 本步骤708与上面描述的步骤704类似,这里不再赘述。

[0059] 步骤709,VCPUm按照现有业务处理方式处理该报文,在完成处理后,执行步骤710。

[0060] 本步骤709之所以不与上述步骤705类似,原因是:VCPUm不为S worker初始开始工作后首个处于工作状态的VCPUI,在VCPUm处于工作状态之前,已经有至少一个VCPUI曾处于工作状态过,并且,该曾经处于工作状态过的至少一个VCPUI可能已经收到业务处理模块比如C worker或者硬件加速单元返回的处理后的报文,为保证这些VCPUI及时处理来自业务处理模块的报文,本发明需要执行下述步骤710。

[0061] 步骤710,VCPUm检测处于非工作状态的VCPUI的复杂业务处理标识,当检测到至少

一个处于非工作状态的VCPu的复杂业务处理标识为表示业务完成的第二值时,执行步骤711,而当检测到处于非工作状态的VCPu中不存在复杂业务处理标识为第二值的VCPu时,继续从S worker的缓存中获取下一个报文,返回执行步骤708。

[0062] 步骤711,保存本VCPum当前在所述数据转发核的运行轨迹至本VCPum对应的栈空间,释放本VCPum占用的所述S worker的资源,切换本VCPum的状态为非工作状态,触使复杂业务处理标识为第二值的其中一个VCPu占用所述S worker的资源并切换为工作状态。之后执行步骤712。

[0063] 步骤712,复杂业务处理标识为第二值、且被切换为工作状态的VCPu依据本VCPu对应的栈空间所记录的运行轨迹处理所述业务处理模块返回的报文,在完成报文的处理后删除本VCPu对应的复杂业务处理标识、以及删除本VCPu对应的栈空间记录的所述运行轨迹,之后本VCPu按照步骤710中VCPum的操作执行。

[0064] 步骤713,VCPum对该报文进行对应的简单处理,将本VCPum的标识携带该处理后的报文中传递给业务处理模块比如C worker或者硬件加速单元执行复杂业务处理,设置本VCPum对应的复杂业务处理标识为表示未完成的第一值比如0,并检测各个处于非工作状态的VCPu的复杂业务处理标识,当检测到至少一个处于非工作状态的VCPu的复杂业务处理标识为表示业务完成的第二值时,执行步骤711,当检测到处于非工作状态的VCPu中不存在复杂业务处理标识为第二值的VCPu、且有至少一个VCPu不存在复杂业务处理标识时,执行步骤714,当检测到每个处于非工作状态的VCPu存在复杂业务处理标识、且该复杂业务处理标识为第一值时,执行步骤716。

[0065] 步骤714,保存本VCPum当前在所述数据转发核的运行轨迹至本VCPum对应的栈空间,释放本VCPum占用的所述S worker的资源,并切换本VCPum的状态为非工作状态,按照设定的VCPu执行顺序选择下一个不存在复杂业务处理标识的VCPu(记为VCPUn)占用所述S worker的资源进入工作状态,之后执行步骤715。

[0066] 步骤715,VCPUn被切换为工作状态后,依据本VCPUn对应的栈空间所记录的运行轨迹从S worker的缓存获取报文,并按照步骤708中VCPum的操作执行本VCPu的操作。

[0067] 步骤716,执行同步等待,直至至少一个处于非工作状态的VCPu对应的复杂业务处理标识为第二值时,保存本VCPum当前在所述S worker的运行轨迹至本VCPum对应的栈空间,释放本VCPum占用的所述S worker的资源,并切换本VCPum的状态为非工作状态,触使复杂业务处理标识为第二值的其中一个VCPu占用所述S worker的资源切换为工作状态并按照步骤712执行操作。

