(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 119299518 B
(45)授权公告日 2025.05.02

(21)申请号 202411823360.1
(22)申请日 2024.12.10
(65)同一申请的已公布的文献号
申请公布号 CN 119299518 A
(43)申请公布日 2025.01.10
(73)专利权人 阿里云计算有限公司
地址 310024 浙江省杭州市西湖区三墩镇灯彩街1008号云谷园区1-2-A06室
(72)发明人 张世杰 周礼 丁宇
(74)专利代理机构 北京太合九思知识产权代理有限公司 11610
专利代理师 孙明子 刘戈
(51)Int.Cl.
H04L 67/568(2022.01)
H04L 67/1074(2022.01)
H04L 49/9047(2022.01)
H04L 69/163(2022.01)
(56)对比文件
CN 107241404 A,2017.10.10
审查员 王星

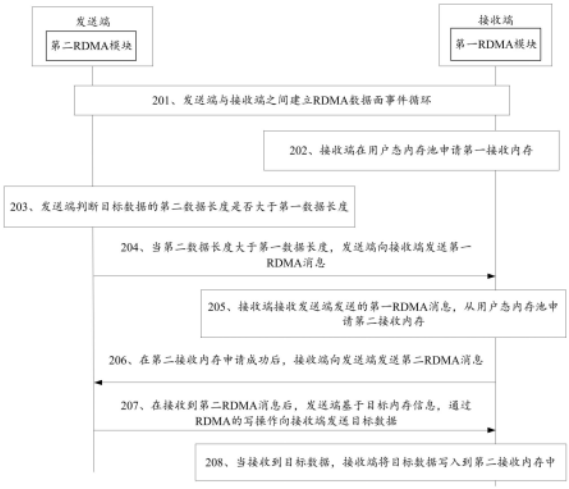
权利要求书4页 说明书13页 附图8页

(54)发明名称

基于RDMA的数据传输方法、装置、电子设备及介质

(57)摘要

本申请提供了一种基于RDMA的数据传输方法、装置、电子设备及介质,属于云计算领域。该方法包括:接收发送端发送的第一RDMA消息,第一RDMA消息为发送端在第一数据长度小于第二数据长度时发送,第一数据长度为接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,第一RDMA消息包括目标数据的第二数据长度;从用户态内存池申请最小数据长度为第二数据长度的第二接收内存;在第二接收内存申请成功后,向发送端发送第二RDMA消息,第二RDMA消息包括第二接收内存的目标内存信息;当接收到目标数据,将目标数据写入到第二接收内存中。本申请能够使得TCP应用程序适配RDMA,从而基于RDMA进行数据传输。



CN 119299518 B

1.一种基于远程直接内存访问RDMA的数据传输方法,其特征在于,所述方法应用于接收端,所述接收端配置有接收线程和选择线程,所述接收线程用于接收并处理多个发送端发送的连接请求,所述选择线程关联数据面事件循环线程和控制面事件循环线程,所述数据面事件循环线程和所述控制面事件循环线程用于所述接收端对发送端发送数据的事件进行监听,所述方法包括:

接收所述发送端发送的第一RDMA消息,所述第一RDMA消息为所述发送端在第一数据长度小于第二数据长度时发送,所述第一数据长度为所述接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,所述第一RDMA消息包括所述发送端本次待发送的目标数据的第二数据长度;

从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度;

在所述第二接收内存申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息,所述第二RDMA消息用于通知所述发送端基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据;

当接收到所述目标数据,将所述目标数据写入到所述第二接收内存中。

2.根据权利要求1所述的方法,其特征在于,所述选择线程对应一个数据面事件通道;

所述接收线程用于接收所述发送端发送的连接请求,对RDMA资源进行初始化操作,在完成初始化操作后,为所述发送端创建RDMA数据通道实例;

所述接收线程还用于将所述RDMA数据通道实例注册到所述选择线程中,并在注册过程中将所述RDMA数据通道实例与所述选择线程对应的数据面事件通道绑定;

所述选择线程用于对所述数据面事件通道进行监听,以获取所述RDMA数据通道实例的数据面事件,所述数据面事件是指所述接收端接收到所述发送端发送的数据的事件。

3.根据权利要求1所述的方法,其特征在于,所述数据面事件循环线程用于执行以下操作;

在开始一轮数据面事件循环后,如果所述接收端通过第一方法获取的数据面事件的数量为0,且任务队列为空,阻塞等待数据面事件通知;

在接收到数据面事件通知后被唤醒,处理所述数据面事件;

在处理完所述数据面事件之后,处理所述任务队列中的任务;

在处理完所述任务队列中的任务之后,开始新一轮的数据面事件循环。

4.根据权利要求3所述的方法,其特征在于,所述数据面事件循环线程还用于执行以下操作:

如果所述接收端通过所述第一方法访问获取到数据面事件的数量不为0,且所述任务队列非空,处理所述数据面事件;

在处理完所述数据面事件之后,处理所述任务队列中的任务;

在处理完所述任务队列中的任务之后,开始新一轮的数据面事件循环。

5.根据权利要求3所述的方法,其特征在于,所述控制面事件循环线程用于执行以下操作:

在开始一轮控制面事件循环后,通过第二方法阻塞获取控制面事件;

在获取到控制面事件之后,将获取到的控制面事件封装成任务,将封装的任务提交到

所述任务队列;

唤醒所述数据面事件循环线程,以使所述数据面事件循环线程处理数据面事件,并在处理完数据面事件之后,处理任务队列中的任务;

在处理完所述任务队列中的任务之后,开始新一轮的控制面事件循环。

6.根据权利要求5所述的方法,其特征在于,所述当接收到所述目标数据,将所述目标数据写入到所述第二接收内存,包括:

当通过所述第二方法监听到已接收到所述目标数据,确定获取到RDMA数据通道实例的数据面事件;

基于所述控制面事件循环线程将获取到的所述RDMA数据通道实例的数据面事件封装成任务,将所述任务加入到所述任务队列中,然后唤醒所述数据面事件循环线程;

通过执行所述数据面事件循环线程处理所述数据面事件;

在处理完所述数据面事件之后,处理所述任务队列中的任务,以将所述目标数据写入到所述第二接收内存。

7.根据权利要求5所述的方法,其特征在于,所述当接收到所述目标数据,将所述目标数据写入到所述第二接收内存,包括:

当通过所述第一方法监听到已接收到所述目标数据,确定获取到RDMA数据通道实例的数据面事件;

如果所述RDMA数据通道实例的数据面事件的数量不为0,且所述任务队列非空,通过执行所述数据面事件循环线程处理所述数据面事件;

在处理完所述数据面事件之后,处理所述任务队列中的任务,以将所述目标数据写入到所述第二接收内存。

8.根据权利要求1所述的方法,其特征在于,所述从所述用户态内存池申请第二接收内存,包括:

从所述用户态内存池中申请最小数据长度为所述第二数据长度的目标用户态内存;

获取所述目标用户态内存所属内存块的起始地址;

检查所述起始地址是否在RDMA内存注册表中,所述RDMA内存注册表中存储有已进行RDMA内存注册的内存块的起始地址与键值之间的对应关系;

当所述起始地址位于所述RDMA内存注册表中,从所述RDMA内存注册表中获取所述起始地址对应的目标键值;

将所述目标用户态内存作为所述第二接收内存,并将所述起始地址和目标键值作为所述目标内存信息,以完成所述第二接收内存的申请。

9.根据权利要求8所述的方法,其特征在于,所述方法还包括:

当所述起始地址未位于所述RDMA内存注册表中,对所述起始地址进行RDMA内存注册,将注册得到的目标键值以及所述起始地址写入到所述RDMA内存注册表中;

将所述目标用户态内存作为所述第二接收内存,并将所述起始地址和所述目标键值作为所述目标内存信息,以完成所述第二接收内存的申请。

10.根据权利要求9所述的方法,其特征在于,所述方法还包括:

如果对所述起始地址进行RDMA内存注册失败,按照访问时间由远及近的顺序,对所述RDMA内存注册表中已注册的各个内存块进行排序;

将所述RDMA内存注册表中位于排序结果前预设比例的内存块淘汰,然后再重新对所述起始地址进行RDMA内存注册。

11.根据权利要求9至10中任一项所述的方法,其特征在于,所述方法还包括:

每隔预设时间检查所述RDMA内存注册表;

如果所述RDMA内存注册表中任一内存块在预设时长内未被访问,将所述内存块淘汰。

12.一种基于远程直接内存访问RDMA的数据传输方法,其特征在于,所述方法应用于发送端,所述方法包括:

判断目标数据的第二数据长度是否大于第一数据长度,所述第一数据长度为接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,所述接收端配置有接收线程和选择线程,所述接收线程用于接收并处理多个发送端发送的连接请求,所述选择线程关联数据面事件循环线程和控制面事件循环线程,所述数据面事件循环线程和所述控制面事件循环线程用于所述接收端对发送端发送数据的事件进行监听;

当所述第二数据长度大于所述第一数据长度,向所述接收端发送第一RDMA消息,所述第一RDMA消息包括所述第二数据长度,所述第一RDMA消息用于通知所述接收端从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度,并在申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息;

在接收到所述第二RDMA消息后,基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据。

13.根据权利要求12所述的方法,其特征在于,所述方法还包括:

当所述第二数据长度不大于所述第一数据长度,通过RDMA的发送操作向所述接收端发送所述目标数据。

14.一种基于远程直接内存访问RDMA的数据传输装置,其特征在于,所述装置为接收端,所述接收端配置有接收线程和选择线程,所述接收线程用于接收并处理多个发送端发送的连接请求,所述选择线程关联数据面事件循环线程和控制面事件循环线程,所述数据面事件循环线程和所述控制面事件循环线程用于所述接收端对发送端发送数据的事件进行监听,所述装置包括:

接收模块,用于接收发送端发送的第一RDMA消息,所述第一RDMA消息为所述发送端在第一数据长度小于第二数据长度时发送,所述第一数据长度为所述接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,所述第一RDMA消息包括所述发送端本次待发送的目标数据的第二数据长度;

申请模块,用于从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度;

发送模块,用于在所述第二接收内存申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息,所述第二RDMA消息用于通知所述发送端基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据;

写入模块,用于当接收到所述目标数据,将所述目标数据写入到所述第二接收内存中。

15.一种基于远程直接内存访问RDMA的数据传输装置,其特征在于,所述装置为发送端,所述装置包括:

