



(12) 发明专利申请

(10) 申请公布号 CN 103677927 A

(43) 申请公布日 2014. 03. 26

(21) 申请号 201310694593. 1

(22) 申请日 2013. 12. 16

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为
总部办公楼

(72) 发明人 申洪 蒋培军

(74) 专利代理机构 北京亿腾知识产权代理事务
所 11309

代理人 李楠

(51) Int. Cl.

G06F 9/445(2006. 01)

G06F 9/54(2006. 01)

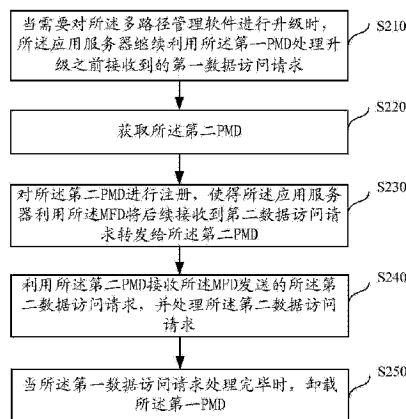
权利要求书2页 说明书11页 附图5页

(54) 发明名称

软件升级的方法及装置

(57) 摘要

本发明实施例涉及一种软件升级的方法及装置。当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一PMD处理升级之前接收到的第一数据访问请求;获取所述第二PMD;对所述第二PMD进行注册,使得所述应用服务器利用所述MFD将后续接收到第二数据访问请求转发给所述第二PMD;利用所述第二PMD接收所述MFD发送的所述第二数据访问请求,并处理所述第二数据访问请求;当所述第一数据访问请求处理完毕时,卸载所述第一PMD。从而可以保证在不中断数据访问请求处理的情况下,升级多路径管理软件。



1. 一种软件升级的方法,其特征在于,所述方法应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述方法用于将所述第一 PMD 升级成第二 PMD,所述方法包括:

当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求;

获取所述第二 PMD;

对所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将后续接收到第二数据访问请求转发给所述第二 PMD;

利用所述第二 PMD 接收所述 MFD 发送的所述第二数据访问请求,并处理所述第二数据访问请求;

当所述第一数据访问请求处理完毕时,卸载所述第一 PMD。

2. 根据权利要求 1 所述的方法,其特征在于,所述在对所述第二 PMD 进行注册之前,所述方法还包括:

向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据,根据所述全局内存数据对所述第二 PMD 进行初始化。

3. 根据权利要求 2 所述的方法,其特征在于,所述向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据包括:

对所述第一 PMD、所述 MFD 和所述第二 PMD 进行兼容性检查,当符合兼容性要求时,向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据。

4. 一种软件升级的方法,其特征在于,所述方法应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述方法用于将所述第一 PMD 升级成第二 PMD,所述方法包括:

当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求;

利用所述 MFD 处理后续接收到的第二数据访问请求;

当所述第一数据访问请求处理完毕时,卸载所述第一 PMD;

获取所述第二 PMD;

对所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将第三数据访问请求转发给所述第二 PMD,其中,所述第三数据访问请求为在对所述第二 PMD 进行注册之后,所述应用服务器接收到的数据访问请求;

利用所述第二 PMD 接收所述 MFD 发送的所述第三数据访问请求,并处理所述第三数据访问请求。

5. 一种软件升级的装置,其特征在于,所述装置应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其

中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述装置用于将所述第一 PMD 升级成第二 PMD,所述装置包括:第一处理单元、获取单元、注册单元、第二处理单元和卸载单元;

所述第一处理单元,用于当需要对所述多路径管理软件进行升级时,继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求;

所述获取单元,用于获取所述第二 PMD;

所述注册单元,用于对所述获取单元获取的所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将后续接收到第二数据访问请求转发给所述第二 PMD;

所述第二处理单元,用于利用所述注册单元注册后的所述第二 PMD 接收所述 MFD 发送的所述第二数据访问请求,并处理所述第二数据访问请求;

所述卸载单元,用于当所述第一数据访问请求处理完毕时,卸载所述第一 PMD。

6. 根据权利要求 5 所述的装置,其特征在于,所述装置还包括:发送单元,用于向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据,根据所述全局内存数据对所述第二 PMD 进行初始化。

7. 根据权利要求 6 所述的装置,其特征在于,所述发送单元具体用于:

对所述第一 PMD、所述 MFD 和所述第二 PMD 进行兼容性检查,当符合兼容性要求时,向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据。

8. 一种软件升级的装置,其特征在于,所述装置应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述装置用于将所述第一 PMD 升级成第二 PMD,所述装置包括:第一处理单元、第二处理单元、卸载单元、获取单元、注册单元和第三处理单元;

所述第一处理单元,用于当需要对所述多路径管理软件进行升级时,继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求;

所述第二处理单元,用于利用所述 MFD 处理后续接收到的第二数据访问请求;

所述卸载单元,当所述第一处理单元处理的第一数据访问请求处理完毕时,卸载所述第一 PMD;

所述获取单元,用于获取所述第二 PMD;

所述注册单元,用于对所述获取单元获取的所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将第三数据访问请求转发给所述第二 PMD,其中,所述第三数据访问请求为在对所述第二 PMD 进行注册之后,所述应用服务器接收到的数据访问请求;

所述第三处理单元,利用所述注册单元注册的所述第二 PMD 接收所述 MFD 发送的所述第三数据访问请求,并处理所述第三数据访问请求。

软件升级的方法及装置

技术领域

[0001] 本发明涉及信息技术领域,尤其涉及一种软件升级的方法及装置。

背景技术

[0002] 在 IT 领域,在应用服务器与存储阵列组成的应用环境中,为了避免应用服务器与存储阵列之间的一条链路出现故障时应用程序上的业务便中断,并且为了提高数据传输的可靠性,在应用服务器与存储阵列之间一般采用多条链路(称为路径)相连的方式。运行在应用服务器的操作系统内核的多路径管理软件就负责管理多条路径,在一条路径发生故障时可以把业务流切换到其他路径,保证业务不中断,但是多路径管理软件通常需要升级。

[0003] 现有技术中,多路径管理软件运行在应用服务器的操作系统的内核态(称为内核驱动程序),而对于内核驱动程序,如果要升级,通常需要中断与待升级内核驱动程序相依赖的业务,对于有的内核驱动程序还需要重启系统才能生效,长期以来多路径管理软件的升级均需重启主机,业务必然会中断一次才能完成多路径管理软件升级。

发明内容

[0004] 本发明实施例提供了一种软件升级的方法及装置,以实现在不中断数据访问请求处理的情况下,升级多路径管理软件。

[0005] 第一方面,提供了一种软件升级的方法,该方法应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述方法用于将所述第一 PMD 升级成第二 PMD,该方法包括:

[0006] 当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求;

[0007] 获取所述第二 PMD;

[0008] 对所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将后续接收到第二数据访问请求转发给所述第二 PMD;

[0009] 利用所述第二 PMD 接收所述 MFD 发送的所述第二数据访问请求,并处理所述第二数据访问请求;

[0010] 当所述第一数据访问请求处理完毕时,卸载所述第一 PMD。

[0011] 根据第一方面,在第一方面的第一种可能的实现方式中,所述在对所述第二 PMD 进行注册之前,所述方法还包括:

[0012] 向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据,根据所述全局内存数据对所述第二 PMD 进行初始化。

[0013] 根据第一方面的第一种可能的实现方式,在第一方面的第二种可能的实现方式

中,所述向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据包括:

[0014] 对所述第一 PMD、所述 MFD 和所述第二 PMD 进行兼容性检查,当符合兼容性要求时,向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据。

[0015] 第二方面,提供了一种软件升级的方法,该方法应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述方法用于将所述第一 PMD 升级成第二 PMD,该方法包括:

[0016] 当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求;

[0017] 利用所述 MFD 处理后续接收到的第二数据访问请求;

[0018] 当所述第一数据访问请求处理完毕时,卸载所述第一 PMD;

[0019] 获取所述第二 PMD;

[0020] 对所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将第三数据访问请求转发给所述第二 PMD,其中,所述第三数据访问请求为在对所述第二 PMD 进行注册之后,所述应用服务器接收到的数据访问请求;

[0021] 利用所述第二 PMD 接收所述 MFD 发送的所述第三数据访问请求,并处理所述第三数据访问请求。

[0022] 第三方面,提供了一种软件升级的装置,该装置应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述装置用于将所述第一 PMD 升级成第二 PMD,该装置包括:第一处理单元、获取单元、注册单元、第二处理单元和卸载单元;

[0023] 所述第一处理单元,用于当需要对所述多路径管理软件进行升级时,继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求;

[0024] 所述获取单元,用于获取所述第二 PMD;

[0025] 所述注册单元,用于对所述获取单元获取的所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将后续接收到第二数据访问请求转发给所述第二 PMD;

[0026] 所述第二处理单元,用于利用所述注册单元注册后的所述第二 PMD 接收所述 MFD 发送的所述第二数据访问请求,并处理所述第二数据访问请求;

[0027] 所述卸载单元,用于当所述第一数据访问请求处理完毕时,卸载所述第一 PMD。

[0028] 根据第三方面,在第三方面的第一种可能的实现方式中,所述装置还包括:发送单元,用于向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据,根据所述全局内存数据对所述第二 PMD 进行初始化。

[0029] 根据第三方面的第一种可能的实现方式,在第三方面的第二种可能的实现方式中,所述发送单元具体用于:

[0030] 对所述第一PMD、所述MFD和所述第二PMD进行兼容性检查,当符合兼容性要求时,向所述第二PMD发送初始化指示信息,使得所述第二PMD通过所述MFD获取所述多路径管理软件运行时所需的全局内存数据。

[0031] 第四方面,提供了一种软件升级的装置,该装置应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动MFD和第一路径管理驱动PMD,所述装置用于将所述第一PMD升级成第二PMD,该装置包括:第一处理单元、第二处理单元、卸载单元、获取单元、注册单元和第三处理单元;

[0032] 所述第一处理单元,用于当需要对所述多路径管理软件进行升级时,继续利用所述第一PMD处理升级之前接收到的第一数据访问请求;

[0033] 所述第二处理单元,用于利用所述MFD处理后续接收到的第二数据访问请求;

[0034] 所述卸载单元,当所述第一处理单元处理的第一数据访问请求处理完毕时,卸载所述第一PMD;

[0035] 所述获取单元,用于获取所述第二PMD;

[0036] 所述注册单元,用于对所述获取单元获取的所述第二PMD进行注册,使得所述应用服务器利用所述MFD将第三数据访问请求转发给所述第二PMD,其中,所述第三数据访问请求为在对所述第二PMD进行注册之后,所述应用服务器接收到的数据访问请求;

[0037] 所述第三处理单元,利用所述注册单元注册的所述第二PMD接收所述MFD发送的所述第三数据访问请求,并处理所述第三数据访问请求。

[0038] 本发明实施例提供的软件升级的方法及装置,当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一PMD处理升级之前接收到的第一数据访问请求;获取所述第二PMD;对所述第二PMD进行注册,使得所述应用服务器利用所述MFD将后续接收到第二数据访问请求转发给所述第二PMD;利用所述第二PMD接收所述MFD发送的所述第二数据访问请求,并处理所述第二数据访问请求;当所述第一数据访问请求处理完毕时,卸载所述第一PMD。这种由第一PMD处理多路径管理软件升级之前接收到的第一数据访问请求,并且由第二PMD同时处理后续接收到的第二数据访问请求方式,且在第一数据访问请求处理完毕时,卸载第一PMD的方式,保证了在将第一PMD升级为第二PMD时,即升级多路径管理软件时,不中断数据访问请求的处理。

附图说明

[0039] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0040] 图1为本发明提供的多路径管理软件逻辑示意图;

[0041] 图2为本发明实施一提供的软件升级的方法流程图;

[0042] 图3为本发明实施例一提供的软件升级的方法示意图之一;

[0043] 图4为本发明实施例一提供的软件升级的方法示意图之二;

[0044] 图5为本发明实施例一提供的软件升级的方法示意图之三;

- [0045] 图 6 为本发明实施例一提供的软件升级的方法示意图之四；
[0046] 图 7 为本发明实施例二提供的软件升级的方法流程图；
[0047] 图 8 为本发明实施例二提供的软件升级的方法示意图之一；
[0048] 图 9 为本发明实施例二提供的软件升级的方法示意图之二；
[0049] 图 10 为本发明实施例二提供的软件升级的方法示意图之三；
[0050] 图 11 为本发明实施例二提供的软件升级的方法示意图之四；
[0051] 图 12 为本发明实施例三提供的软件升级的装置示意图；
[0052] 图 13 为本发明实施例四提供的软件升级的装置示意图。

具体实施方式

[0053] 下面通过附图和实施例,对本发明的技术方案做进一步的详细描述。

[0054] 在实际的应用环境中,通常存储系统包括应用服务器和存储设备,应用服务器和所述存储设备之间通过多条路径进行通信,其中应用服务器中包含多路径管理软件,而多路径管理软件又包括两个驱动:多路径框架驱动(Multipath Framework Driver,MFD)和路径管理驱动(Path Management Driver,PMD),多路径框架驱动与操作系统强依赖但是与多路径业务特性功能无关,这个驱动在多路径管理软件升级过程中维持不变;路径管理驱动一般可以为多个,它与多路径业务功能强相关但是与操作系统不依赖,在多路径管理驱动升级过程中进行更新。

[0055] 图 1 为本发明提供的多路径管理软件逻辑示意图。图 1 中,多路径管理软件 10 包括两个驱动:多路径框架驱动 101 和路径管理驱动 102。多路径框架驱动 101 与操作系统上下驱动层进行 I/O 直接交互,实现与系统设备栈关联的功能,举例为,在应用服务器与存储设备组成的应用环境中,当在存储设备上新增或者删除 LUN 或者磁盘空间时,多路径框架驱动 101 即可捕获到该新增或者删除磁盘空间的事件,然后通知路径管理驱动 102 去新建或删除存储设备与应用服务器之间的物理路径;多路径框架驱动 101 同时还实现一个很重要的功能,即对多路径管理驱动中所有全局数据的管理,包括内存申请和释放,通过对外提供接口让路径管理驱动 102 获取全局内存地址和配置数据,但是路径管理驱动 102 只使用但不负责这些数据的生命周期。