[0068] 至此,完成图7所示的流程。

[0069] 通过图7所示流程可以看出,本发明中,当任一处于工作状态的VCPu识别出本VCPu能独自处理当前报文涉及的业务(即该业务为简单业务),则处理该报文,而当识别出本VCPu能独自处理当前报文(记为报文1)涉及的复杂业务,则仅对该报文1进行简单的处理,并传递该处理后的报文1给业务处理模块,同时,主动释放本VCPu占用的S worker的资源,由其他VCPu重新切换为工作状态继续处理报文,如此切换N次(假如N个简单业务的处理时间至少等于1个复杂业务的处理时间),这样,当报文1的复杂业务已经处理完毕返回给之前处理该报文1的VCPu后,再由之前处理该报文1的VCPu切换回工作状态,继续处理该报文1,具体如图8示出。这能够避免现有技术中S worker等待同步业务所浪费的CPU,提高了

Sworker的资源利用效率,也能有效提升复杂业务的系统转发能力。

[0070] 以上对本发明提供的方法进行了描述,下面对本发明提供的多核设备进行描述:

[0071] 本发明实施例提供了多核设备的结构。其中,所述多核设备包括至少一数据转发核,而所述数据转发核包括:通过采用虚拟化技术从所述数据转发核虚拟化出的N个VCPU;其中,N个VCPU对应不同的栈空间、且同一时刻只有一个VCPU占用所述数据转发核的资源处于工作状态,而其它VCPU处于非工作状态。

[0072] 图9示出了每一VCPU的结构。

[0073] 如图9所示,每一VCPU包括:

[0074] 识别单元,用于在自身VCPU占用所述数据转发核资源处于工作状态时识别出当前处理的报文涉及的业务是否为自身VCPU不能独自执行的业务;

[0075] 传递单元,用于在所述识别单元的识别结果为是时,对所述报文进行复杂业务之前的简单业务处理,将处理后的报文传递给用于执行所述复杂业务的业务处理模块,并发送处理通知给处理单元;

[0076] 处理单元,用于接收所述处理通知后,保存自身VCPU当前在所述数据转发核的运行轨迹至自身VCPU对应的栈空间,释放自身VCPU当前占用的数据转发核资源,从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源进入工作状态,同时将自身VCPU的状态切换为非工作状态。

[0077] 优选地,所述处理单元进一步设置所述VCPU的复杂业务处理标识为表示业务未完成的第一值;

[0078] 基于此,所述处理单元从处于非工作状态的VCPU中选择一个VCPU占用该被释放的数据转发核资源并进入工作状态包括:

[0079] 检测每一处于非工作状态的VCPU的复杂业务处理标识;

[0080] 当检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值时,选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源并进入工作状态,其中,任一VCPU对应的复杂业务处理标识为第二值,是指该VCPU传递至所述业务处理模块的报文已被所述业务处理模块进行复杂业务处理且返回至该VCPU;

[0081] 当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU、且有至少一个VCPU不存在复杂业务处理标识时,按照设定的VCPU执行顺序选择下一个不存在复杂业务处理标识的VCPU占用该被释放的数据转发核资源并进入工作状态;

[0082] 当检测到每一处于非工作状态的VCPU存在复杂业务处理标识、且该复杂业务处理标识为第一值时,执行同步等待,直至检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值,选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源进入工作状态。

[0083] 优选地,所述处理单元进一步在所述识别单元的识别结果否时,对所述报文进行对应的业务处理,在完成处理后,检测每一处于非工作状态的VCPU对应的复杂业务处理标识;

[0084] 当检测到至少一个处于非工作状态的VCPU的复杂业务处理标识为第二值时,保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间,释放本VCPU当前占用的数据转发核资源,将本VCPU的状态从当前的工作状态切换为非工作状态,并选择复杂业务

处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源进入工作状态；

[0085] 当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU时，继续执行下一个报文的处理。

[0086] 优选地，本发明中，每一VCPU进一步包括：业务单元和获取单元。

[0087] 其中，所述业务单元用于在自身VCPU的复杂业务处理标识为第二值且被选择进入工作状态时，执行以下操作：

[0088] 获取所述业务处理模块返回的经过复杂业务处理的报文；

[0089] 依据本VCPU对应的栈空间记录的运行轨迹处理所述业务处理模块返回的报文；

[0090] 在完成报文的处理后删除本VCPU对应的栈空间记录的所述运行轨迹、以及删除VCPU的复杂业务处理标识，并检测每一处于非工作状态的VCPU对应的复杂业务处理标识，