判断模块,用于判断目标数据的第二数据长度是否大于第一数据长度,所述第一数据长度为接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,所述接收端配置有接收线程和选择线程,所述接收线程用于接收并处理多个发送端发送的连接请求,所述选择线程关联数据面事件循环线程和控制面事件循环线程,所述数据面事件循环线程和所述控制面事件循环线程用于所述接收端对发送端发送数据的事件进行监听;

发送模块,用于当所述第二数据长度大于所述第一数据长度,向所述接收端发送第一RDMA消息,所述第一RDMA消息包括所述第二数据长度,所述第一RDMA消息用于通知所述接收端从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度,并在申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息;

所述发送模块,还用于在接收到所述第二RDMA消息后,基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据。

16.一种电子设备,其特征在于,包括处理器以及存储器;所述存储器存储至少一条程序代码;所述至少一条程序代码用于被所述处理器调用并执行,以实现如权利要求1至11中任一项所述的基于远程直接内存访问RDMA的数据传输方法,或权利要求12或13所述的基于RDMA的数据传输方法。

17.一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储有至少一条计算机程序,所述至少一条计算机程序被处理器执行时能够实现如权利要求1至11中任一项所述的基于远程直接内存访问RDMA的数据传输方法,或权利要求12或13所述的基于RDMA的数据传输方法。

18.一种计算机程序产品,其特征在于,所述计算机程序产品包括计算机程序,所述计算机程序被处理器执行时能够实现如权利要求1至11中任一项所述的基于远程直接内存访问RDMA的数据传输方法,或权利要求12或13所述的基于RDMA的数据传输方法。

基于RDMA的数据传输方法、装置、电子设备及介质

技术领域

[0001] 本申请涉及云计算技术领域,特别涉及一种基于RDMA的数据传输方法、装置、电子设备及介质。

背景技术

[0002] 目前,大多数应用程序都是基于TCP/IP(Transmission Control Protocol/Internet Protocol,传输控制协议/网际协议)开发的TCP应用程序。随着数据中心、分布式系统及云计算技术的发展,TCP应用程序对电子设备的传输性能要求越来越高。然而,受限于TCP/IP本身的局限性,在基于TCP/IP传输数据时,需要在用户态与内核态之间频繁地进行数据拷贝以及上下文切换等,导致TCP应用程序延迟较大、吞吐量低、CPU(Central Processing Unit)占用率较高,严重限制了TCP应用程序性能的提升。

[0003] RDMA(Remote Direct Memory Access,远程直接内存访问)作为一种直接存储器访问技术,能够使TCP应用程序绕过操作系统内核与CPU,直接与RDMA模块(比如网卡)进行通信,从而将TCP应用程序的数据从一个电子设备的用户态内存直接传输到另一个电子设备的用户态内存中。由于绕过了操作系统内核与CPU,因而能够避免频繁的数据拷贝、上下文切换带来的开销,真正地实现了低延迟、高吞吐量及低CPU占用率。

[0004] 然而,TCP/IP是基于字节流的数据传输协议,而RDMA是基于消息的数据传输协议,这使得TCP应用程序客户端对每次发送的数据长度没有限制,而RDMA模块对每次传输的数据长度有要求,因此,如何使得TCP应用程序适配RDMA,从而基于RDMA进行数据传输,成为当前亟需解决的问题。

发明内容

[0005] 本申请实施例提供了一种基于RDMA的数据传输方法、装置、电子设备及介质,能够使得TCP应用程序适配RDMA,从而基于RDMA进行数据传输。所述技术方案如下:

[0006] 第一方面,提供了一种基于RDMA的数据传输方法,所述方法应用于接收端,所述方法包括:

[0007] 接收发送端发送的第一RDMA消息,所述第一RDMA消息为所述发送端在第一数据长度小于第二数据长度时发送,所述第一数据长度为所述接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,所述第一RDMA消息包括所述发送端本次待发送的目标数据的第二数据长度;

[0008] 从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度;

[0009] 在所述第二接收内存申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息,所述第二RDMA消息用于通知所述发送端基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据;

[0010] 当接收到所述目标数据,将所述目标数据写入到所述第二接收内存中。

[0011] 第二方面,提供了一种基于RDMA的数据传输方法,所述方法应用于发送端,所述方法包括:

[0012] 判断目标数据的第二数据长度是否大于第一数据长度,所述第一数据长度为接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度;

[0013] 当所述第二数据长度大于所述第一数据长度,向所述接收端发送第一RDMA消息,所述第一RDMA消息包括所述第二数据长度,所述第一RDMA消息用于通知所述接收端从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度,并在申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息;

[0014] 在接收到所述第二RDMA消息后,基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据。

[0015] 第三方面,提供了一种基于远程直接内存访问RDMA的数据传输装置,所述装置为接收端,所述装置包括:

[0016] 接收模块,用于接收发送端发送的第一RDMA消息,所述第一RDMA消息为所述发送端在第一数据长度小于第二数据长度时发送,所述第一数据长度为所述接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,所述第一RDMA消息包括所述发送端本次待发送的目标数据的第二数据长度;

[0017] 申请模块,用于从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度;

[0018] 发送模块,用于在所述第二接收内存申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息,所述第二RDMA消息用于通知所述发送端基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据;

[0019] 写入模块,用于当接收到所述目标数据,将所述目标数据写入到所述第二接收内存中。

[0020] 第四方面,提供了一种基于远程直接内存访问RDMA的数据传输装置,所述装置为发送端,所述装置包括:

[0021] 判断模块,用于判断目标数据的第二数据长度是否大于第一数据长度,所述第一数据长度为接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度;

[0022] 发送模块,用于当所述第二数据长度大于所述第一数据长度,向所述接收端发送第一RDMA消息,所述第一RDMA消息包括所述第二数据长度,所述第一RDMA消息用于通知所述接收端从所述用户态内存池申请第二接收内存,所述第二接收内存所能存储数据的最小数据长度为所述第二数据长度,并在申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息;

[0023] 所述发送模块,还用于在接收到所述第二RDMA消息后,基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据。

[0024] 第五方面,提供了一种电子设备,包括处理器以及存储器;所述存储器存储至少一条程序代码;所述至少一条程序代码用于被所述处理器调用并执行,以实现第一方面所述的基于RDMA的数据传输方法,或第二方面所述的基于RDMA的数据传输方法。

[0025] 第六方面,提供了一种计算机可读存储介质,所述计算机可读存储介质中存储有至少一条计算机程序,所述至少一条计算机程序被处理器执行时能够实现第一方面所述的基于RDMA的数据传输方法,或第二方面所述的基于RDMA的数据传输方法。

[0026] 第七方面,提供了一种计算机程序产品,所述计算机程序产品包括计算机程序,所述计算机程序被处理器执行时能够实现第一方面所述的基于RDMA的数据传输方法,或第二方面所述的基于RDMA的数据传输方法。

[0027] 本申请实施例提供的技术方案带来的有益效果是:

[0028] 由于TCP协议并不关注每次发送数据的数据长度,因而作为发送端的TCP应用程序客户端每次向接收端发送的数据长度并不是确定的,而RDMA协议要求接收端预先准备接收内存,而接收内存的大小是确定的,这样发送端每次发送数据的数据长度可能大于接收端预先准备的接收内存的最大数据长度,也可能小于接收端预先准备的接收内存的最大数据长度,为使TCP应用程序能够适配RDMA,发送端在基于RDMA向接收端发送数据之前,需要判断本次待发送的目标数据的第二数据长度是否大于接收端预先申请的第一接收内存所能存储数据的第一数据长度,如果第二数据长度不超过第一数据长度,也即是,第一接收内存能够存储下目标数据,则可通过RDMA的发送操作将目标数据发送给接收端,接收端在接收到目标数据之后,将目标数据写入到第一接收内存中,从而实现基于RDMA的数据传输;如果第二数据长度超过第一数据长度,也即是,第一接收内存无法存储下目标数据,发送端向接收端发送第一RDMA消息,以通知接收端重新从用户态内存池中申请一块新的内存,即第二接收内存,接收端申请成功后,向发送端发送携带第二接收内存的目标内存信息的第二RDMA消息,发送端接收到第二RDMA消息后,基于目标内存信息,通过RDMA的写操作将目标数据发送给接收端,接收端在接收到目标数据之后,将目标数据写入到第二接收内存中,从而实现基于RDMA的数据传输。

附图说明

[0029] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例描述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申请的一些实施例,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。

[0030] 图1是相关技术提供的一种SMC-R的系统架构图;

[0031] 图2是本申请实施例提供的一种基于RDMA的数据传输方法的流程图;

[0032] 图3是本申请实施例提供的一种发送端和接收端之间的RDMA数据面事件循环建立过程的示意图;

[0033] 图4是本申请实施例提供的一种数据面事件循环线程和控制面事件循环线程的工作过程示意图;

[0034] 图5是本申请实施例提供的一种RDMA内存注册过程的流程图;

[0035] 图6是本申请实施例提供的另一种基于RDMA的数据传输方法的流程图;

[0036] 图7是本申请实施例提供的一种基于RDMA的数据传输装置的结构示意图;

[0037] 图8是本申请实施例提供的另一种基于RDMA的数据传输装置的结构示意图;

[0038] 图9示出了本申请一个示例性实施例提供的一种电子设备的结构框图。

具体实施方式

[0039] 为使本申请的目的、技术方案和优点更加清楚,下面将结合附图对本申请实施方式作进一步地详细描述。

[0040] 可以理解,本申请实施例所使用的术语“每个”、“多个”及“任一”等,多个包括两个或两个以上,每个是指对应的多个中的每一个,任一是指对应的多个中的任意一个。举例来说,多个词语包括10个词语,而每个词语是指这10个词语中的每一个词语,任一词语是指10个词语中的任意一个词语。

[0041] 需要说明的是,本申请所涉及的用户信息(包括但不限于用户设备信息、用户个人信息等)和数据(包括但不限于用于分析的数据、存储的数据、展示的数据等),均为经用户授权或者经过各方充分授权的信息和数据,并且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准,并提供有相应的操作入口,供用户选择授权或者拒绝。

[0042] 在执行本申请实施例之前,首先对本申请涉及的名词进行解释。

[0043] RDMA是为了解决网络传输中服务端(即接收端)数据处理的延迟而产生的。RDMA通过网络把TCP应用程序的数据直接写入到电子设备的存储区,然后将数据从一个系统快速移动到远程系统存储器中,而不对操作系统造成任何影响。RDMA消除了外部存储器复制和上下文切换的开销,因而能解放内存带宽和CPU周期,提升了系统的性能。