[0056] 路径管理驱动 102,不与系统具体驱动及设备栈层次直接关联,解除了与系统的耦合,相应地,升级多路径管理软件 10 时,只需要升级路径管理驱动 102。路径管理驱动 102 是一堆业务功能函数集,实现多路径的业务逻辑功能,包括:路径管理,举例为,物理路径的上报删除,即在接收到多路径框架驱动 101 发送的管理路径的通知时,新建或删除存储设备与应用服务器之间的物理路径;选路负载,举例为,通过轮询算法、最小队列深度算法或最小任务选路算法去有效利用各条物理路径,以达到负载均衡;错误处理,举例为,通过错误码翻译、I/O 重试和 I/O 倒换等进行错误处理;路径健康管理,周期性对各条物理路径进行检查,对出现故障的路径进行隔离并降级检查,恢复已修复好的路径;I/O 切换,举例为,在已存在的 n 条物理路径中,第 i 条路径出现故障,则将原本通过第 i 条路径转发的业务流重定向至其他路径进行业务流的转发。

[0057] 需要说明的是,在多路径管理软件 10 中包括多个路径管理驱动 102 时,各路径管理驱动 102 通过其提供的对外接口注册关于各路径管理驱动 102 的信息,在有业务流下发

时,多路径框架驱动 101 根据一定的管理规则(举例为,在某一时刻,调用某个接口)调用不同的接口,以实现对各路径管理驱动 102 的调用,即在一个多路径管理驱动中可同时运行多个路径管理驱动 102,从而为多路径管理软件升级不中断业务提供了基础。

[0058] 图 2 为本发明实施一提供的软件升级的方法流程图,该方法用于将第一路径管理驱动升级成第二路径管理驱动,如图 2 所示,本实施例具体包括以下步骤:

[0059] 步骤 210,当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求。

[0060] 参见图 3 所示的本发明实施例一提供的软件升级的方法示意图之一,图 3 中,多路径管理软件 30 包括两个驱动:多路径框架驱动 MFD301 和第一路径管理驱动 PMD302,对多路径管理软件 30 的升级即为对第一路径管理驱动 302 的升级。具体地,多路径框架驱动 301 定义一个运行时多路径内存数据管理的数据结构,它通过但不限于虚拟磁盘列表和管理的存储产品信息列表将多路径管理软件 30 运行时的全局内存数据统一管理起来。这样可以保证在多路径管理软件 30 包括多个路径管理驱动,且在升级其中一个路径管理驱动时,运行时全局公共内存数据仍然可用,其他路径管理驱动能够获得到,这种多个路径管理驱动共享的同一套多路径内存数据可实现业务流的并行处理,还可以实现在不影响业务流的下发和返回的情况下实现将旧路径管理驱动升级成新路径管理驱动,此处,对业务流的处理通过对数据访问请求的处理来实现,而通常的数据访问请求即为 I/O 请求。此外,多路径框架驱动 301 还需定义一个管理 I/O 的数据结构,该结构包含指向系统 I/O 结构体的指针,还包含一个记录 I/O 处理的路径管理驱动的 HOOK 指针,该 HOOK 指针即为路径管理驱动对外提供的函数接口,这样的设计是为了保证在有多个路径管理驱动时,也即可以调用多个路径管理驱动同时处理 I/O 时,保证 I/O 的下发和返回都是同一个路径管理驱动处理,满足升级过程中, I/O 从升级前的路径管理驱动平滑过渡到升级后的路径管理驱动而业务流的转发不受影响的要求。具体地,当需要对所述多路径管理软件 30 进行升级时,应用服务器继续利用所述第一路径管理驱动处理升级之前接收到的第一数据访问请求,具体实现为,多路径框架驱动 301 定义一个管理 I/O 的数据结构,该数据结构中包含一个记录 I/O 处理的第一路径管理驱动 302 的 HOOK 指针,该 HOOK 指针即为第一路径管理驱动 302 对外提供的函数接口,记录为第一接口,通过第一路径管理驱动 302 对应的第一接口调用第一路径管理驱动 302 (该第一接口是由第一路径管理驱动 302 事先注册到多路径框架驱动 301 中),然后该第一路径管理驱动 302 选择相应的物理路径进行第一数据访问请求的处理,其中第一路径管理驱动 302 为升级前的路径管理驱动。

[0061] 需要说明的是,对于多路径管理软件 30 运行时所需的内存数据,多路径框架驱动 301 通过定义数据结构的方式对上述内存数据进行管理,可以将上述内存数据分为基本部分和扩展部分,基本部分包含对象基本信息等跨版本的公共成员,足以实现业务流处理过程中最简单的选路,这部分数据一直保持不变;扩展部分包存支撑复杂路径管理功能的,很有可能随版本升级而修改新增成员变量,为各版本路径管理驱动私有部分,由各路径管理驱动自己构造维护,适应变化。

[0062] 步骤 220,获取所述第二 PMD。

[0063] 参见图 4 所示的本发明实施例一提供的软件升级的方法示意图之二,图 4 中,应用服务器先获取第二路径管理驱动 403。

[0064] 步骤 230, 对所述第二 PMD 进行注册, 使得所述应用服务器利用所述 MFD 将后续接收到第二数据访问请求转发给所述第二 PMD。

[0065] 图 4 中, 应用服务器在获取到第二路径管理驱动 403 之后, 检查多路径框架驱动 401、第一路径管理驱动 402 和第二路径管理驱动 403 的兼容性, 当符合兼容性要求时, 将第二路径管理驱动 403 加载到操作系统的内核中, 并向第二路径管理驱动 403 发送初始化指示信息; 第二路径管理驱动 403 接收到上述初始化指示信息之后, 通过所述多路径框架驱动 401 对外提供的接口获取多路径管理软件 40 运行时所需的全局内存数据, 根据上述全局内存数据对自身进行初始化, 完成初始化的第二路径管理驱动 403 就具备了处理数据访问请求的能力, 包括 I/O 的下发和返回; 最后对第二路径管理驱动 403 进行注册, 具体地, 第二路径管理驱动 403 向路径框架驱动 401 发送注册接口请求信息, 路径框架驱动 401 接收到请求信息后, 可在其定义的管理 I/O 的数据结构中新增一个 I/O 处理的第二路径管理驱动 403 的 HOOK 指针(记录为第二接口), 之后多路径框架驱动 401 通过第二路径管理驱动 403 对应的第二接口, 即可实现将后续接收到的第二数据访问请求转发给第二路径管理驱动 403。

[0066] 步骤 240, 利用所述第二 PMD 接收所述 MFD 发送的所述第二数据访问请求, 并处理所述第二数据访问请求。

[0067] 参见图 5 所示的本发明实施例一提供的软件升级的方法示意图之三, 图 5 中, 在对第二路径管理驱动 503 进行注册之后, 多路径框架驱动 501 将新注册的第二接口标识为优先使用的状态, 以实现当应用服务器接收到后续的第二数据访问请求时, 多路径框架驱动 501 通过标识为优先使用状态的第二接口向第二路径管理驱动 503 发送第二数据访问请求, 由第二路径管理驱动 503 处理接收到的第二数据访问请求。图 5 中, 第一路径管理驱动 502 和第二路径管理驱动 503 同时在处理数据访问请求, 因为它们都可以通过多路径框架驱动 501 对外提供的接口获取全局内存数据。上述这种第一路径管理驱动 502 和第二路径管理驱动 503 同时在处理数据访问请求的方式, 保证了不中断数据访问请求处理的情况下, 升级多路径管理软件。