[0091] 当检测到至少一个处于非工作状态的VCPU对应的复杂业务处理标识为第二值时，保存本VCPU当前在所述数据转发核的运行轨迹至本VCPU对应的栈空间，释放本VCPU当前占用的数据转发核资源，将本VCPU的状态从当前的工作状态切换为非工作状态，并选择复杂业务处理标识为第二值的其中一个VCPU占用该被释放的数据转发核资源并进入工作状态；

[0092] 当检测到处于非工作状态的VCPU中不存在复杂业务处理标识为第二值的VCPU时，继续执行下一个报文的处理。

[0093] 所述获取单元用于在自身VCPU不存在复杂业务处理标识且被选择进入工作状态时，执行以下操作：从所述数据转发核的缓存中获取当前待处理的报文，并依据本VCPU对应的栈空间记录的运行轨迹处理获取的报文。

[0094] 至此，完成图9所示的VCPU结构描述。

[0095] 以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明保护的范围之内。

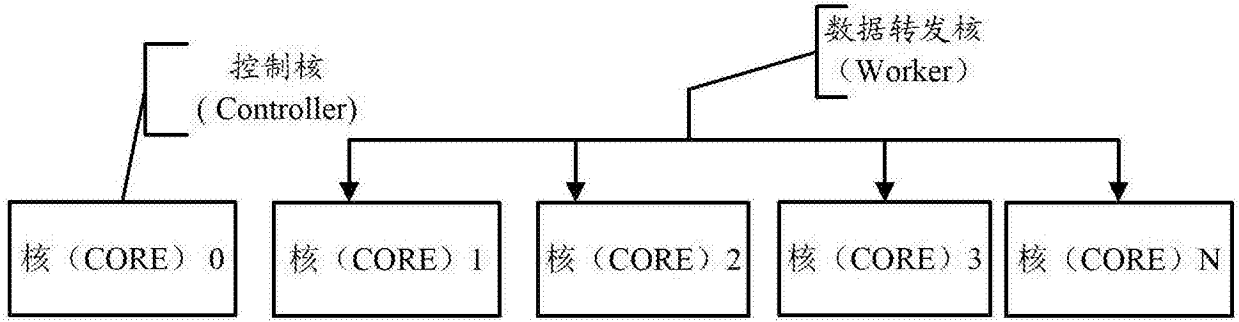


图1

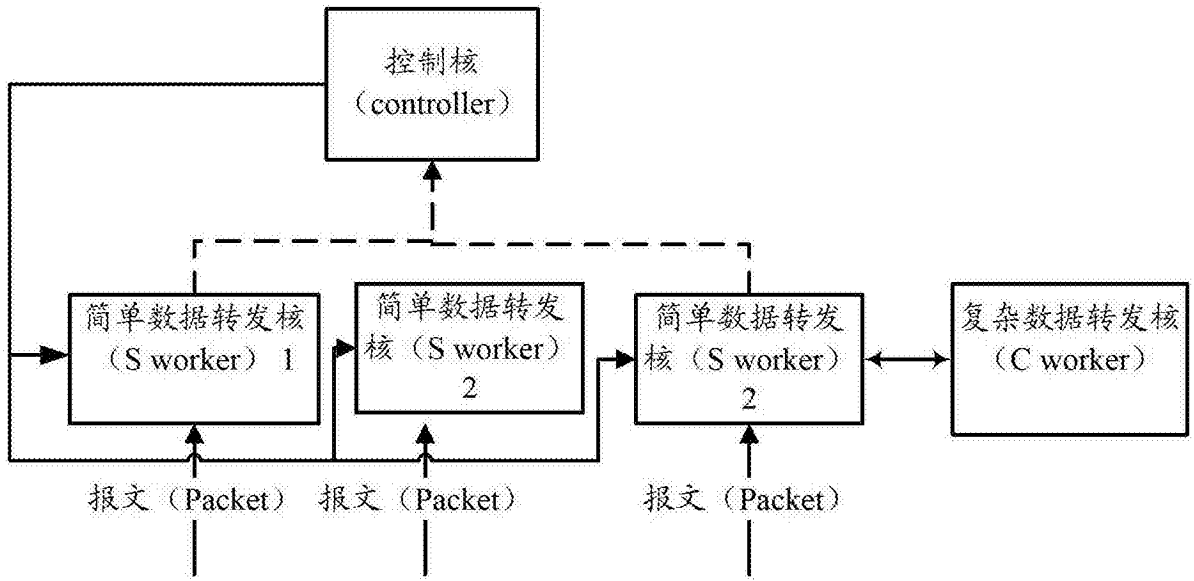