[0044] RDMA内存注册:接收端在接收数据之前,需要进行内存注册(Memory Register,MR),每个内存注册时都会得到一个远程的Key和一个本地的Key(r_key,l_key)。本地Key被本地的主机通道适配器(Host Channel Adapter,HCA)用来访问本地内存。远程Key提供给远程HCA,用来在RDMA操作期间允许远程进程访问本地的系统内存。

[0045] IB verbs是InfiniBand标准中定义的RDMA编程API(Application Programming Interface,TCP应用程序编程接口)规范,为TCP应用程序提供了直接访问网络硬件的能力,从而实现高效、低延迟的数据传输。本申请所涉及的方法比如ibv_post_recv()、ibv_post_send()、ibv_poll_cq()、ibv_get_cq_event()、rdma_get_cm_event()等均来自该规范中的API定义。

[0046] RDMA send/recv:是RDMA中的双端操作,完成一次通信过程需要两端CPU的参与。接收端首先通过ibv_post_recv()准备接收内存,然后发送端通过ibv_post_send()发送数据。

[0047] RDMA write:接收端预先准备一块内存,并进行RDMA内存注册,得到一个Key(即远程Key),然后将该Key以及内存信息返回给发送端,发送端携带该Key即可对该内存进行写操作,而不需要接收端的CPU的参与。

[0048] 随着数据中心、分布式系统、高性能计算领域的快速发展,网络设备性能显著提升。然而,网络设备性能提升的同时,网络性能与CPU算力的失配问题逐渐显露。传统TCP/IP网络中,CPU不仅负责网络报文的封装、解析,还负责在用户态与内核态之间搬运数据,随着网络带宽的增加,CPU的算力面临越来越大的压力。以TCP/IP网络的一次数据发送与接收过程为例,发送端的CPU先将数据从用户态内存拷贝至内核态内存,在内核态协议栈中完成数据包封装,再由DMA(Direct Memory Access,直接存储器访问)控制器将封装好的数据包搬运到NIC(Network Interface Card,网络接口板)上发送给接收端的NIC。接收端的NIC接收

到数据包之后,通过DMA控制器将该数据包搬运到内核态内存中,由内核协议栈解析,层层剥离帧首或包头,然后由CPU将有效负载(payload)拷贝到用户态内存中,完成一次数据传输。

[0049] 在一次数据传输过程中,CPU需要负责用户态与内核态间的数据拷贝以及网络报文的封装、解析工作。这些工作占用了大量CPU资源,使得CPU在数据密集型场景下无法将算力用到更有益的地方。因此,解决网络性能与CPU算力失配问题成为了高性能网络发展的关键。考虑到摩尔定律逐渐失效,CPU性能短时间内发展缓慢,将网络数据处理工作从CPU卸载到硬件设备已成为主流解决方案。

[0050] 在RDMA网络中,具备RDMA能力的网卡RNIC能够从发送端的用户态内存中直接获取数据,完成数据封装后再传输到接收端,接收端RNIC接收到数据后,将接收到的数据解析剥离,将有效负载(payload)直接放入用户态内存中完成数据传输。这一过程中CPU除了必要的控制面功能外,几乎不用参与数据传输,数据就像是通过RNIC直接写入到远程节点的内存中一样。因此,与传统网络相比,RDMA将CPU从网络传输中解放了出来,使得网络传输就像是远程内存直接访问一样方便快捷。

[0051] 目前业界提供了SMC-R(Shared Memory Communication over RDMA)的无侵入式技术。参见图1,SMC-R工作于内核空间,向上支持用户态程序通过Socket接口描述的网络行为,向下使用IBverbs接口。RDMA资源的使用、管理与维护均可由SMC-R协议栈完成,TCP应用程序不会感知到内核中的RDMA实体,使得RDMA网络传输对TCP应用程序透明无侵入的替换,提供了高性能的软硬件协同网络。但是,SMC-R工作在内核态,不能充分发挥RDMA内核旁路的特性,且SMR-R为了兼容socket接口,无法实现零拷贝。

[0052] 为真正地实现内核旁路和数据零拷贝,本申请通过对TCP应用程序的通信层代码进行了改动,摒弃了TCP的Socket接口,而是直接使用RDMA的IBverbs接口,由于无需为了兼容Socket而付出代价,因而降低了资源消耗。另外,IBverbs接口中的数据面方法工作在用户态,能够真正地做到内核旁路,大量减少了上下文切换。另外,通过对TCP应用程序的用户态内存池进行RDMA内存注册,从而能够在基于RDMA传输数据时直接对用户态内存池中注册的内存进行读写,通过无侵入的RDMA内存池增强,不仅实现了数据传输过程的零拷贝,而且无需重复开发,降低了开发成本。

[0053] 本申请实施例提供了一种基于RDMA的数据传输方法,以发送端和接收端执行本申请实施例为例,参见图2,本申请实施例提供的方法流程包括:

[0054] 201、发送端与接收端之间建立RDMA数据面事件循环。

[0055] 其中,发送端和接收端均为TCP应用程序的客户端。为实现基于RDMA的通信,发送端和接收端中均配置有RDMA模块,该RDMA模块能够实现RDMA协议相关的接口,可以为RDMA硬件,也可以为RDMA软件。该RDMA硬件可以是具有RDMA功能的网卡(比如RNIC),该RDMA软件可以为具有RDMA功能的软件代码。为便于区分发送端所在电子设备中配置的RDMA模块,以及接收端所在电子设备中配置的RDMA模块,可以将接收端对应的RDMA模块称为第一RDMA模块,将发送端对应的RDMA模块称为第二RDMA模块。

[0056] 当发送端所在的电子设备开机时,会在初始化阶段为发送端配置多个线程,以便于后续发送端发送数据,该多个线程包括选择线程(selector线程)以及服务线程等。当接收端所在的电子设备开始时,也会在初始化阶段,为接收端配置多个线程,以便于后续接收

端接收数据,该多个线程包括接收线程(acceptor线程)、选择线程及服务线程等。当发送端想要基于RDMA向接收端发送数据时,发送端可以通过第二RDMA模块向接收端发送连接请求。由于RDMA是基于消息进行通信的,为了更好对接收端通过第一RDMA模块接收到的、发送端发送数据的事件进行监听,从而及时对接收到的数据进行读写,接收端在接收到发送端发送的连接请求之后,需要建立发送端与接收端之间的RDMA数据面事件循环(之所以建立的是事件循环,是因为发送端和接收端之间所建立的链接属于长链接,建立后可以多次进行数据的收发)。接收端在建立与发送端的RDMA数据面事件循环时,可以借助初始化阶段所配置的接收线程和选择线程进行建立。

[0057] 其中,接收端的接收线程可以接收并处理多个发送端发送的连接请求,与现有方案中接收端侧的接收线程只能接收并处理一个连接请求相比,接收线程的利用率更高。接收端的选择线程可以为一个线程池,该线程池包括多个选择线程,比如图3中所示的多个RdmaEventLoop。每个选择线程可以对应一个唯一的数据面事件通道,该数据面事件通道为接收端通过第一RDMA模块接收数据的一个数据通道,数据面事件是指接收端通过第一RDMA模块接收发送端发送数据的事件。具体地,该接收线程用于接收发送端发送的连接请求,响应于该连接请求,对第一RDMA模块相关的RDMA资源(比如队列资源)进行初始化操作,在完成初始化操作后,为发送端创建RDMA数据通道实例,然后从选择线程池中选择一个空闲未注册过的选择线程,进而将RDMA数据通道实例注册到该选择线程中,在注册过程中可以将所创建的RDMA数据通道实例与该选择线程对应的数据面事件通道绑定。选择线程用于对所绑定的数据面事件通道进行监听,以获取RDMA数据通道实例的数据面事件。

[0058] 图3示出了发送端和接收端之间的RDMA数据面事件循环建立过程,参见图3,多个客户端(即发送端)可以向服务端(即接收端)发送连接请求,服务端的RdmaEventLoop(接收线程)接收多个客户端发送的连接请求,并对每个客户端对应的RDMA资源进行初始化,在完成初始化操作后,为每个客户端创建RdmaChannel(RDMA数据通道实例),然后从选择线程池中为每个客户端选择一个空闲未注册的RdmaEventLoop(选择线程),进而将为每个客户端创建的RdmaChannel注册到为其选择的RdmaEventLoop(选择线程)中,在注册过程中,将每个客户端对应的RdmaChannel与所注册的RdmaEventLoop(选择线程)对应的数据面事件通道进行绑定,从而可以通过监听每个数据面事件通道,获取所绑定的RDMA数据通道实例的数据面事件,并在获取到所绑定的RDMA数据通道实例的数据面事件后,调用服务线程中的相应的事件处理器对获取到的事件进行处理。

[0059] 在本申请中,每个选择线程都会关联两个线程,一个线程是数据面事件循环线程,可以表示为cq_thread,该cq_thread负责数据面事件循环;另一个线程是控制面事件循环线程,可以表示为cm_thread,该cm_thread负责控制面事件循环。控制面事件循环线程在获取到控制面事件后并不会处理,而是提交给数据面事件循环线程进行处理,从而提供了一个单线程封闭的环境,保障了线程的安全,处理过程无锁化。基于数据面事件循环线程和控制面事件循环线程,可以分别实现RDMA数据面事件循环和RDMA控制面事件循环。下面将分别介绍这两个循环。

[0060] RDMA数据面事件循环

[0061] 在开始一轮数据面事件循环后,接收端可以通过第一方法访问第一RDMA模块,从而非阻塞获取第一RDMA模块接收发送端发送数据的数据面事件,该第一方法可以为ibv_

poll_cq方法等,如果接收端通过第一方法访问第一RDMA模块获取的数据面事件的数量为0,检测任务队列(该任务队列为数据面事件循环线程所拥有的一个队列,用于处理非输入输出任务)是否为空,如果该任务队列为空,则阻塞等待数据面事件通知,在阻塞等待数据面事件通知时,可以使用第三方法来获取,该第三方法可以为ibv_get_cq_event等。在接收到数据面事件通知后,唤醒数据面事件循环线程,然后调用该数据面事件循环线程处理数据面事件,并在处理完数据面事件之后,处理任务队列中的任务,在处理完任务队列中的任务之后,开始新一轮的数据面事件循环。如果接收端通过第一方法访问第一RDMA模块获取到数据面事件的数量不为0,且任务队列非空,此时数据面事件循环线程处于工作状态,可以调用该数据面事件循环线程处理数据面事件,并在处理完数据面事件之后,处理任务队列中的任务,在处理完任务队列中的任务之后,开始新一轮的数据面事件循环。