[0068] 需要说明的是, 上述过程中, 当第二路径管理驱动 503 向多路径框架驱动 501 注册第二接口之后, 多路径框架驱动 501 通过修改新注册的第二接口的状态实现后续数据访问请求的处理, 多路径框架驱动 501 还可通过其他的管理规则实现后续数据访问请求的处理, 举例为, 可以通过设置时间段的方式, 即某一时刻调用某个固定的接口。

[0069] 步骤 250, 当所述第一数据访问请求处理完毕时, 卸载所述第一 PMD。

[0070] 参见图 6 所示的本发明实施例一提供的软件升级的方法示意图之四, 图 6 中, 在多路径框架驱动 601 重定向新业务流之后, 即通过新注册的第二接口向第二路径管理驱动 603 发送第二数据访问请求, 并由第二路径管理驱动 603 处理接收到的第二数据访问请求时, 应用服务器向第一路径管理驱动 602 发送注销命令, 由第一路径管理驱动 602 注销与多路径框架驱动 601 之间的第一接口。具体地, 在第一路径管理驱动 602 接收到注销命令之后, 判断其之前处理的第一数据访问请求是否全部处理完毕, 如果没有全部处理完毕, 则继续判断; 如果全部处理完毕, 则表示第一路径管理驱动 602 处理完成了所有的业务流, 然后第一路径管理驱动 602 注销与多路径框架驱动 601 之间的第一接口, 清理其在多路径框架驱动 601 中的内部数据, 并设置停工标记为已停工。应用服务器定期查询第一路径管理驱

动 602 的停工标记,当该停工标记为已停工时,表示第一数据访问请求已经全部处理完毕,卸载第一路径管理驱动 602,整个多路径管理软件升级完成。

[0071] 图 7 为本发明实施二提供的软件升级的方法流程图,该方法用于将第一路径管理驱动升级成第二路径管理驱动,如图 7 所示,本实施例具体包括以下步骤:

[0072] 步骤 710,当需要对所述多路径管理软件进行升级时,所述应用服务器继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求。

[0073] 参见图 8 所示的本发明实施例二提供的软件升级的方法示意图之一,图 8 中,多路径管理软件 80 包括两个驱动:多路径框架驱动 MFD801 和第一路径管理驱动 PMD802,对多路径管理软件 80 的升级即为对第一路径管理驱动 802 的升级。具体地,多路径框架驱动 801 定义一个运行时多路径内存数据管理的数据结构,它通过但不限于虚拟磁盘列表和管理的存储产品信息列表将多路径管理软件 80 运行时的全局内存数据统一管理起来。这样可以保证在多路径管理软件 80 包括多个路径管理驱动,且在升级其中一个路径管理驱动时,运行时全局公共内存数据仍然可用,其他路径管理驱动能够获取得到,这种多个路径管理驱动共享的同一套多路径内存数据可实现业务流的并行处理,还可以实现在不影响业务流的下发和返回的情况下实现将旧路径管理驱动升级成新路径管理驱动,此处,对业务流的处理通过对数据访问请求的处理来实现,而通常的数据访问请求即为 I/O 请求。此外,多路径框架驱动 301 还需定义一个管理 I/O 的数据结构,该结构包含指向系统 I/O 结构体的指针,还包含一个记录 I/O 处理的路径管理驱动的 HOOK 指针,该 HOOK 指针即为路径管理驱动对外提供的函数接口,这样的设计是为了保证在有多个路径管理驱动时,也即可以调用多个路径管理驱动同时处理 I/O 时,保证 I/O 的下发和返回都是同一个路径管理驱动处理,满足升级过程中, I/O 从升级前的路径管理驱动平滑过渡到升级后的路径管理驱动而业务流的转发不受影响的要求。具体地,当需要对所述多路径管理软件 80 进行升级时,应用服务器继续利用所述第一路径管理驱动处理升级之前接收到的第一数据访问请求,具体实现为,多路径框架驱动 801 定义一个管理 I/O 的数据结构,该数据结构中包含一个记录 I/O 处理的第一路径管理驱动 802 的 HOOK 指针,该 HOOK 指针即为第一路径管理驱动 802 对外提供的函数接口,记录为第一接口,通过第一路径管理驱动 802 对应的第一接口调用第一路径管理驱动 802 (该第一接口是由第一路径管理驱动 802 事先注册到多路径框架驱动 801 中),然后该第一路径管理驱动 802 选择相应的物理路径进行第一数据访问请求的处理,其中第一路径管理驱动 802 为升级前的路径管理驱动。

[0074] 需要说明的是,对于多路径管理软件 80 运行时所需的内存数据,多路径框架驱动 801 通过定义数据结构的方式对上述内存数据进行管理,可以将上述内存数据分为基本部分和扩展部分,基本部分包含对象基本信息等跨版本的公共成员,足以实现业务流处理过程中最简单的选路,这部分数据一直保持不变;扩展部分包存支撑复杂路径管理功能的,很有可能随版本升级而修改新增成员变量,为各版本路径管理驱动私有部分,由各路径管理驱动自己构造维护,适应变化。

[0075] 步骤 720,利用所述 MFD 处理后续接收到的第二数据访问请求。

[0076] 参见图 9 所示的本发明实施例二提供的软件升级的方法示意图之二,图 9 中,应用服务器先检查多路径框架驱动 901、第一路径管理驱动 902 和第二路径管理驱动 903 的兼容性,当符合兼容性要求时,利用多路径框架驱动 901 处理后续接收到的第二数据访问请求。

具体实现为,多路径框架驱动 901 管理 I/O 的数据结构中还包括一个指向系统 I/O 结构体的指针,通过该指针,多路径框架驱动 901 可以直接选择相应的物理路径,进行上述 I/O 的下发和返回。

[0077] 步骤 730,当所述第一数据访问请求处理完毕时,卸载所述第一 PMD。

[0078] 图 9 中,在利用所述多路径框架驱动 901 处理后续接收到的第二数据访问请求之后,应用服务器向第一路径管理驱动 902 发送注销命令,由第一路径管理驱动 902 注销与多路径框架驱动 901 之间的第一接口。具体地,在第一路径管理驱动 902 接收到注销命令之后,判断其之前处理的第一数据访问请求是否全部处理完毕,如果没有全部处理完毕,则继续判断;如果全部处理完毕,则表示第一路径管理驱动 902 处理完成了所有的业务流,然后第一路径管理驱动 902 注销与多路径框架驱动 901 之间的第一接口,清理其在多路径框架驱动 901 中的内部数据,并设置停工标记为已停工。应用服务器定期查询第一路径管理驱动 902 的停工标记,当该停工标记为已停工时,表示第一数据访问请求已经全部处理完毕,卸载第一路径管理驱动 902。

