(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108694092 B
(45)授权公告日 2021.01.15

(21)申请号 201810453062.6
(22)申请日 2018.05.11
(65)同一申请的已公布的文献号
申请公布号 CN 108694092 A
(43)申请公布日 2018.10.23
(73)专利权人 华中科技大学
地址 430074 湖北省武汉市洪山区珞喻路1037号
(72)发明人 王多强 金海 张弛
(74)专利代理机构 北京海虹嘉诚知识产权代理有限公司 11129
代理人 何志欣 侯越玲
(51)Int.Cl.
G06F 9/54(2006.01)

审查员 梁滔

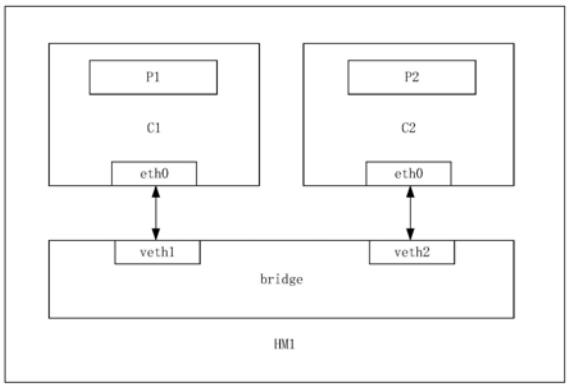
权利要求书2页 说明书12页 附图3页

(54)发明名称

一种面向并行应用的容器通信方法和系统

(57)摘要

本发明涉及一种面向并行应用的容器通信方法和系统，该方法包括：第一容器中的第一进程需要和第二容器中的第二进程进行通信且第一和第二容器处于同一个宿主机上的情况下，建立一个不同于基于TCP协议的第一信道的第二信道，第一容器将通信数据发送到第一容器和/或第二容器在宿主机上申请的共享内存区域并将通信数据的元数据通过第一信道发送给第二容器，在所述第二进程根据接收到的所述元数据确认接收所述通信数据的情况下，通过第二信道将所述通信数据传递给第二容器且通过第一信道将接收数据的确认消息反馈给第一进程。



CN 108694092 B

1.一种面向并行应用的容器通信方法,其特征在于,其包括:

在第一容器(C1)中的第一进程(P1)需要和第二容器(C2)中的第二进程(P2)进行通信且第一和第二容器(C2)处于同一个宿主机(HM1)上的情况下,由所述宿主机(HM1)在所述第一容器(C1)和第二容器(C2)之间建立一个不同于基于TCP协议的第一信道的第二信道,

所述第一容器(C1)将所述第一进程(P1)与所述第二进程(P2)进行通信的通信数据发送到第一容器(C1)和/或第二容器(C2)在宿主机(HM1)上申请的共享内存区域并将通信数据的元数据通过第一信道发送给第二容器(C2),

在所述第二进程(P2)根据接收到的所述元数据确认接收所述通信数据的情况下,所述第一容器(C1)通过所述第二信道将所述通信数据传递给第二容器(C2)且所述第二进程(P2)通过第一信道将接收数据的确认消息反馈给第一进程(P1),

由第一容器C1根据通信数据的重要程度划分安全等级且在通信数据的安全等级超过预设安全阈值之时,第二信道采用不同于第二加密机制的第三加密机制,其中,第二加密机制采用三重对称加密,第三加密机制采用非对称加密,

所述元数据为描述数据的数据,是描述数据属性的信息,用来支持的功能包括:指示存储位置、历史数据、资源查找、文件记录。

2.如权利要求1所述的方法,其特征在于,所述方法还包括:

对所述第一容器(C1)和所述第二容器(C2)所处的位置进行判断,其包括:

在所述第一进程(P1)和所述第二进程(P2)需要进行通信的情况下,所述第一进程(P1)尝试通过调用操作系统的Socket编程接口与所述第二进程(P2)建立通信,

当所述第一进程(P1)和所述第二进程(P2)通过Socket编程接口实现消息传递时,获取第一容器(C1)的第一IP地址和第二容器(C2)的第二IP地址并结合对应的子网掩码计算第一容器(C1)的第一网络号和第二容器(C2)的第二网络号,