[0062] 图4示出了数据面事件循环线程的工作过程,参见图4,开始一轮数据面事件循环后,接收端通过ibv_poll_cq方法非阻塞地获取数据面事件,并获取数据面事件的数量pollNum,如果pollNum==0并且任务队列为空,则通过ibv_get_cq_event方法阻塞等待数据面事件通知,在接收到数据面事件通知后,唤醒cq_thread线程,然后开始新一轮的数据面事件循环,以通过cq_thread线程处理任务队列中的任务;如果条件pollNum并不等于0且任务队列非空,则先处理数据面事件(如果有),再处理任务队列中的任务(如果有),然后开始新一轮的数据面事件循环。

[0063] RDMA控制面事件循环

[0064] 在开始一轮控制面事件循环后,控制面事件循环线程通过第二方法阻塞获取控制面事件,该第二方法可以为rdma_get_cm_event()等。在获取到控制面事件之后,控制面事件循环线程将获取到的控制面事件封装成任务,并将封装的任务提交到数据面事件循环线程对应的任务队列,然后唤醒该数据面事件循环线程,以使数据面事件循环线程数据面事件,并在处理完数据面事件之后,处理任务队列中的任务,并处理完任务队列中的任务之后,开始新一轮的控制面事件循环。

[0065] 图4示出了控制面事件循环线程的工作过程,参见图4,在开始一轮数据面事件循环后,cm_thread通过rdma_get_cm_event()阻塞获取控制面事件,在获取到控制面事件后,将控制面事件处理封装成任务,并将封装的任务提交到cq_thread对应的任务队列,然后唤醒cq_thread,以便cq_thread及时处理数据面事件,并在处理完数据面事件之后,处理任务队列中的任务,然后再开始新一轮的控制面事件循环。

[0066] 通过在发送端与接收端之间构建RDMA数据面事件循环,可以对接收端发送数据的事件进行监听,从而对接收到的数据进行即时处理。

[0067] 202、接收端在用户态内存池申请第一接收内存。

[0068] 通常,TCP应用程序配置有用户态内存池,当需要使用内存时,TCP应用程序会向操作系统申请一块内存(也被称为chunk),所申请的内存块可有由用户态内存池自行管理。RDMA模块在内存使用上与TCP应用程序不同,无法直接使用TCP的用户态内存池。相关技术通过侵入式地改造TCP的用户态内存池,或为RDMA定制一个用户态内存池,可以使得RDMA能够使用内存。但是,这些方法不仅会增加后续代码的维护成本,而且会增加成本。为此,本申请实施例提供一种无侵入的RDMA内存池增强组件,基于该组件通过RDMA内存注册,可以将TCP的用户态内存池复用于RDMA,而无需重复开发。其中,RDMA内存注册时需要指定内存的

起始地址addr和长度len,注册成功后为所申请的内存分配唯一的Key,后续RDMA模块携带该Key才能对这块内存进行读/写操作。

[0069] 由于发送端每次发送数据的数据长度是不确定的,为避免内存资源的浪费,接收端在发送端发送数据之前,可以预先申请数据长度为第二数据长度的第一接收内存。接收端从用户态内存池中申请第一接收内存的过程,具体包括:接收端从用户态内存池中申请数据长度为第二数据长度的用户态内存(在将该用户态内存提交给第一RDMA模块之前,接收端可以对该用户态内存进行数据写入),然后以该用户态内存所属内存块作为RDMA内存注册的最小单位进行注册。在进行注册时,获取该用户态内存所属内存块的起始地址,接着,检查该起始地址是否在RDMA内存注册表中,该RDMA内存注册表中存储有已进行RDMA内存注册的内存块的起始地址与键值之间的对应关系,如果该起始地址位于RDMA内存注册表中,则返回该起始地址对应的键值,进而通过ibv_post_recv方法将第一接收内存的内存信息提供给第一RDMA模块,以便于第一RDMA模块对该内存块进行读写,其中,第一接收内存的内存信息包括内存块的起始地址、内存块的长度、键值等;如果该起始地址未位于RDMA内存注册表中,则在第一RDMA模块中,对该起始地址进行注册,将注册得到的键值和该起始地址写入RDMA内存注册表中,进而通过ibv_post_recv方法将第一接收内存的内存信息提供给第一RDMA模块,以便于第一RDMA模块对该内存块进行读写。

[0070] 此处需要说明一点,接收端与发送端只需预先约定第一接收内存的第一数据长度,不需要约定内存地址等信息,在约定好第一接收内存的第一数据长度之后,发送端在发送数据之前,判断出接收端预先准备的第一接收内存的第一数据长度大于待发送的目标数据的第二数据长度,则将目标数据通过第二RDMA模块发送给接收端的第一RDMA模块,第一RDMA模块直接将目标数据写入到第一接收内存中。

[0071] 203、发送端判断目标数据的第二数据长度是否大于第一数据长度。

[0072] 其中,第一数据长度为接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度。发送端在向接收端发送目标数据之前,可以判断目标数据的第二数据长度是否大于第一数据长度,如果第二数据长度大于第一数据长度,则执行步骤204;如果第二数据长度不大于第一数据长度,则接收端可以通过第二RDMA模块的发送操作(RDMA send操作)向接收端发送目标数据。

[0073] 204、当第二数据长度大于第一数据长度,发送端向接收端发送第一RDMA消息。

[0074] 其中,第一RDMA消息包括第二数据长度等。

[0075] 205、接收端接收发送端发送的第一RDMA消息,从用户态内存池申请第二接收内存。

[0076] 其中,第二接收内存所能存储数据的最小数据长度为第二数据长度。接收端从用户态内存池申请第二接收内存时,具体包括:接收端从用户态内存池中申请最小数据长度为第二数据长度的目标用户态内存,进而获取目标用户态内存所属内存块的起始地址,然后检查起始地址是否在RDMA内存注册表中,当起始地址位于RDMA内存注册表中,从RDMA内存注册表中获取起始地址对应的目标键值,进而将目标用户态内存作为第二接收内存,并将起始地址和目标键值等作为目标内存信息提供给第一RDMA模块,以完成第二接收内存的申请;当起始地址未位于RDMA内存注册表中,对起始地址进行RDMA内存注册,将注册得到的目标键值以及起始地址写入到RDMA内存注册表中,进而将目标用户态内存作为第二接收内

存,并将起始地址和目标键值等作为目标内存信息提供给第一RDMA模块,以完成第二接收内存的申请。

[0077] 图5示出了接收端从用户态内存中申请接收内存的过程,参见图5,接收端从TCP内存池(即TCP的用户态内存)申请一块内存buf,然后获取该内存buf所属内存块的起始地址addr,然后检查RDMA注册表中是否存储该起始地址addr,如果RDMA注册表中存储有该起始地址addr,则返回该起始地址addr对应的key,进而将该内存buf和key等内存信息提交给RDMA模块中,以便于该RDMA模块可以对该内存进行读写;如果RDMA注册表中未存储该起始地址addr,则对该起始地址addr进行RDMA内存注册,并在注册后返回key,进而将起始地址addr和key写入到RDMA注册表中,然后将该内存buf和key等内存信息提交给RDMA模块中,以便于该RDMA模块可以对该内存进行读写。

[0078] 进一步地,本申请提供了一种淘汰机制,基于该淘汰机制对注册的内存块进行RDMA内存反注册,从而将其从RDMA内存注册表中删除,以提高内存注册的成功率以内存资源的利用率。该淘汰机制包括但不限于如下两种情况:

[0079] 第一种情况、考虑到RDMA模块对进行内存注册的内存块的数量是有限制的,如果接收端注册的内存块超过限制的数量后,内存注册将会失败,因此,当对起始地址进行RDMA内存注册失败后,接收端可以按照访问时间由远及近的顺序,对RDMA内存注册表中已注册的各个内存块进行排序,然后将RDMA内存注册表中位于排序结果前预设比例的内存块淘汰,然后再重新对起始地址进行RDMA内存注册。其中,预设比例可以为2%、5%等。

[0080] 第二种情况、接收端可以每隔预设时间检查RDMA内存注册表,如果RDMA内存注册表中任一内存块在预设时长内未被访问,则将该内存块淘汰。其中,预设时间可以为5分钟、10分钟等。

[0081] 206、在第二接收内存申请成功后,接收端向发送端发送第二RDMA消息。

[0082] 其中,第二RDMA消息包括第二接收内存的目标内存信息等,目标内存信息包括第二接收内存的起始地址、第二接收内存的数据长度、目标键值等。

[0083] 207、在接收到第二RDMA消息后,发送端基于目标内存信息,通过RDMA的写操作向接收端发送目标数据。

[0084] 当接收到第二RDMA消息后,发送端将第二数据长度与第二接收内存所能存储数据的数据长度进行比较,由于第二接收内存所能存储数据的最小数据长度为第二数据长度,因而此时第二数据长度小于第二接收内存所能存储数据的数据长度,发送端基于目标内存信息,通过第二RDMA模块的写操作(RDMA write操作)向接收端发送目标数据。

[0085] 本申请的RDMA send/recv操作和RDMA write操作适用不同场景,对于小量数据,使用RDMA send/recv操作,虽然需要提前分配内存,内存利用率较低,但是能够降低传输时延;对于大量数据,使用RDMA write操作相比RDMA send/recv操作多了两次协商接收内存的通信过程,虽然传输时延较大,但是传输的数据量大,内存利用率高。

[0086] 208、当接收到目标数据,接收端将目标数据写入到第二接收内存中。

[0087] 在本申请实施例中,控制面事件循环线程和数据面事件循环线程都可以对第一RDMA模块接收发送端发送数据的事件进行监听,针对两种不同的监听方式的监听过程,下面将分别进行介绍。

[0088] 在一种可能的实现方式中,接收端通过第二方法对第一RDMA模块接收数据的事件

进行监听,当通过第二方法监听到第一RDMA模块已接收到目标数据,接收端确定获取到RDMA数据通道实例的数据面事件,进而基于控制面事件循环线程,将获取到的RDMA数据通道实例的数据面事件封装成任务,并将任务加入到任务队列中,然后唤醒数据面事件循环线程,执行数据面事件循环线程,以控制第一RDMA模块处理数据面事件,并在处理完数据面事件之后,处理任务队列中的任务,以将目标数据写入到第二接收内存。

[0089] 在另一种可能的实现方式中,接收端通过第一方法对第一RDMA模块接收数据的事件进行监听,当通过第一方法监听到第一RDMA模块已接收到目标数据,确定获取到RDMA数据通道实例的数据面事件,如果RDMA数据通道实例的数据面事件的数量不为0,且任务队列非空,执行数据面事件循环线程,以控制第一RDMA模块处理数据面事件,并在处理完数据面事件之后,处理任务队列中的任务,将目标数据写入到第二接收内存。