[0079] 步骤 730,获取所述第二 PMD。

[0080] 参见图 10 所示的本发明实施例二提供的软件升级的方法示意图之三,图 10 中,应用服务器获取第二路径管理驱动 1003。

[0081] 步骤 740,对所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将第三数据访问请求转发给所述第二 PMD,其中,所述第三数据访问请求为在对所述第二 PMD 进行注册之后,所述应用服务器接收到的数据访问请求。

[0082] 图 10 中,应用服务器在获取到第二路径管理驱动 1003 之后,将第二路径管理驱动 1003 加载到操作系统的内核中,并向第二路径管理驱动 1003 发送初始化指示信息;第二路径管理驱动 1003 接收到上述初始化指示信息之后,通过所述多路径框架驱动 1001 对外提供的接口获取多路径管理软件 100 运行时所需的全局内存数据,根据上述全局内存数据对自身进行初始化,完成初始化的第二路径管理驱动 1003 就具备了处理数据访问请求的能力,包括 I/O 的下发和返回;最后对第二路径管理驱动 1003 进行注册,具体地,第二路径管理驱动 1003 向路径框架驱动 1001 发送注册接口请求信息,路径框架驱动 1001 接收到请求信息后,可在其定义的管理 I/O 的数据结构中新增一个 I/O 处理的第二路径管理驱动 1003 的 HOOK 指针(记录为第二接口),之后多路径框架驱动 1001 通过第二路径管理驱动 1003 对应的第二接口,即可实现将第三数据访问请求转发给第二路径管理驱动 1003,其中,第三数据访问请求为在对所述第二路径管理驱动 1003 进行注册之后,应用服务器接收到的数据访问请求。图 10 中,多路径框架驱动 1001 和第二路径管理驱动 1003 同时在处理数据访问请求的方式,保证了不中断业务流转发的情况下,升级多路径管理软件。

[0083] 步骤 750,利用所述第二 PMD 接收所述 MFD 发送的所述第三数据访问请求,并处理所述第三数据访问请求。

[0084] 参见图 11 所示的本发明实施例二提供的软件升级的方法示意图之四,图 11 中,在对第二路径管理驱动 1103 进行注册之后,多路径框架驱动 1101 将新注册的第二接口标识为优先使用的状态,以实现当应用服务器接收到后续的第三数据访问请求时,多路径框架驱动 1101 通过标识为优先使用状态的第二接口向第二路径管理驱动 1103 发送第三数据访问请求,由第二路径管理驱动 1103 处理接收到的第三数据访问请求。

[0085] 需要说明的是,上述过程中,当第二路径管理驱动 1103 向多路径框架驱动 1101 注册第二接口之后,多路径框架驱动 1101 通过修改新注册的第二接口的状态实现后续数据访问请求的处理,多路径框架驱动 1101 还可通过其他的管理规则实现后续数据访问请求的处理,举例为,可以通过设置时间段的方式,即某一时刻调用某个固定的接口。

[0086] 图 12 为本发明实施例三提供的软件升级的装置示意图,所述装置可以用于执行图 2 所示的方法。图 12 中,该装置应用于存储系统中,所述存储系统包括应用服务器和存储设备,所述应用服务器和所述存储设备之间通过多条路径进行通信,其中所述应用服务器中包含多路径管理软件,所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD,所述装置用于将所述第一 PMD 升级成第二 PMD,所述装置包括:第一处理单元 1201、获取单元 1202、注册单元 1203、第二处理单元 1204 和卸载单元 1205。

[0087] 所述第一处理单元 1201,用于当需要对所述多路径管理软件进行升级时,继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求。

[0088] 多路径框架驱动定义一个管理 I/O 的数据结构,该数据结构中包含一个记录 I/O 处理的第一路径管理驱动的 HOOK 指针,该 HOOK 指针即为第一路径管理驱动对外提供的函数接口,记录为第一接口,通过第一路径管理驱动对应的第一接口调用第一路径管理驱动(该第一接口是由第一路径管理驱动事先注册到多路径框架驱动中),然后该第一路径管理驱动选择相应的物理路径进行第一数据访问请求的处理,其中第一路径管理驱动为升级前的路径管理驱动。

[0089] 所述获取单元 1202,用于获取所述第二 PMD。

[0090] 所述注册单元 1203,用于对所述获取单元 1202 获取的所述第二 PMD 进行注册,使得所述应用服务器利用所述 MFD 将后续接收到第二数据访问请求转发给所述第二 PMD。

[0091] 首先,第二路径管理驱动通过所述多路径框架驱动对外提供的接口获取多路径管理软件运行时所需的全局内存数据,根据上述全局内存数据对自身进行初始化,完成初始化的第二路径管理驱动就具备了处理数据访问请求的能力,包括 I/O 的下发和返回;然后对第二路径管理驱动进行注册,具体地,第二路径管理驱动向路径框架驱动发送注册接口请求信息,路径框架驱动接收到请求信息后,可在其定义的管理 I/O 的数据结构中新增一个 I/O 处理的第二路径管理驱动的 HOOK 指针(记录为第二接口),之后多路径框架驱动通过第二路径管理驱动对应的第二接口,即可实现将后续接收到的第二数据访问请求转发给第二路径管理驱动。

[0092] 所述第二处理单元 1204,用于利用所述注册单元注册后的所述第二 PMD 接收所述 MFD 发送的所述第二数据访问请求,并处理所述第二数据访问请求。

[0093] 在对第二路径管理驱动进行注册之后,多路径框架驱动将新注册的第二接口标识为优先使用的状态,以实现当应用服务器接收到后续的第二数据访问请求时,多路径框架驱动通过标识为优先使用状态的第二接口向第二路径管理驱动发送第二数据访问请求,由第二路径管理驱动处理接收到的第二数据访问请求。

[0094] 所述卸载单元 1205,用于当所述第一数据访问请求处理完毕时,卸载所述第一 PMD。

[0095] 可选地,所述装置还包括:发送单元 1206,用于向所述第二 PMD 发送初始化指示信息,使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据,

根据所述全局内存数据对所述第二 PMD 进行初始化。

[0096] 所述发送单元 1206 具体用于：对所述第一 PMD、所述 MFD 和所述第二 PMD 进行兼容性检查，当符合兼容性要求时，向所述第二 PMD 发送初始化指示信息，使得所述第二 PMD 通过所述 MFD 获取所述多路径管理软件运行时所需的全局内存数据。

[0097] 本发明实施例三提供的装置植入了本发明实施例一提供的方法，因此，本发明提供的装置的具体工作过程，在此不复赘述。

[0098] 图 13 为本发明实施例四提供的软件升级的装置示意图，所述装置可以用于执行图 7 所示的方法。图 13 中，该装置应用于存储系统中，所述存储系统包括应用服务器和存储设备，所述应用服务器和所述存储设备之间通过多条路径进行通信，其中所述应用服务器中包含多路径管理软件，所述多路径管理软件包含多路径框架驱动 MFD 和第一路径管理驱动 PMD，所述装置用于将所述第一 PMD 升级成第二 PMD，所述装置包括：第一处理单元 1301、第二处理单元 1302、卸载单元 1303、获取单元 1304、注册单元 1305 和第三处理单元 1306。