图2

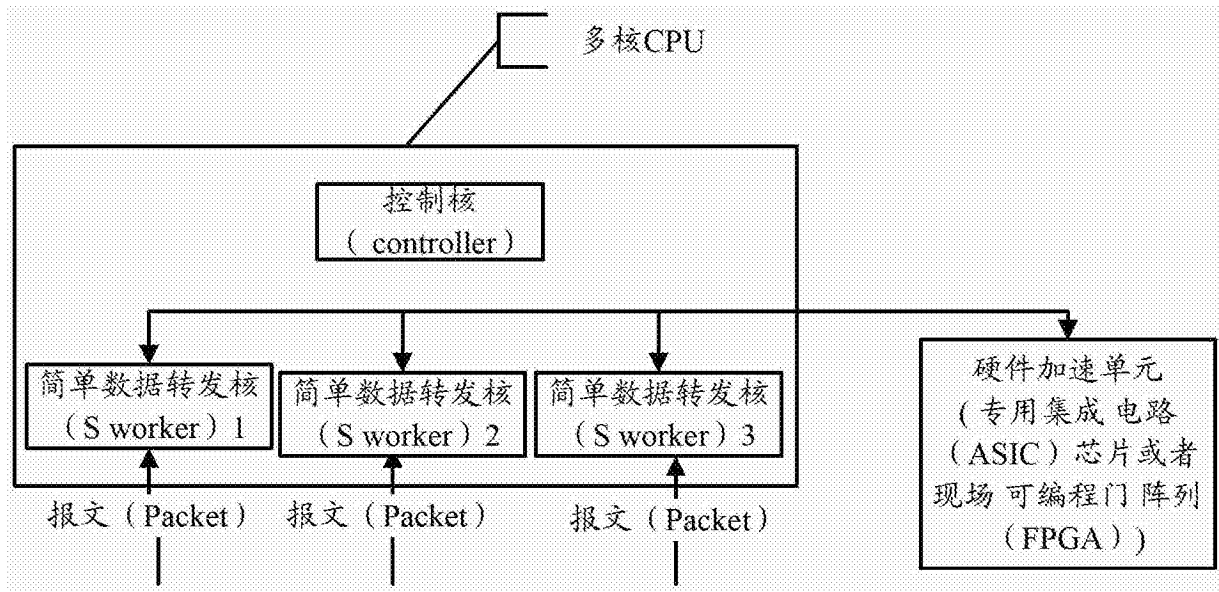


图3

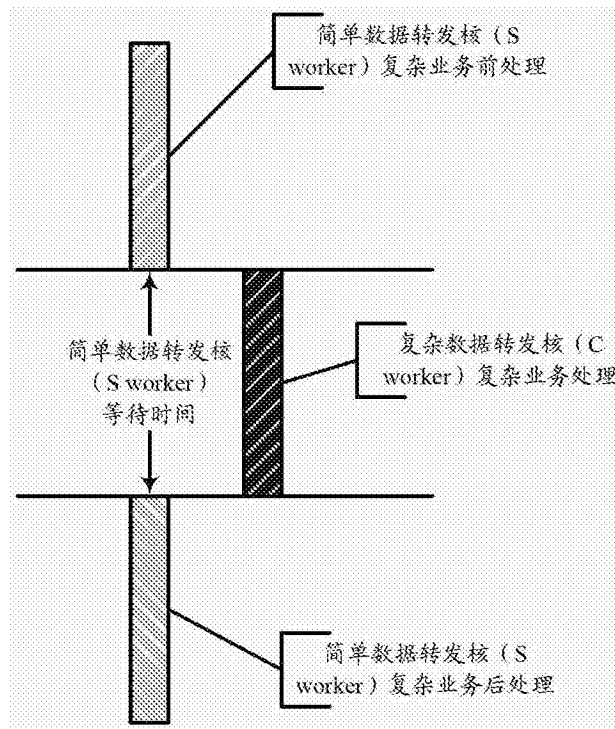


图4

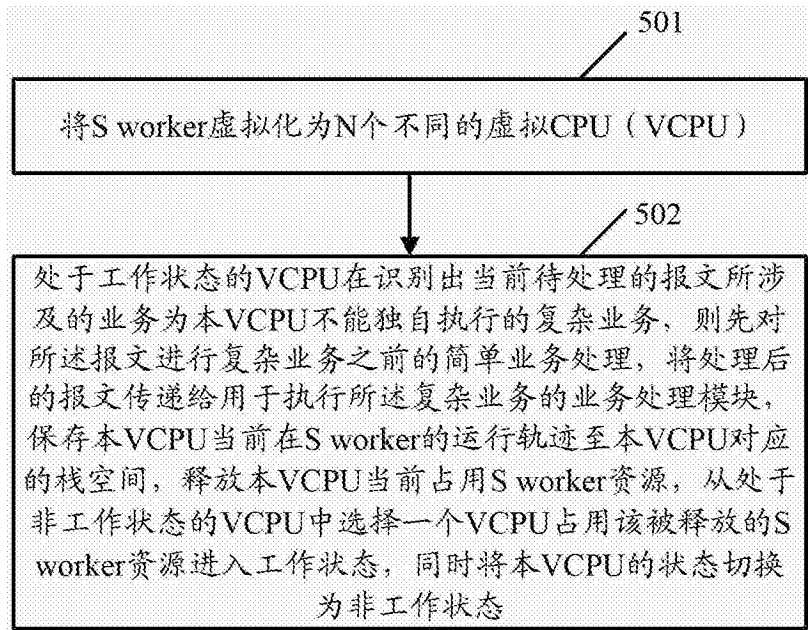


图5