[0090] 图6示出了一种基于RDMA的数据传输方法,参见图6,接收端预先准备一块数据长度为N的接收内存recvBuf,发送端在发送数据长度为M的数据之前,判断数据长度M是否大于N,如果数据长度M小于N,则通过RDMA send操作向recvBuf写入数据;如果数据长度M大于N,则通知接收端准备长度为M的内存块,接收端接收到通知消息后,从内存池(用户态内存池)申请长度为M的内存块writeBuf,然后返回writeBuf的信息,接收端接收到该writeBuf的信息之后,通过RDMA write操作向writeBuf写入数据。

[0091] 上述所有可选技术方案,可以采用任意结合形成本申请的可选实施例,在此不再一一赘述。

[0092] 请参考图7,其示出了本申请实施例提供了一种基于RDMA的数据传输装置的结构示意图,该装置为接收端,该装置可以通过软件、硬件或者二者结合实现,成为电子设备的全部或一部分,该装置包括:

[0093] 接收模块701,用于接收模块,用于接收发送端发送的第一RDMA消息,所述第一RDMA消息为所述发送端在第一数据长度小于第二数据长度时发送,所述第一数据长度为所述接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度,所述第一RDMA消息包括所述发送端本次待发送的目标数据的第二数据长度;

[0094] 申请模块702,用于从用户态内存池申请第二接收内存,第二接收内存所能存储数据的最小数据长度为第二数据长度;

[0095] 发送模块703,用于在所述第二接收内存申请成功后,向所述发送端发送第二RDMA消息,所述第二RDMA消息包括所述第二接收内存的目标内存信息,所述第二RDMA消息用于通知所述发送端基于所述目标内存信息,通过RDMA的写操作向所述接收端发送所述目标数据;

[0096] 写入模块704,用于当接收到目标数据,将目标数据写入到第二接收内存中。

[0097] 在本申请的另一个实施例中,接收端配置有接收线程和选择线程,选择线程对应一个数据面事件通道;

[0098] 接收线程用于接收发送端发送的连接请求,对RDMA资源进行初始化操作,在完成初始化操作后,为发送端创建RDMA数据通道实例;

[0099] 接收线程还用于将RDMA数据通道实例注册到选择线程中,并在注册过程中将RDMA数据通道实例与选择线程对应的数据面事件通道绑定;

[0100] 选择线程用于对数据面事件通道进行监听,以获取RDMA数据通道实例的数据面事

件,数据面事件是指接收端接收到发送端发送的数据的事件。

[0101] 在本申请的另一个实施例中,选择线程关联数据面事件循环线程程,数据面事件循环线程用于执行以下操作:

[0102] 在开始一轮数据面事件循环后,如果接收端通过第一方法获取的数据面事件的数量为0,且任务队列为空,阻塞等待数据面事件通知;

[0103] 在接收到数据面事件通知后被唤醒,然后处理数据面事件;

[0104] 在处理完数据面事件之后,处理任务队列中的任务;

[0105] 在处理完任务队列中的任务之后,开始新一轮的数据面事件循环。

[0106] 在本申请的另一个实施例中,数据面事件循环线程还用于执行以下操作:

[0107] 如果接收端通过第一方法获取到数据面事件的数量不为0,且任务队列非空,处理数据面事件;

[0108] 在处理完数据面事件之后,处理任务队列中的任务;

[0109] 在处理完任务队列中的任务之后,开始新一轮的数据面事件循环。

[0110] 在本申请的另一个实施例中,选择线程还关联有控制面事件循环线程,控制面事件循环线程用于执行以下操作:

[0111] 在开始一轮控制面事件循环后,通过第二方法阻塞获取控制面事件;

[0112] 在获取到控制面事件之后,将获取到的控制面事件封装成任务,将封装的任务提交到任务队列;

[0113] 唤醒数据面事件循环线程,以使数据面事件循环线程处理数据面事件,并在处理完数据面事件之后,处理任务队列中的任务;

[0114] 在处理完任务队列中的任务之后,开始新一轮的控制面事件循环。

[0115] 在本申请的另一个实施例中,写入模块704,用于当通过第二方法监听到已接收到目标数据,确定获取到RDMA数据通道实例的数据面事件;基于控制面事件循环线程将获取到的RDMA数据通道实例的数据面事件封装成任务,将任务加入到任务队列中,然后唤醒数据面事件循环线程;通过执行数据面事件循环线程处理数据面事件;在处理完数据面事件之后,处理任务队列中的任务,以将目标数据写入到第二接收内存。

[0116] 在本申请的另一个实施例中,写入模块704,用于当通过第一方法监听到已接收到目标数据,确定获取到RDMA数据通道实例的数据面事件;如果RDMA数据通道实例的数据面事件的数量不为0,且任务队列非空,通过数据面事件循环线程处理数据面事件;在处理完数据面事件之后,处理任务队列中的任务,以将目标数据写入到第二接收内存。

[0117] 在本申请的另一个实施例中,申请模块702,用于从用户态内存池中申请最小数据长度为第二数据长度的目标用户态内存;获取目标用户态内存所属内存块的起始地址;检查起始地址是否在RDMA内存注册表中,RDMA内存注册表中存储有已进行RDMA内存注册的内存块的起始地址与键值之间的对应关系;当起始地址位于RDMA内存注册表中,从RDMA内存注册表中获取起始地址对应的目标键值;将目标用户态内存作为第二接收内存,并将起始地址和目标键值作为目标内存信息,以完成第二接收内存的申请。

[0118] 在本申请的另一个实施例中,该装置还包括:

[0119] 注册模块,用于当起始地址未位于RDMA内存注册表中,对起始地址进行RDMA内存注册,将注册得到的目标键值以及起始地址写入到RDMA内存注册表中;

[0120] 确定模块,用于将目标用户态内存作为第二接收内存;

[0121] 确定模块,还用于将起始地址和目标键值作为目标内存信息,以完成第二接收内存的申请。

[0122] 在本申请的另一个实施例中,该装置还包括:

[0123] 排序模块,用于如果对起始地址进行RDMA内存注册失败,按照访问时间由远及近的顺序,对RDMA内存注册表中已注册的各个内存块进行排序;

[0124] 注册模块,用于将RDMA内存注册表中位于排序结果前预设比例的内存块淘汰,然后再重新对起始地址进行RDMA内存注册。

[0125] 在本申请的另一个实施例中,该装置还包括:

[0126] 检查模块,用于每隔预设时间检查RDMA内存注册表;

[0127] 淘汰模块,用于如果RDMA内存注册表中任一内存块在预设时长内未被访问,将内存块淘汰。

[0128] 请参考图8,其示出了本申请实施例提供了一种基于RDMA的数据传输装置的结构示意图,该装置为发送端,该装置可以通过软件、硬件或者二者结合实现,成为电子设备的全部或一部分,该装置包括:

[0129] 判断模块801,用于判断目标数据的第二数据长度是否大于第一数据长度,第一数据长度为接收端从用户态内存池申请的第一接收内存所能存储数据的最大数据长度;

[0130] 发送模块802,用于当第二数据长度大于第一数据长度,向接收端发送第一RDMA消息,第一RDMA消息包括第二数据长度,第一RDMA消息用于通知接收端从用户态内存池申请第二接收内存,第二接收内存所能存储数据的最小数据长度为第二数据长度,并在申请成功后,向发送端发送第二RDMA消息,第二RDMA消息包括第二接收内存的目标内存信息;

[0131] 发送模块802,还用于在接收到第二RDMA消息后,基于目标内存信息,通过RDMA的写操作向接收端发送目标数据。

[0132] 在本申请的另一个实施例中,该装置还包括:

[0133] 发送模块,还用于当第二数据长度不大于第一数据长度,通过RDMA的发送操作向接收端发送目标数据。

[0134] 图9示出了本申请一个示例性实施例提供的一种电子设备900的结构框图。通常,电子设备900包括有:处理器901和存储器902。

[0135] 处理器901可以采用DSP(Digital Signal Processing,数字信号处理)、FPGA(Field—Programmable Gate Array,现场可编程门阵列)、PLA(Programmable Logic Array,可编程逻辑阵列)中的至少一种硬件形式来实现。处理器901也可以包括主处理器和协处理器,主处理器是用于对在唤醒状态下的数据进行处理的处理器;协处理器是用于对在待机状态下的数据进行处理的低功耗处理器。在一些实施例中,处理器901可以在集成有GPU(Graphics Processing Unit,图像处理器),GPU用于负责显示屏所需要显示的内容的渲染和绘制。一些实施例中,处理器901还可以包括人工智能处理器,该人工智能处理器用于处理有关机器学习的计算操作。

[0136] 存储器902可以包括一个或多个计算机可读存储介质,该计算机可读存储介质可以是非临时性计算机可读存储介质,例如,所述非临时性计算机可读存储介质可以是CD-ROM(Compact Disc Read-Only Memory,只读光盘)、ROM、RAM(Random Access Memory,随机

存取存储器)、磁带、软盘和光数据存储设备等。该计算机可读存储介质中存储有至少一条计算机程序,该至少一条计算机程序被执行时能够实现上述基于RDMA的数据传输方法。

[0137] 当然,上述电子设备必然还可以包括其他部件,例如输入/输出接口、通信组件等。输入/输出接口为处理器和外围接口模块之间提供接口,上述外围接口模块可以是输出设备、输入设备等。通信组件被配置为便于电子设备和其他设备之间有线或无线方式的通信等。

[0138] 本领域技术人员可以理解,图9中示出的结构并不构成对电子设备900的限定,可以包括比图示更多或更少的组件,或者组合某些组件,或者采用不同的组件布置。

[0139] 本申请实施例提供了一种计算机可读存储介质,所述计算机可读存储介质中存储有至少一条计算机程序,所述至少一条计算机程序被处理器执行时能够实现上述基于RDMA的数据传输方法。

[0140] 本申请实施例提供了一种计算机程序产品,所述计算机程序产品包括计算机程序,所述计算机程序被处理器执行时能够实现上述基于RDMA的数据传输方法。

[0141] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统,装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0142] 以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