在第一网络号和第二网络号相同的情况下确认第一容器(C1)和第二容器(C2)处于同一个宿主机(HM1)上。

3.如权利要求2所述的方法,其特征在于,所述建立一个不同于基于TCP协议的第一信道的第二信道的处理包括:

在确认第一容器(C1)和第二容器(C2)处于同一个宿主机(HM1)上之后,先等待第一进程(P1)与第二进程(P2)之间完成建立基于TCP协议的所述第一信道,

然后根据第一信道的TCP连接的第一语义信息建立一个能用于对所述通信数据进行传输的第二信道。

4.如权利要求1所述的方法,其特征在于,所述方法还包括:

根据所述接收数据的确认消息确定第一进程(P1)与第二进程(P2)之间是否存在进一步的数据交换,

若是,则在确认了第一容器(C1)的第一IP地址和第二容器(C2)的第二IP地址均未发生变化的情况下,继续使用所述共享内存区域、已建立的所述第一信道和所述第二信道按照后续通信数据通过第二信道传输且后续通信数据的元数据通过第一信道传输的方式完成第一进程(P1)和第二进程(P2)的数据通信;

若否,则先协商释放第二信道,并在第二信道释放后才释放第一信道。

5.如权利要求1所述的方法,其特征在于,所述第一容器(C1)将通信数据发送到第一容

器(C1)和/或第二容器(C2)在宿主机(HM1)上申请的共享内存区域的处理包括：

识别第一进程(P1)通过Socket编程接口将所述通信数据发送到内核的状态，然后由驱动接口先将所述通信数据从内核拷贝到共享内存区域。

6.如权利要求5所述的方法，其特征在于，通过所述第二信道将所述通信数据传递给第二容器(C2)是通过调用驱动接口将所述通信数据拷贝到第二进程(P2)的进程空间来实现的。

7.如权利要求4所述的方法，其特征在于，在通过第二信道传递通信数据之前，第二进程(P2)通过加密的第一信道向第一进程(P1)传输对称密钥，第一进程(P1)先使用所述对称密钥将所述通信数据加密后再通过第二信道将所述通信数据传递给第二容器(C2)。

8.如权利要求7所述的方法，其特征在于，所述对称密钥是由随机算法产生的。

9.一种面向并行应用的容器通信系统，其特征在于，该系统包括：至少一个处理器以及至少一个计算机可读存储介质，被配置为执行以下操作：

在第一容器(C1)中的第一进程(P1)需要和第二容器(C2)中的第二进程(P2)进行通信且第一和第二容器(C2)处于同一个宿主机(HM1)上的情况下，由所述宿主机(HM1)在所述第一容器(C1)和第二容器(C2)之间建立一个不同于基于TCP协议的第一信道的第二信道，

所述第一容器(C1)将所述第一进程(P1)与所述第二进程(P2)进行通信的通信数据发送到第一容器(C1)和/或第二容器(C2)在宿主机(HM1)上申请的共享内存区域并将通信数据的元数据通过第一信道发送给第二容器(C2)，

在所述第二进程(P2)根据接收到的所述元数据确认接收所述通信数据的情况下，所述第一容器(C1)通过所述第二信道将所述通信数据传递给第二容器(C2)且所述第二进程(P2)通过第一信道将接收数据的确认消息反馈给第一进程(P1)，

由第一容器C1根据通信数据的重要程度划分安全等级且在通信数据的安全等级超过预设安全阈值之时，第二信道采用不同于第二加密机制的第三加密机制，其中，第二加密机制采用三重对称加密，第三加密机制采用非对称加密，

所述元数据为描述数据的数据，是描述数据属性的信息，用来支持的功能包括：指示存储位置、历史数据、资源查找、文件记录。

10.如权利要求9所述的系统，其特征在于，所述至少一个处理器以及至少一个计算机可读存储介质还被配置为执行以下操作：

对所述第一容器(C1)和所述第二容器(C2)所处的位置进行判断，其包括：

在所述第一进程(P1)和所述第二进程(P2)需要进行通信的情况下，所述第一进程(P1)尝试通过调用操作系统的Socket编程接口与所述第二进程(P2)建立通信，

当所述第一进程(P1)和所述第二进程(P2)通过Socket编程接口实现消息传递时，获取第一容器(C1)的第一IP地址和第二容器(C2)的第二IP地址并结合对应的子网掩码计算第一容器(C1)的第一网络号和第二容器(C2)的第二网络号，