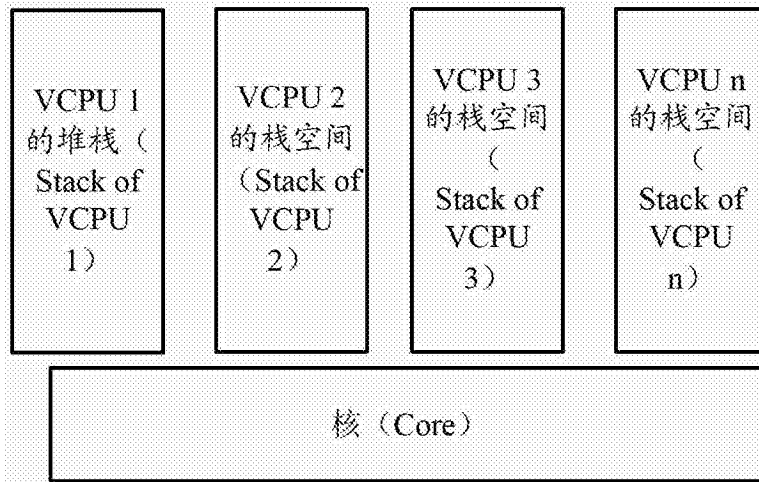


图6

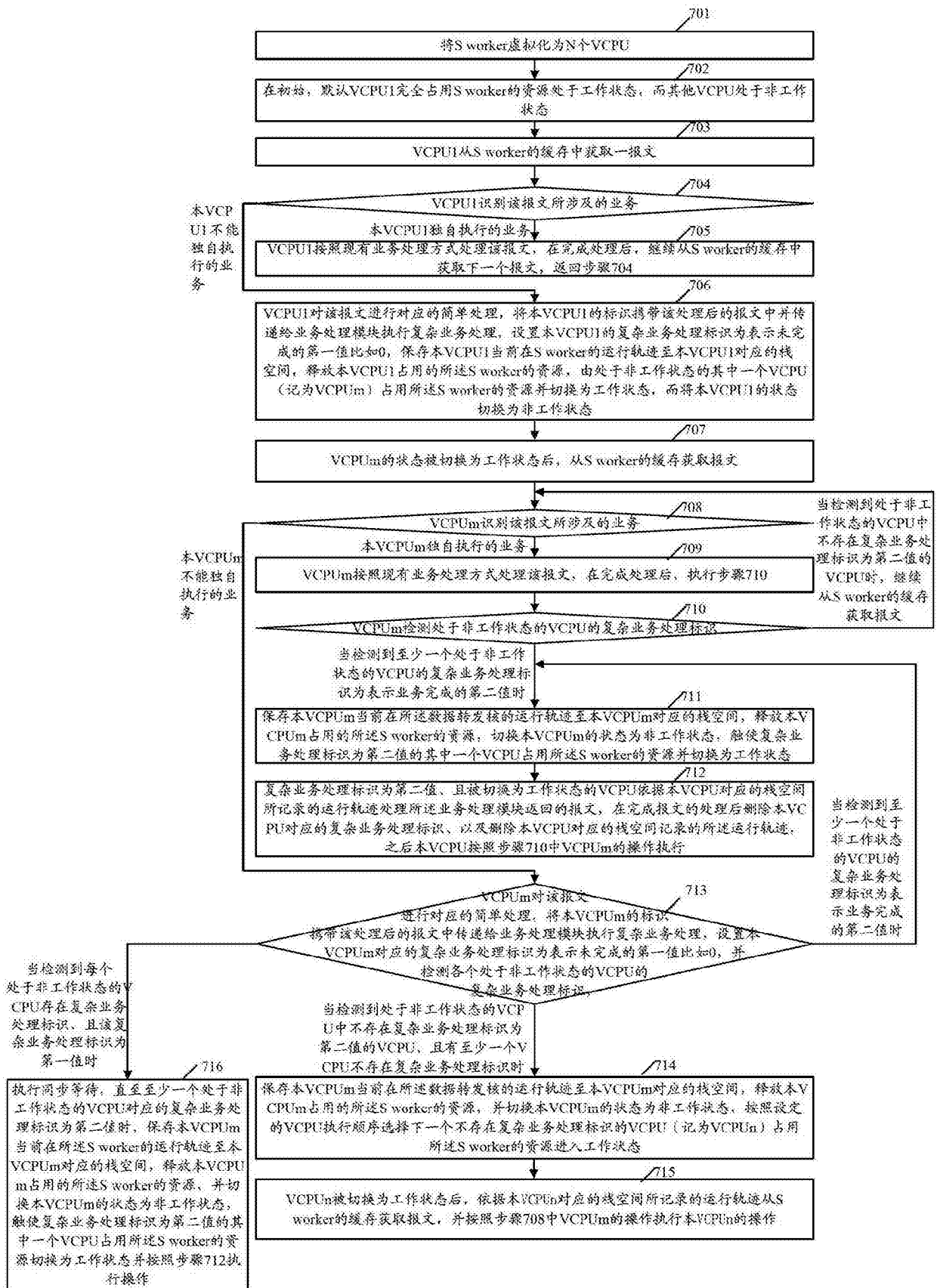


图7

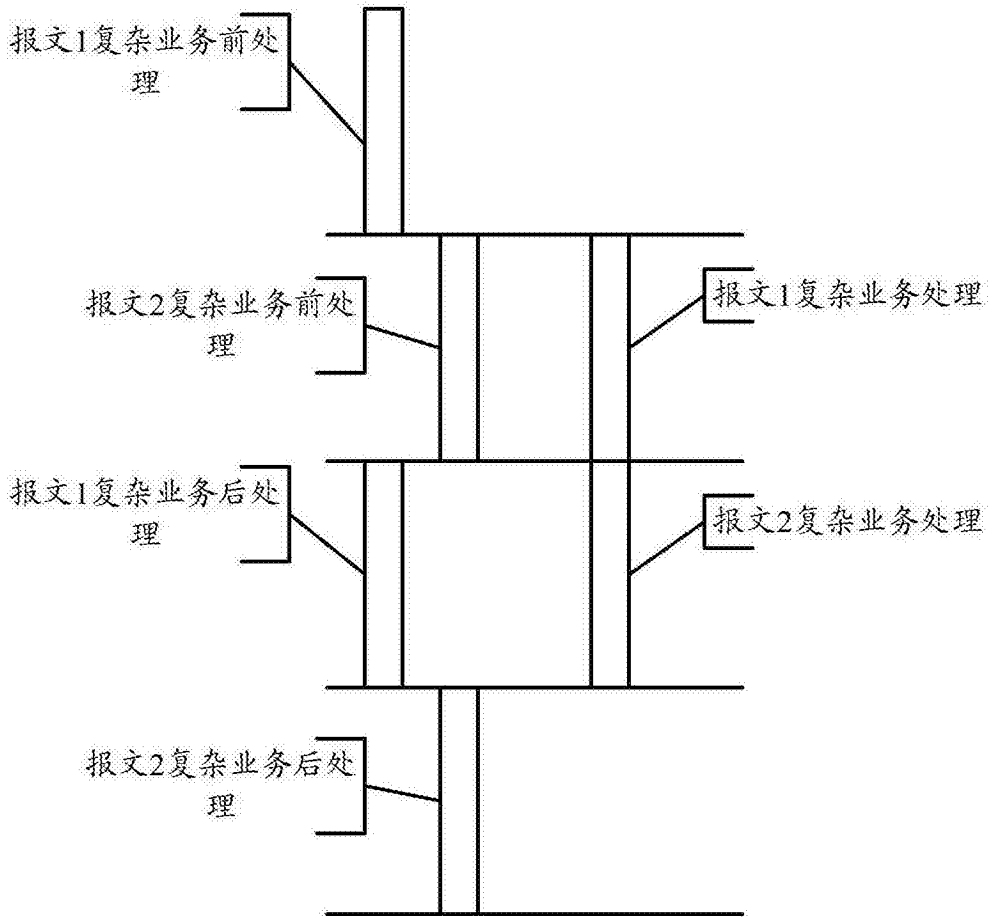


图8

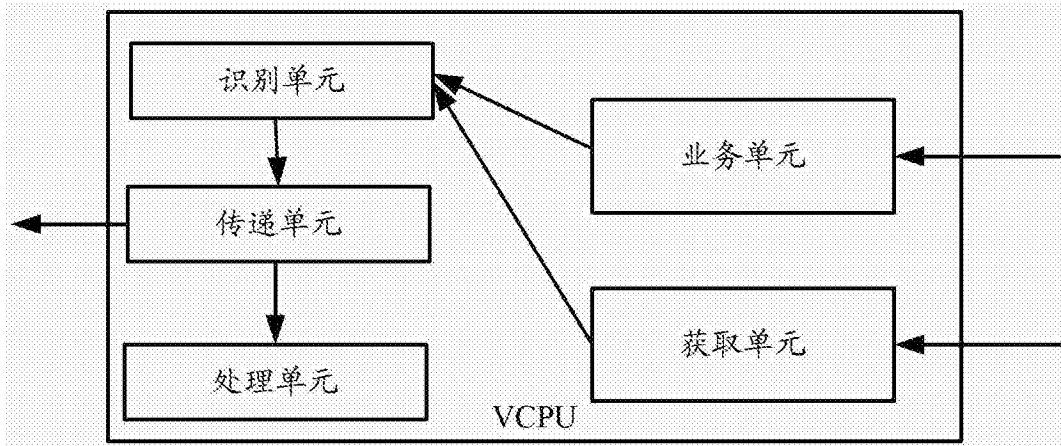


图9