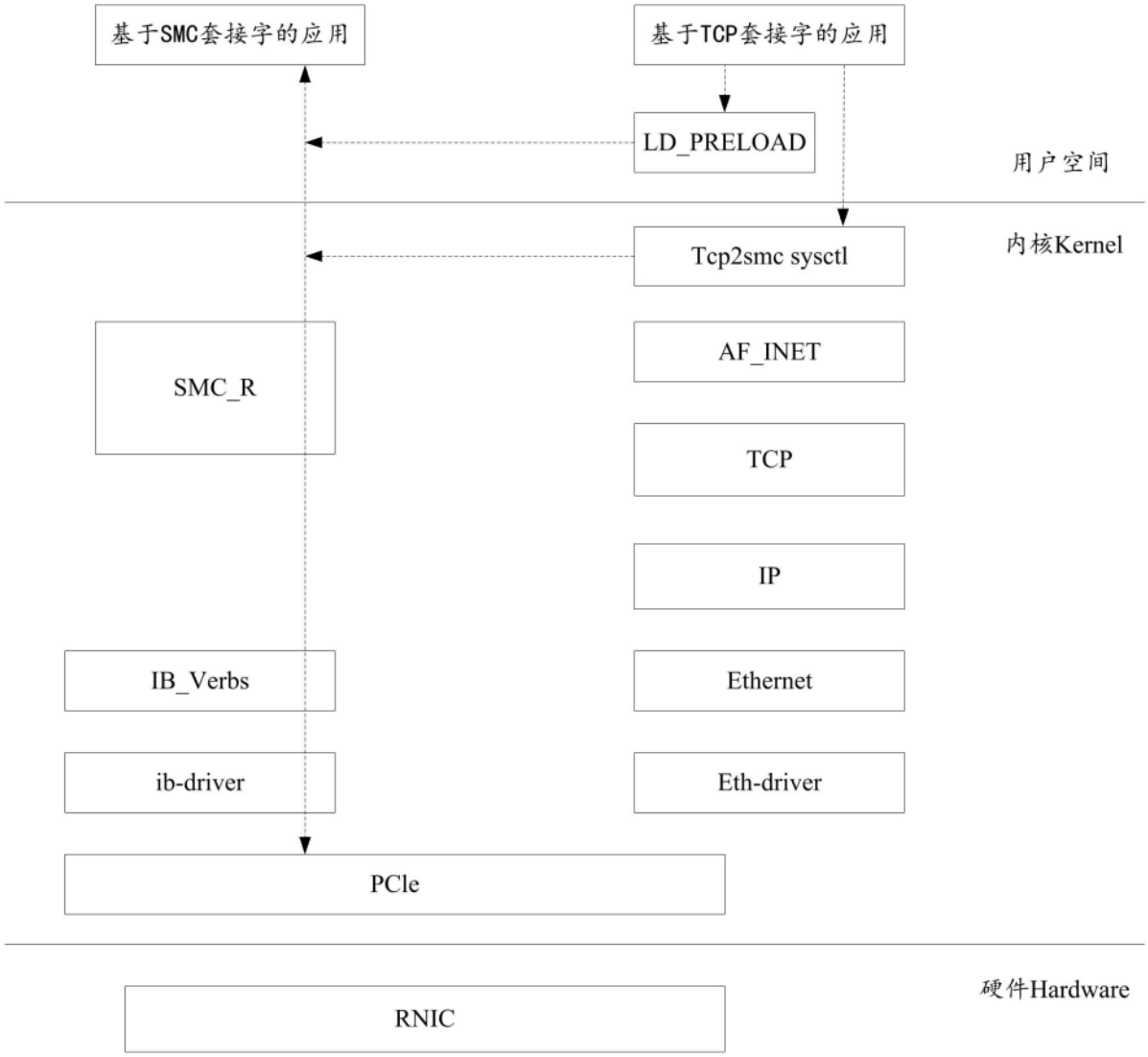

基于SMC套接字的应用
基于TCP套接字的应用
LD_PRELOAD
用户空间
Tcp2smc sysctl
内核Kernel
SMC_R
AF_INET
TCP
IP
IB_Verbs
Ethernet
ib-driver
Eth-driver
PCIe
硬件Hardware
RNIC

图1

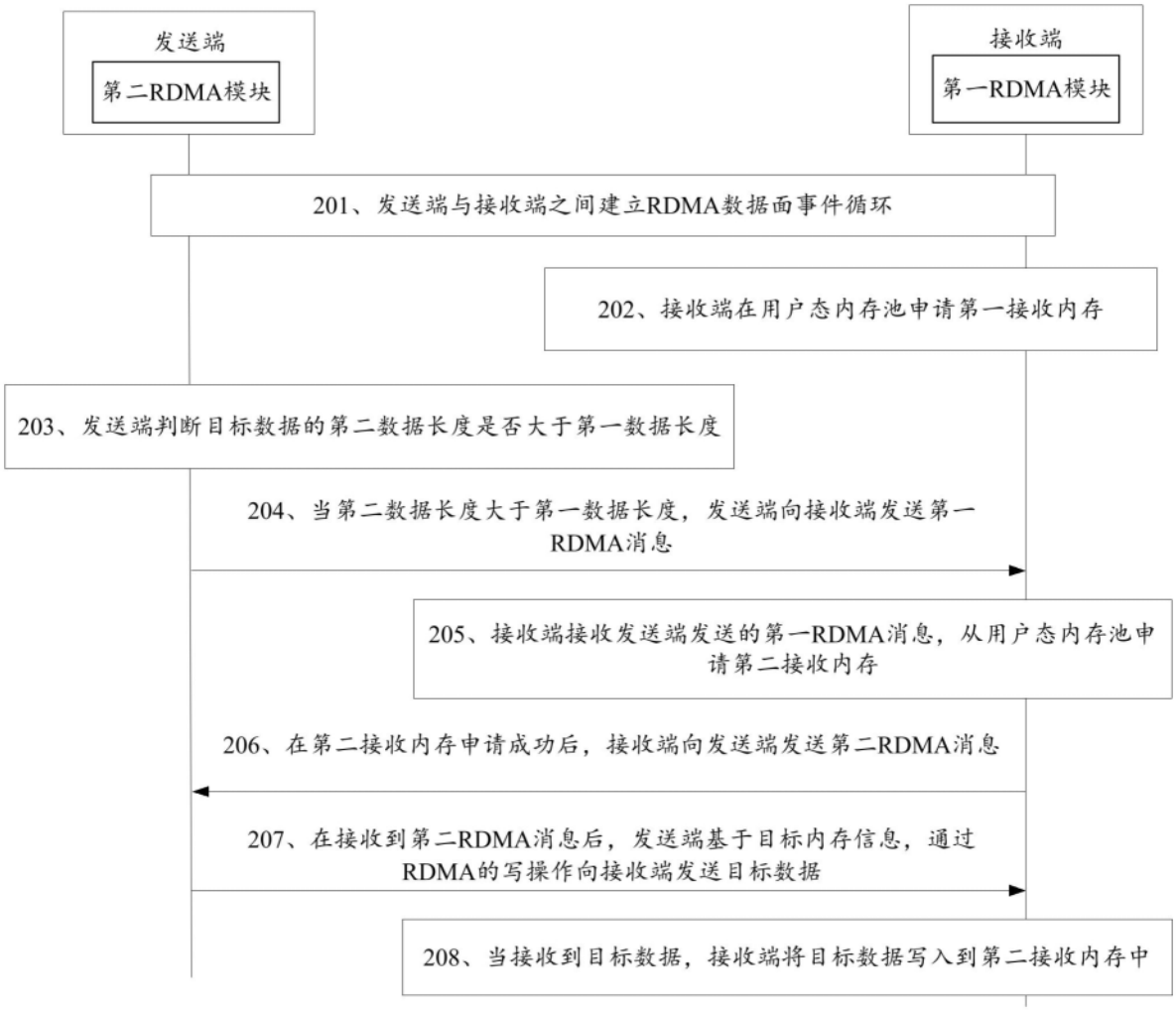

发送端
第二RDMA模块
接收端
第一RDMA模块
201、发送端与接收端之间建立RDMA数据面事件循环
202、接收端在用户态内存池申请第一接收内存
203、发送端判断目标数据的第二数据长度是否大于第一数据长度
204、当第二数据长度大于第一数据长度，发送端向接收端发送第一RDMA消息
205、接收端接收发送端发送的第一RDMA消息，从用户态内存池申请第二接收内存
206、在第二接收内存申请成功后，接收端向发送端发送第二RDMA消息
207、在接收到第二RDMA消息后，发送端基于目标内存信息，通过RDMA的写操作向接收端发送目标数据
208、当接收到目标数据，接收端将目标数据写入到第二接收内存中

图2

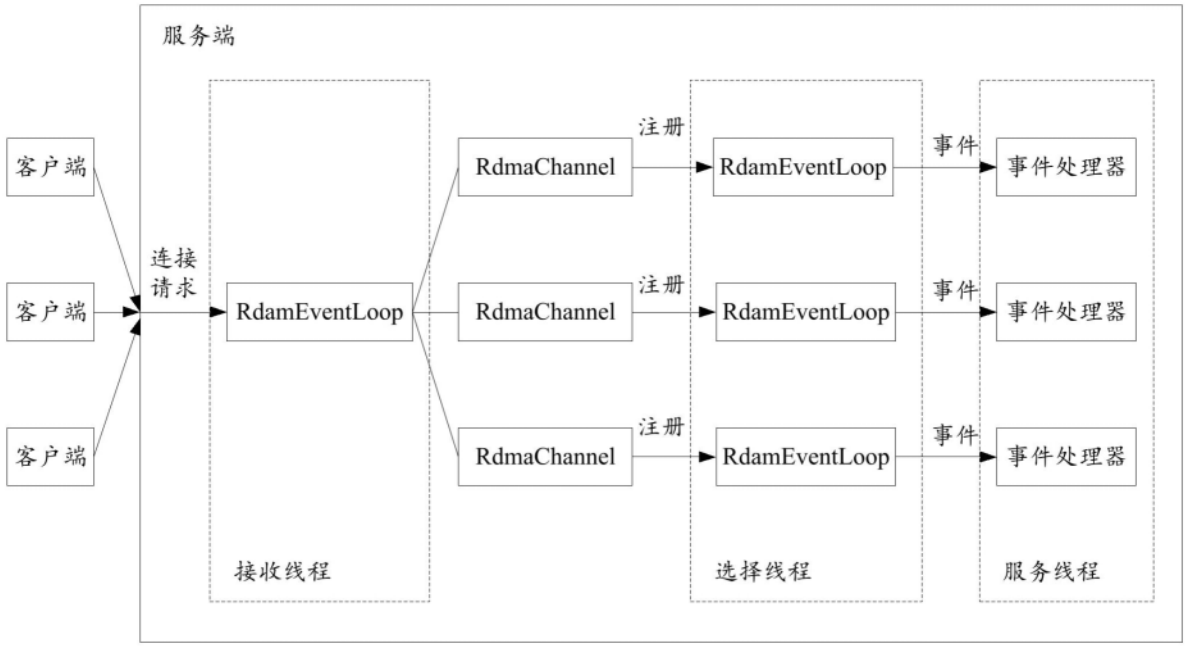

服务端
客户端
客户端
客户端
连接请求
RdamEventLoop
RdmaChannel
RdmaChannel
RdmaChannel
注册
注册
注册
RdamEventLoop
RdamEventLoop
RdamEventLoop
事件
事件
事件
事件处理器
事件处理器
事件处理器
接收线程
选择线程
服务线程