在第一网络号和第二网络号相同的情况下确认第一容器(C1)和第二容器(C2)处于同一个宿主机(HM1)上。

一种面向并行应用的容器通信方法和系统

技术领域

[0001] 本发明涉及通信领域，尤其涉及一种面向并行应用的容器通信方法和系统。

背景技术

[0002] 随着微型计算机处理器技术的飞速发展，过去十年间，处理器的计算能力按照“摩尔定律”所描述的速度在进行提高。目前，单台服务器能够提供的处理能够远超过去的服务器处理能力。而应用程序对计算的需求却没有增长得如此迅速，因此导致了现有的计算机处理能力普遍存在剩余的情况。为了进一步提高计算机处理资源的利用率，虚拟化技术被人们所提出。通过在同一个物理机上运行多个虚拟机，能够明显提高服务器的处理能力和资源利用率，同时也带来了在线迁移、动态部署、负载均衡等优势。

[0003] 随着技术的不断发展，科研单位和企业对于高性能计算的需求量不断增长，而高性能计算所需计算资源却因为不同单位的需求差异巨大，很多用户都只有在进行实际高性能计算时才能确定具体的计算资源需求，且不同情况对于计算需求量也会有比较大的差异，而科研单位的自建高性能集群有时无法满足用户的资源需求。还有一些单位仅仅暂时需要高性能计算计算资源，自己构建高性能计算集群的成本高、周期长。针对以上两种典型情况，使用云计算服务商提供的高性能计算虚拟服务器或虚拟实例可以很好地解决资源临时短缺和集群构建成本高、周期长的问题。

[0004] 然而，在使用虚拟化的物理机系统中，往往为了提高资源的利用率，在物理机上运行多台虚拟机，导致虚拟设备的数量远远超过物理机上资源的数量和能够承受的范围，导致了严重的资源过载现象。资源过载引起资源请求得不到及时响应的情况，直接导致高性能并行应用的计算速度受到影响。

[0005] 众所周知，典型的高性能并行应用都采用了基于消息传递的编程模型MPI(Message Passing Interface)。传统物理机情况下，为了优化同一主机上的多个MPI进程效率，MPI库提供了Shared Memory和Cross Memory Attach两种进程消息传递通道来优化相同主机上不同进程间消息的传递效率。而在云计算环境下，运行在同一宿主机上面的虚拟机或容器上面的不同MPI进程之间只能采用Socket即网络通信的方式进行消息传递，进程之间无法使用MPI库中提供的Shared Memory和Cross Memory Attach通道进行快速的进程通信，使得消息传递的速度和效率明显降低，严重影响了高性能并行应用的计算速度。

[0006] 针对高性能并行应用在容器虚拟化环境中存在的进程通信效率低的问题，有研究者提出了修改MPI库，并让MPI库能够检测到处于同一主机上相邻容器的方法，该方法能够将默认情况下不同容器MPI进程间的通信方式从网络通道修改为采用Shared Memory通信的模式，在一定程度上提高了高性能并行应用的计算效率。但是该方法需要修改现有的MPI部署环境，且修改MPI环境之后，还需要修改MPI应用的源代码才能支持这种优化。与此同时，若其他未采用MPI编程模型的高性能并行应用的通信效率将得不到任何优化，即该方法仅限于MPI编程模型，不能提高在容器中运行的其他应用例如事务型应用的通信效率。

[0007] 有研究者提出了一种基于共享内存的容器通信方法，该方法提供了一种基于客户

服务器模式的通信框架，使用该框架，应用必须修改应用的源代码，并且基于该通信框架进行编译，才能使用共享内存的方式进行容器通信，尽管该方式能够在一定程度上提高了同主机的通信效率，但是该方法的兼容性和可操作性不强，且现有的代码很多，改动成本大，因此实用性不强。