[0099] 所述第一处理单元 1301，用于当需要对所述多路径管理软件进行升级时，继续利用所述第一 PMD 处理升级之前接收到的第一数据访问请求。

[0100] 多路径框架驱动定义一个管理 I/O 的数据结构，该数据结构中包含一个记录 I/O 处理的第一路径管理驱动的 HOOK 指针，该 HOOK 指针即为第一路径管理驱动对外提供的函数接口，记录为第一接口，通过第一路径管理驱动对应的第一接口调用第一路径管理驱动（该第一接口是由第一路径管理驱动事先注册到多路径框架驱动中），然后该第一路径管理驱动选择相应的物理路径进行第一数据访问请求的处理，其中第一路径管理驱动为升级前的路径管理驱动。

[0101] 所述第二处理单元 1302，用于利用所述 MFD 处理后续接收到的第二数据访问请求。

[0102] 多路径框架驱动管理 I/O 的数据结构中还包含一个指向系统 I/O 结构体的指针，通过该指针，多路径框架驱动可以直接选择相应的物理路径，进行上述 I/O 的下发和返回。

[0103] 所述卸载单元 1303，当所述第一处理单元处理的第一数据访问请求处理完毕时，卸载所述第一 PMD。

[0104] 所述获取单元 1304，用于获取所述第二 PMD。

[0105] 所述注册单元 1305，用于对所述获取单元 1304 获取的所述第二 PMD 进行注册，使得所述应用服务器利用所述 MFD 将第三数据访问请求转发给所述第二 PMD，其中，所述第三数据访问请求为在对所述第二 PMD 进行注册之后，所述应用服务器接收到的数据访问请求。

[0106] 首先，第二路径管理驱动通过所述多路径框架驱动对外提供的接口获取多路径管理软件运行时所需的全局内存数据，根据上述全局内存数据对自身进行初始化，完成初始化的第二路径管理驱动就具备了处理数据访问请求的能力，包括 I/O 的下发和返回；然后对第二路径管理驱动进行注册，具体地，第二路径管理驱动向路径框架驱动发送注册接口请求信息，路径框架驱动接收到请求信息后，可在其定义的管理 I/O 的数据结构中新增一个 I/O 处理的第二路径管理驱动的 HOOK 指针（记录为第二接口），之后多路径框架驱动通过第二路径管理驱动对应的第二接口，即可实现将第三数据访问请求转发给第二路径管理驱动，其中，第三数据访问请求为在对所述第二路径管理驱动进行注册之后，应用服务器接收

到的数据访问请求。

[0107] 所述第三处理单元 1306, 利用所述注册单元 1305 注册的所述第二 PMD 接收所述 MFD 发送的所述第三数据访问请求, 并处理所述第三数据访问请求。

[0108] 在对第二路径管理驱动进行注册之后, 多路径框架驱动将新注册的第二接口标识为优先使用的状态, 以实现当应用服务器接收到后续的第三数据访问请求时, 多路径框架驱动通过标识为优先使用状态的第二接口向第二路径管理驱动发送第三数据访问请求, 由第二路径管理驱动处理接收到的第三数据访问请求。

[0109] 可见, 本发明实施例提供的软件升级的方法及装置, 可以保证在不中断数据访问请求处理的情况下, 升级多路径管理软件。

[0110] 专业人员应该还可以进一步意识到, 结合本文中所公开的实施例描述的各示例的单元及算法步骤, 能够以电子硬件、计算机软件或者二者的结合来实现, 为了清楚地说明硬件和软件的可互换性, 在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这些功能究竟以硬件还是软件方式来执行, 取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能, 但是这种实现不应认为超出本发明的范围。

[0111] 结合本文中所公开的实施例描述的方法或算法的步骤可以用硬件、处理器执行的软件模块, 或者二者的结合来实施。软件模块可以置于随机存储器 (RAM)、内存、只读存储器 (ROM)、电可编程 ROM、电可擦除可编程 ROM、寄存器、硬盘、可移动磁盘、CD-ROM、或技术领域内所公知的任意其它形式的存储介质中。

[0112] 以上所述的具体实施方式, 对本发明的目的、技术方案和有益效果进行了进一步详细说明, 所应理解的是, 以上所述仅为本发明的具体实施方式而已, 并不用于限定本发明的保护范围, 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, 均应包含在本发明的保护范围之内。