图3

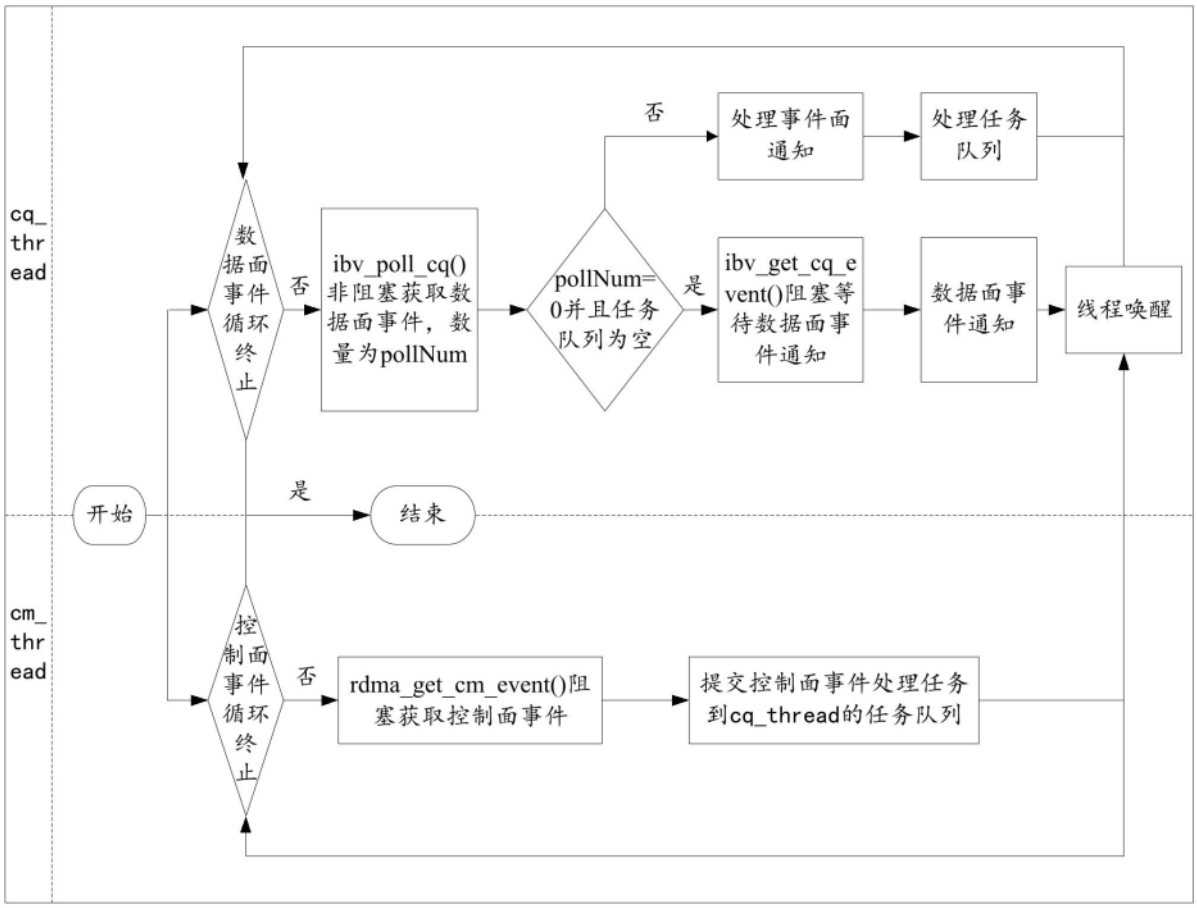

cq_thread
cm_thread
开始
数据面事件循环终止
否
ibv_poll_cq()非阻塞获取数据面事件，数量为pollNum
pollNum=0并且任务队列为空
否
是
处理事件面通知
处理任务队列
ibv_get_cq_event()阻塞等待数据面事件通知
数据面事件通知
线程唤醒
是
结束
控制面事件循环终止
否
rdma_get_cm_event()阻塞获取控制面事件
提交控制面事件处理任务到cq_thread的任务队列

图4

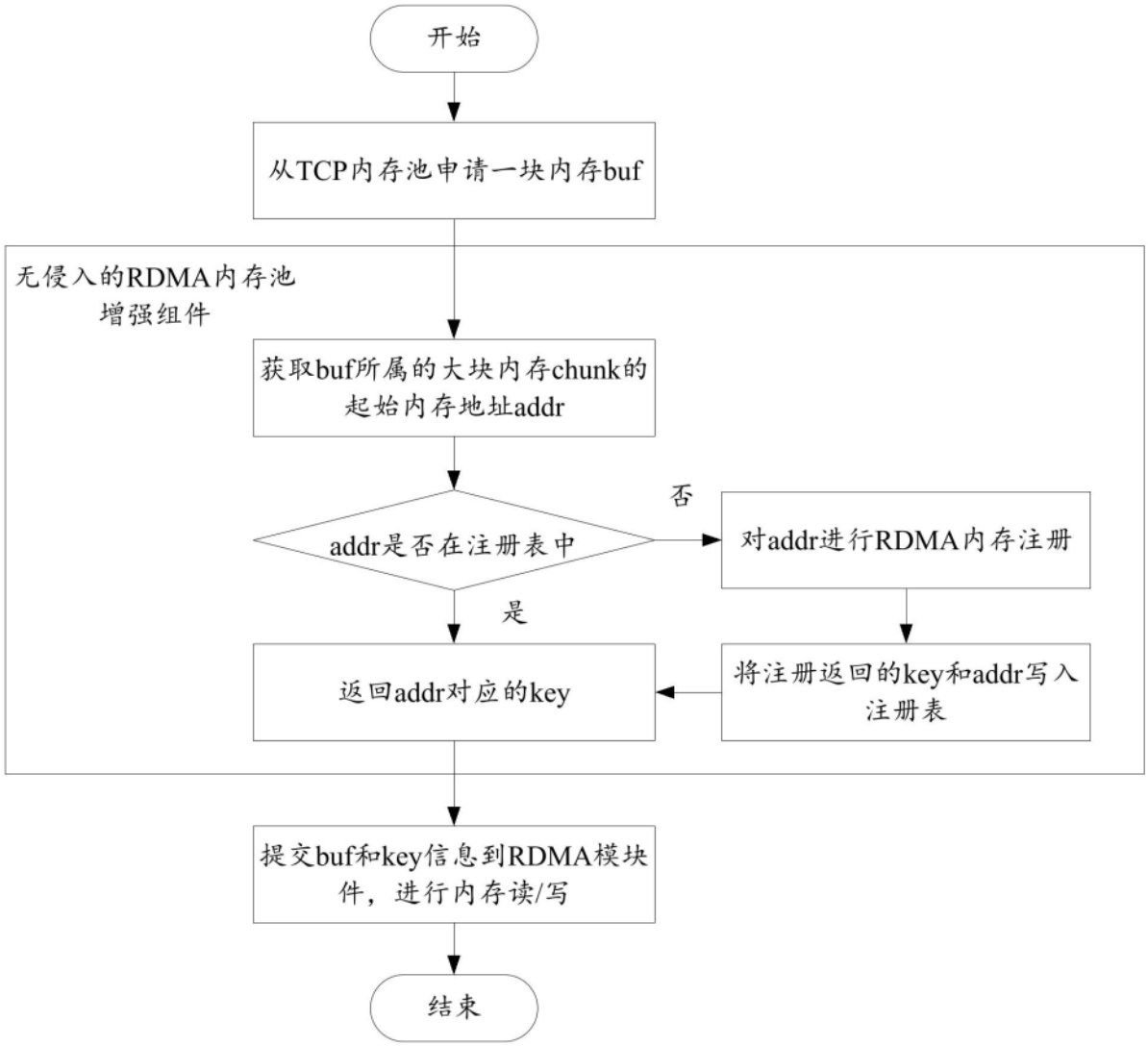

开始
从TCP内存池申请一块内存buf
无侵入的RDMA内存池
增强组件
获取buf所属的大块内存chunk的
起始内存地址addr
addr是否在注册表中
否
对addr进行RDMA内存注册
是
返回addr对应的key
将注册返回的key和addr写入
注册表
提交buf和key信息到RDMA模块
件，进行内存读/写
结束

图5

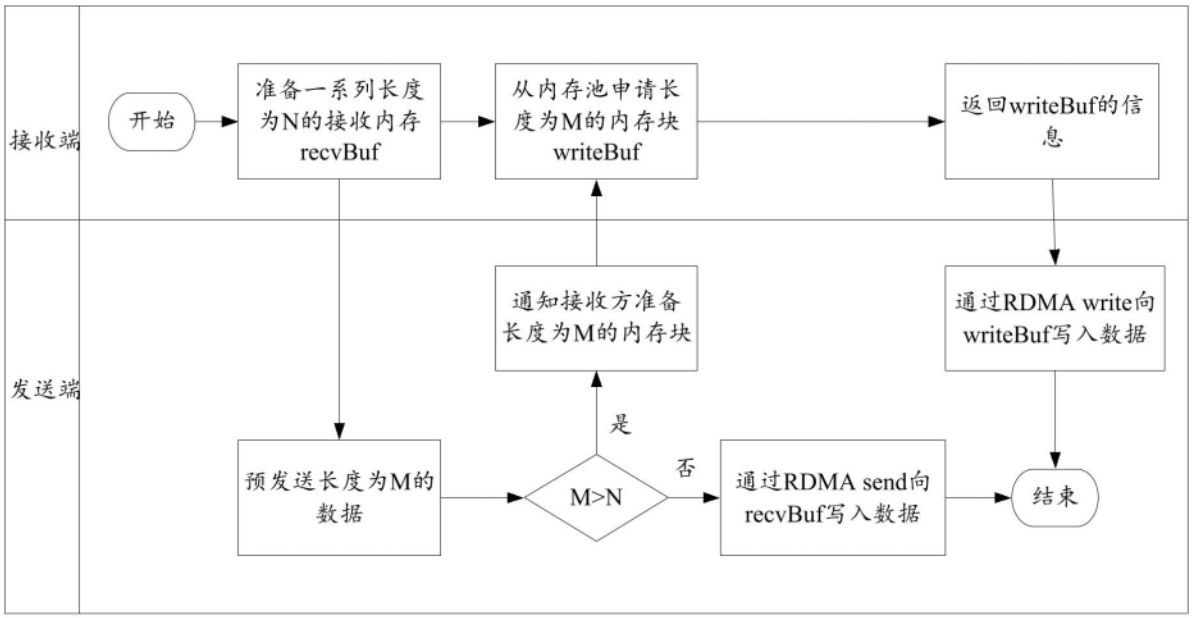

接收端
开始
准备一系列长度为N的接收内存recvBuf
从内存池申请长度为M的内存块writeBuf
返回writeBuf的信息
发送端
通知接收方准备长度为M的内存块
通过RDMA write向writeBuf写入数据
是
否
预发送长度为M的数据
M>N
通过RDMA send向recvBuf写入数据
结束