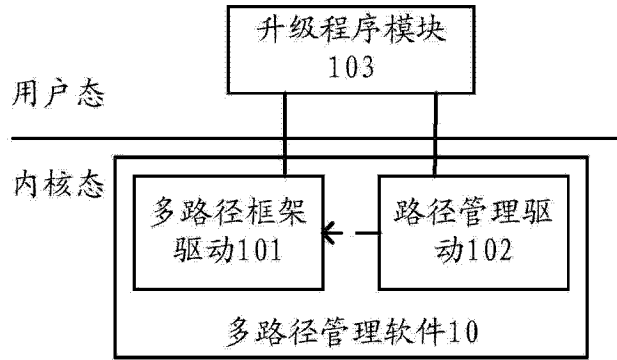


图 1

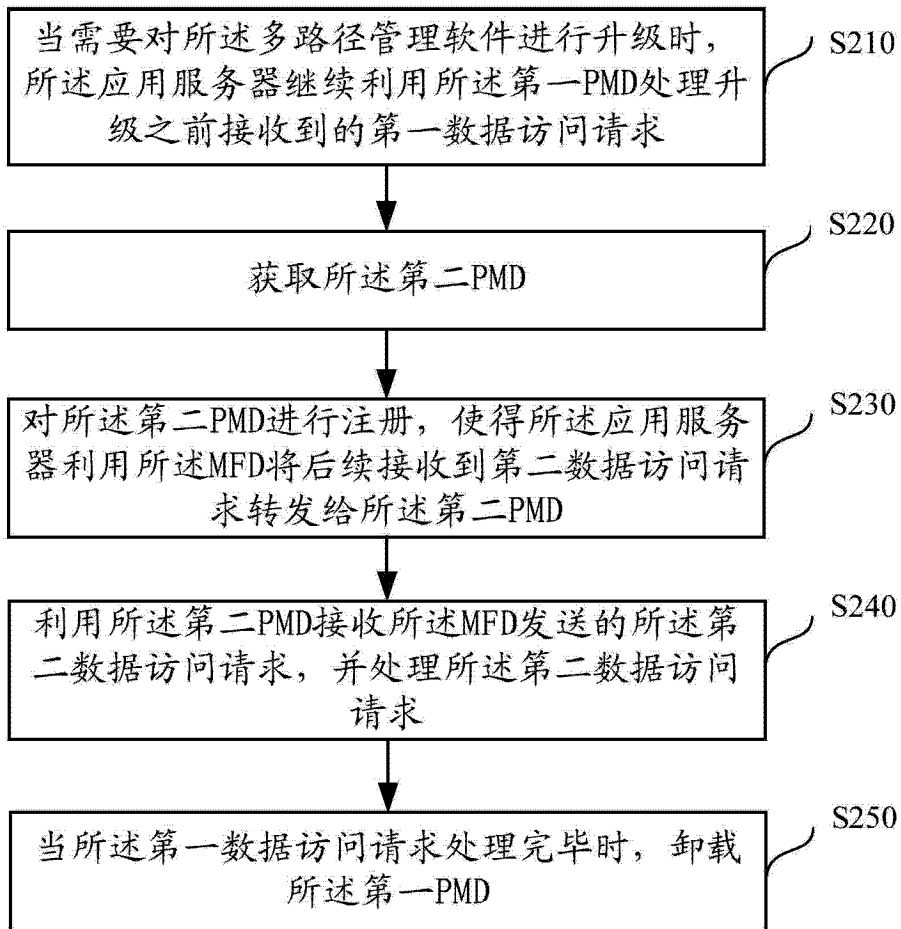


图 2

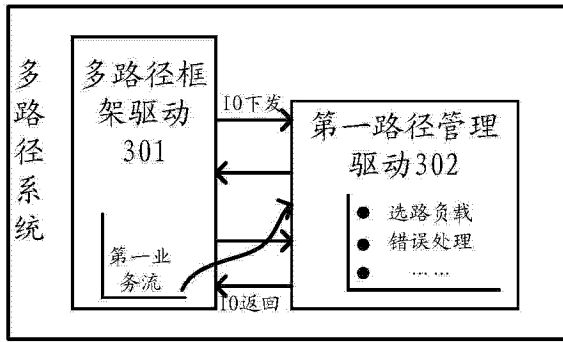


图 3

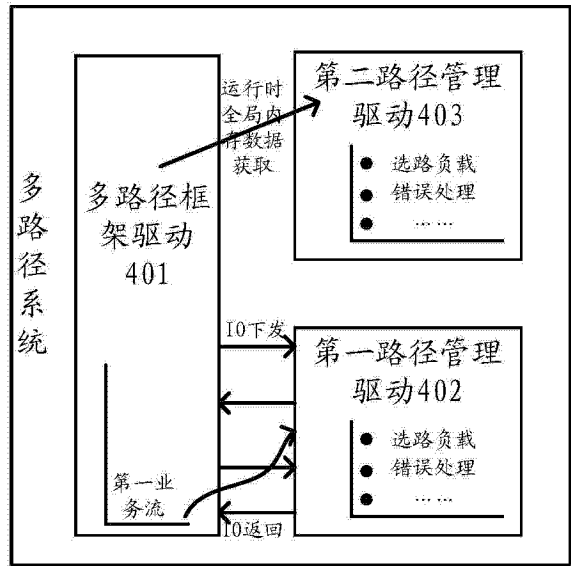


图 4

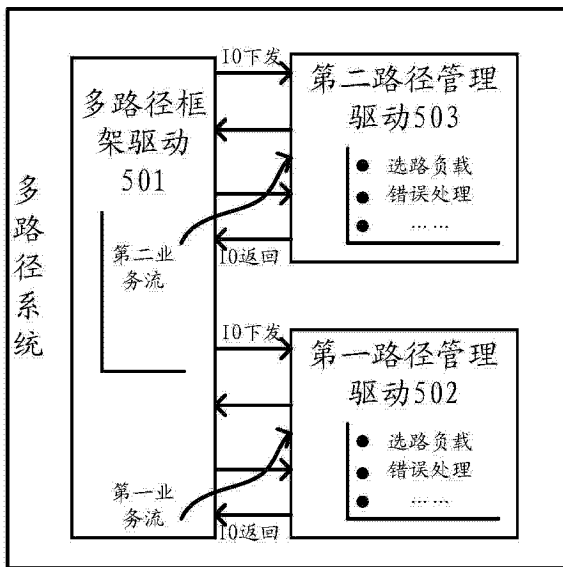


图 5

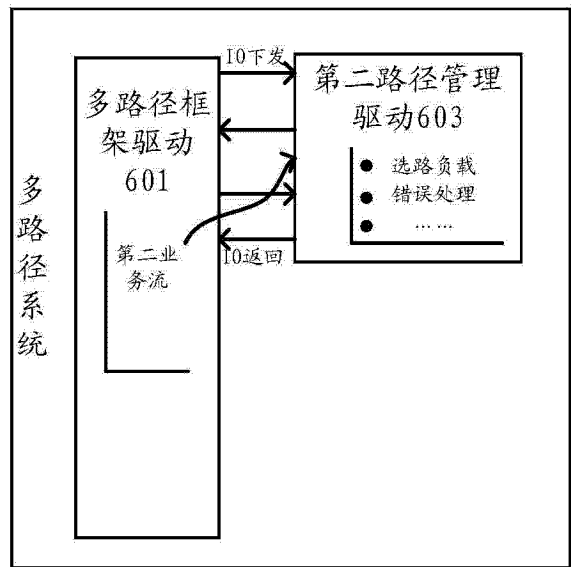


图 6

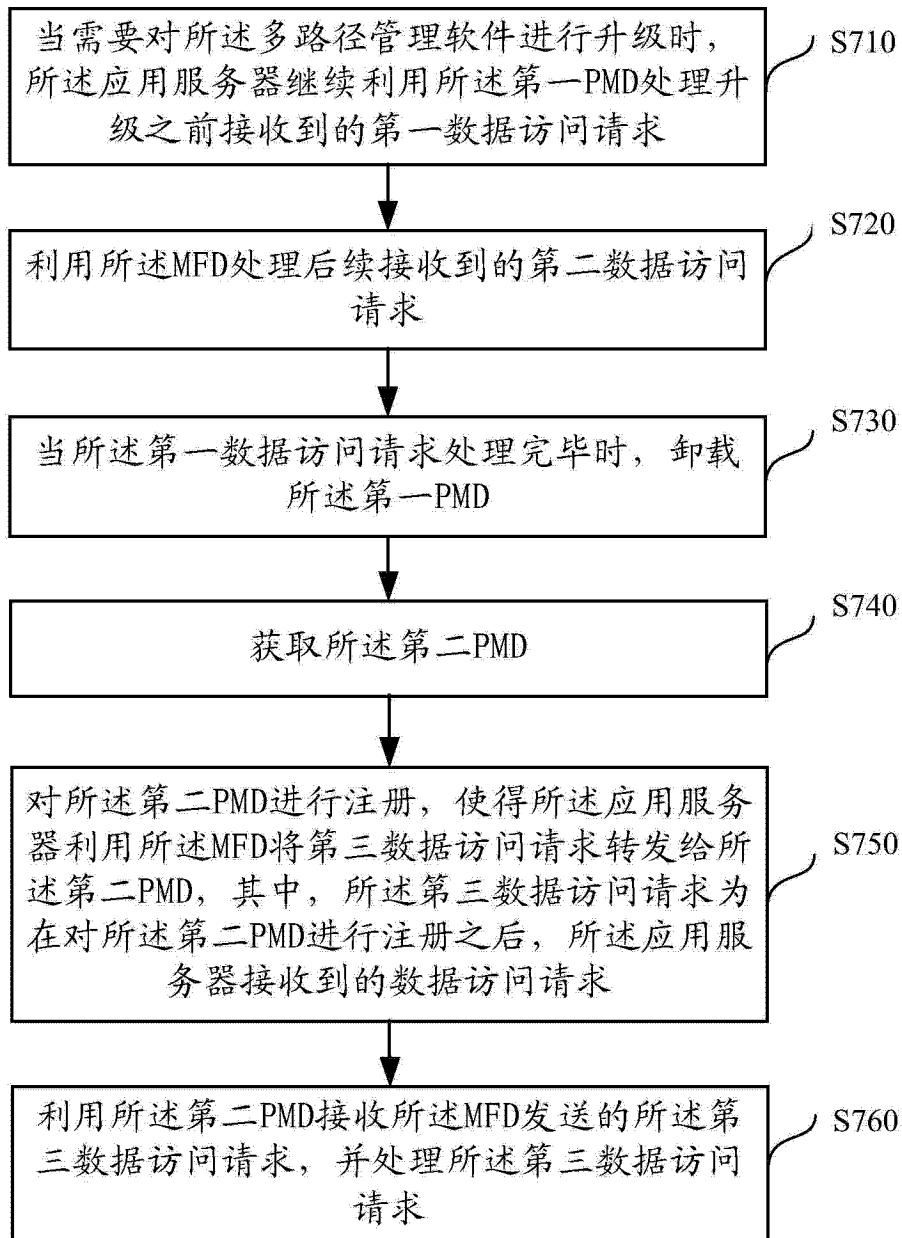