图6

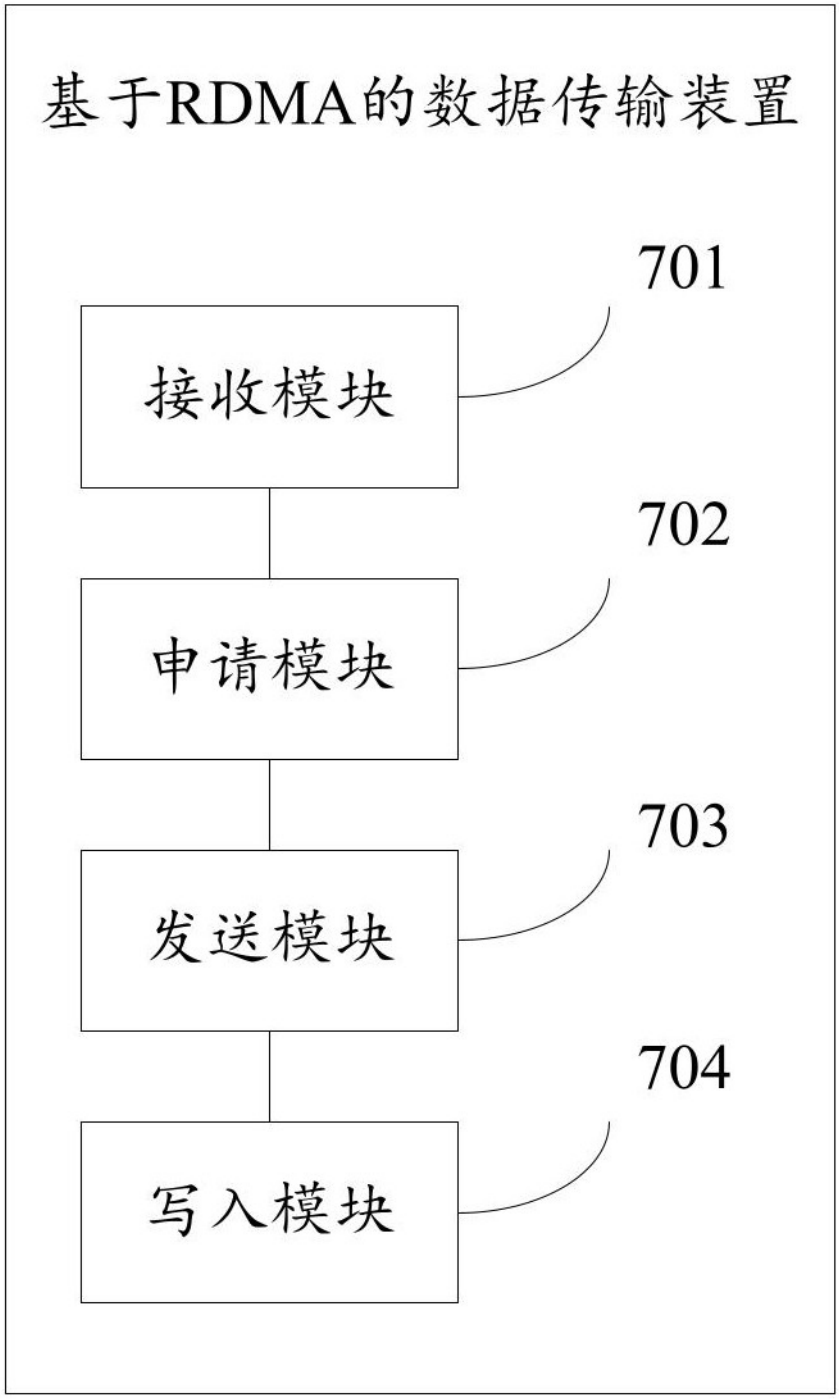


图7

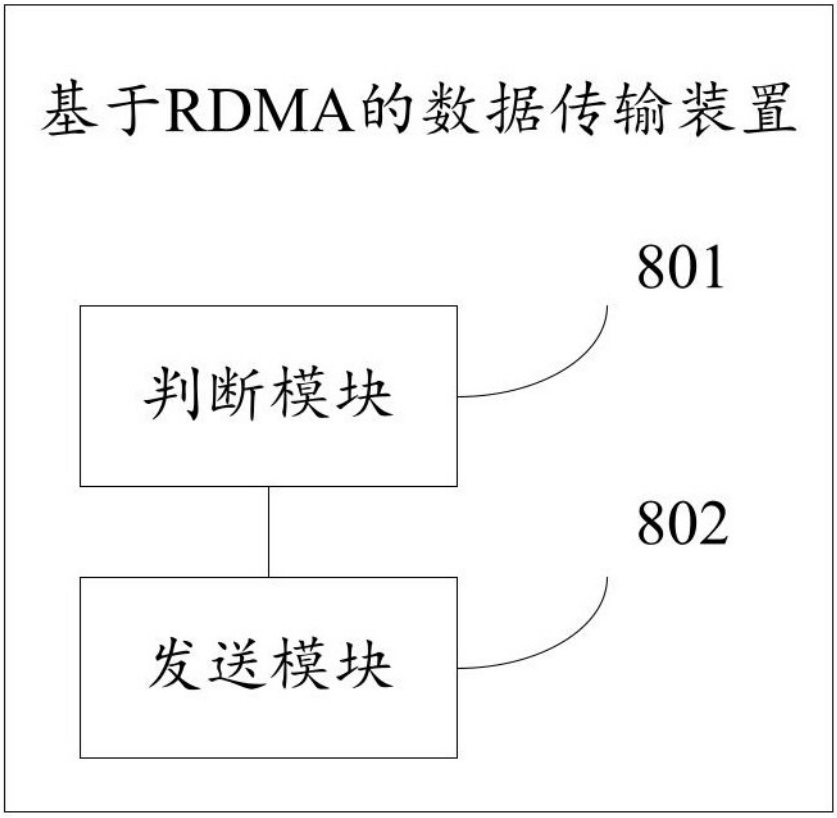


图8

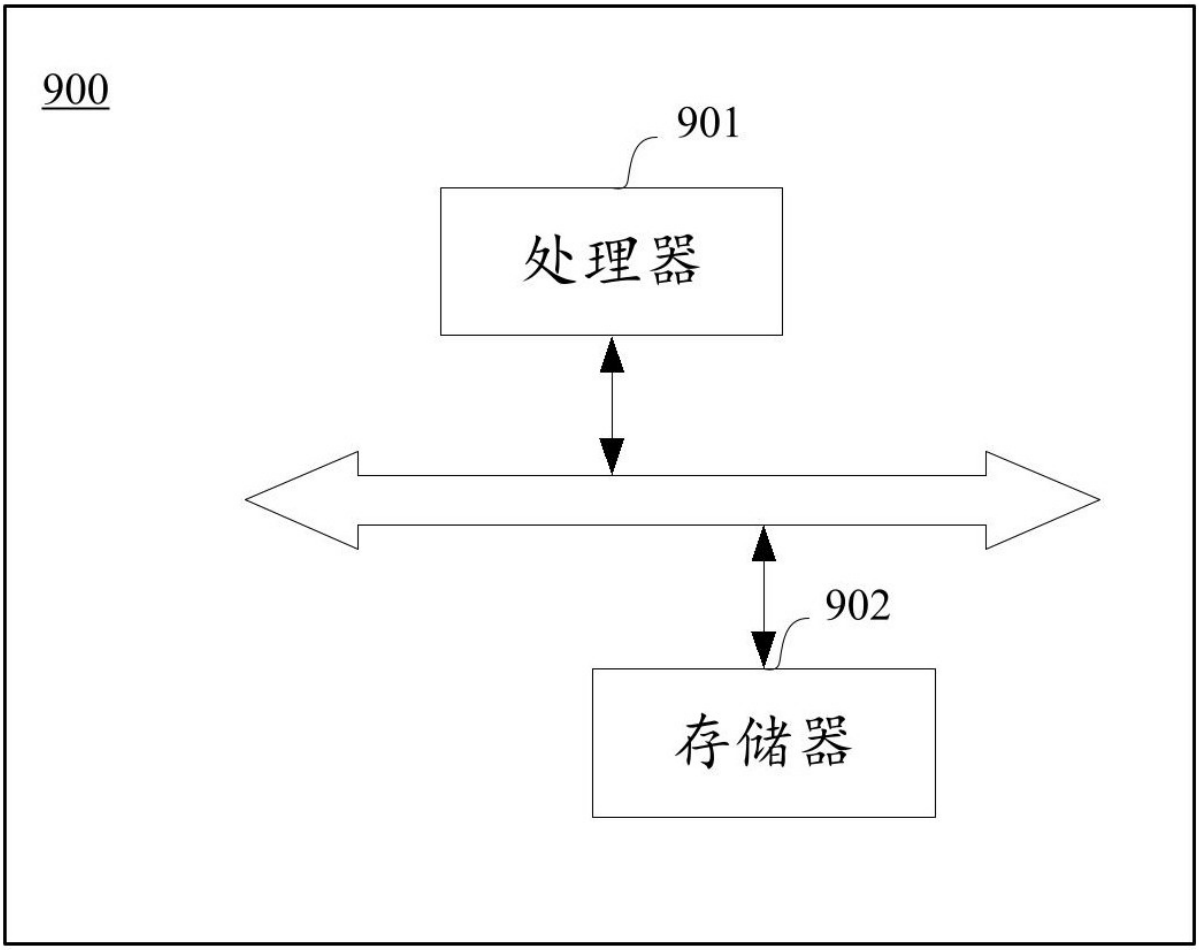


图9