图 7

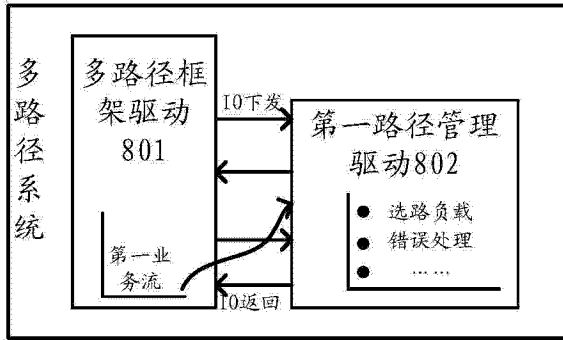


图 8

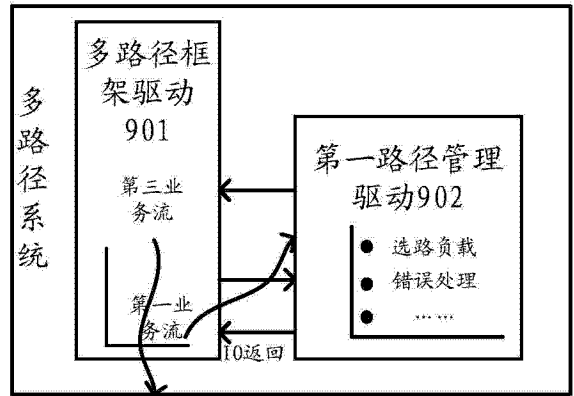


图 9

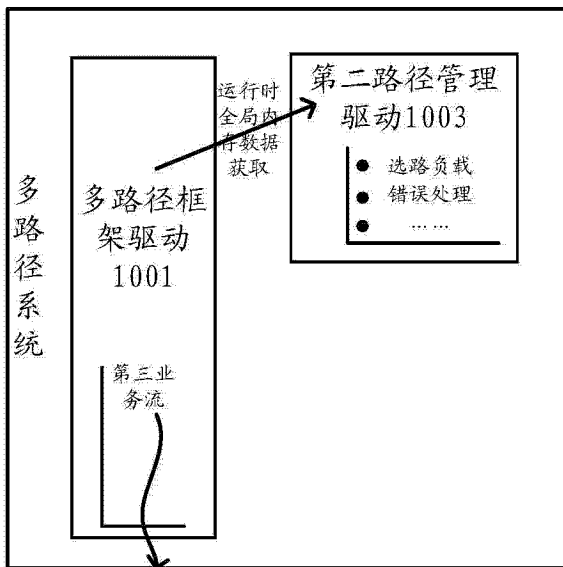


图 10

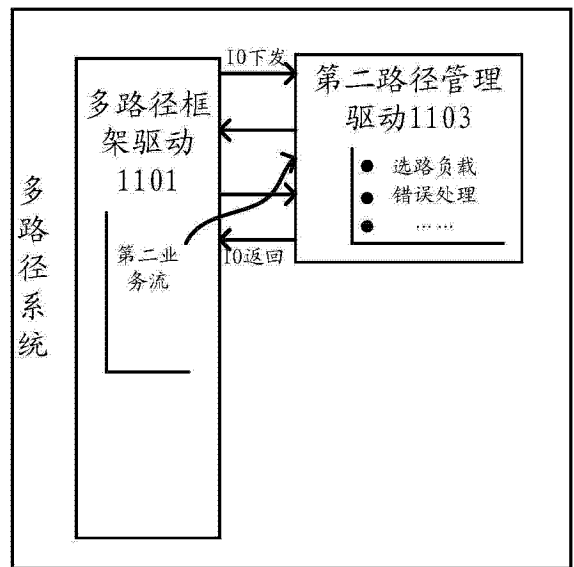


图 11

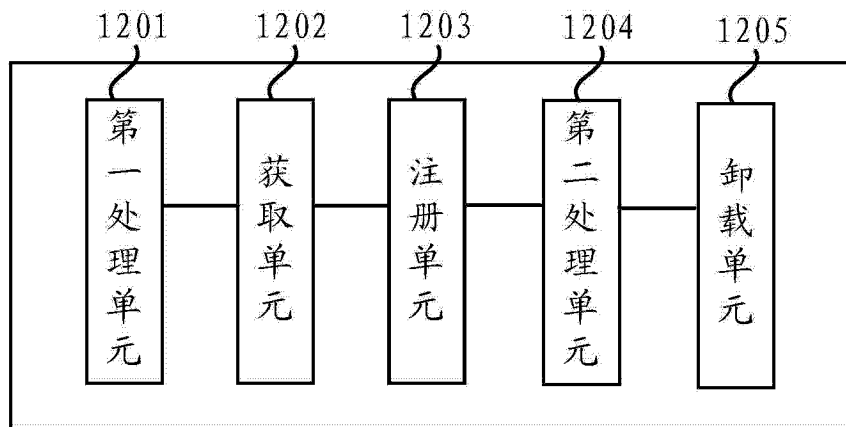


图 12

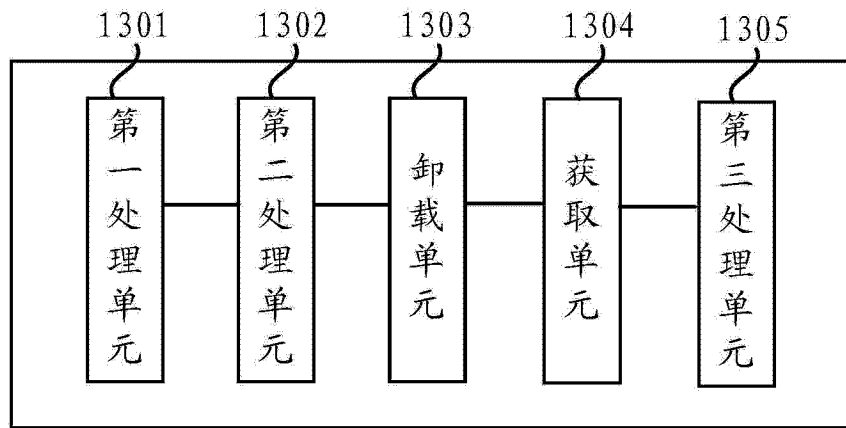


图 13