[0008] 还有研究者提供了另外一种基于共享内存的容器通信方式，该方法通过使用网络共享文件系统中共享文件的方式来进行容器之间的消息传递和协商，尽管该方法最终通过使用共享内存的形式进行容器间的数据传输，但是由于建立共享内存区域、数据拷贝、共享内存区域释放等这些控制信息的传递时通过文件来进行传递的，进行通信的容器双方均需要打开文件，读取或写入文件，导致整个通信过程建立的效率非常低，不仅需要进行至少两次的用户态与内核态之间的切换，而且还可能需要进行内存空间的申请，导致整个通信系统的效率并没有得到很大提高。

[0009] 公开号为CN105847108A的中国专利文献公开了一种容器间的通信方法及装置。该方法包括：第一虚拟网桥向子网内除第一虚拟网桥外的第二虚拟网桥发送第一消息，第一消息包括第一容器的地址信息和第一虚拟网桥的标识信息；第一虚拟网桥接收第二虚拟网桥发送的第二消息，第二消息包括第二容器的地址信息和第二虚拟网桥的标识信息；若第一虚拟网桥的标识信息和第二虚拟网桥的标识信息相同，则第一虚拟网桥将第二容器的地址信息发送给第一容器，以使第一容器依据第二容器的地址信息与第二容器通信。该专利申请实现了分散在不同Docker服务器上且属于同一用户的容器之间的正常通信，容器间通信不需要通过广播方式广播待发送的报文，提高了容器之间信息交互的保密性。但是，其通信的效率不高。

发明内容

[0010] 针对现有技术之不足，本发明提供了一种面向并行应用的容器通信方法和系统，该方法和系统能够优化运行在同一宿主机中不同容器里面所有类型的高性能并行应用的通信效率，使得同一主机中所有容器间通信都采用内存模式而不是Socket通信模式，这样可以极大提高同主机上不同容器间进程的通信效率，减少宿主机处理器进行网络I/O操作的等待时间，提高了高性能并行应用计算效率，同时进一步提高了系统资源的利用率，从而更好地满足并行应用的计算和通信需求，解决现有技术情况下，并行应用通信效率低和性能瓶颈的问题。

[0011] 根据一个优选实施方式，本发明公开了一种面向并行应用的容器通信方法，该方法包括：在第一容器中的第一进程需要和第二容器中的第二进程进行通信且第一和第二容器处于同一个宿主机上的情况下，由所述宿主机在所述第一容器和第二容器之间建立一个不同于基于TCP协议的第一信道的第二信道，所述第一容器将所述第一进程与所述第二进程进行通信的通信数据发送到第一容器和/或第二容器在宿主机上申请的共享内存区域并将通信数据的元数据通过第一信道发送给第二容器，在所述第二进程根据接收到的所述元数据确认接收所述通信数据的情况下，所述第一容器通过所述第二信道将所述通信数据传递给第二容器且所述第二进程通过第一信道将接收数据的确认消息反馈给第一进程。本发明的第一信道作为面向连接的确认通道，第二信道作为通信数据传输的专用信道，其能够避免篡改，提高安全性，且将内部通信开销降至最低，还提高了通信数据的传输速度，也能

避免通信过程中的冲突问题。首先，通过共享内存通信的稳定性不如通过TCP连接通信。其次，如果将数据和数据的元数据通过相同的信道发送也存在安全性问题。因此，通过该方式，将通信数据及其元数据通过不同的信道发送，提高了本发明的安全性。而且，本发明是将数据量小的元数据通过稳定性更好的基于TCP协议的第一信道发送，保证了数据传递的准确性和可靠性。而通过将数据量更大的通信数据通过传输效率更高的共享内存的方式传输，提高了并行应用通信的效率。

[0012] 根据一个优选实施方式，所述方法还包括：对所述第一容器和所述第二容器所处的位置进行判断，其包括：在所述第一进程和所述第二进程需要进行通信的情况下，所述第一进程尝试通过调用操作系统的Socket编程接口与所述第二进程建立通信，当所述第一进程和所述第二进程通过Socket编程接口实现消息传递时，获取第一容器的第一IP地址和第二容器的第二IP地址并结合对应的子网掩码计算第一容器的第一网络号和第二容器的第二网络号，在第一网络号和第二网络号相同的情况下确认第一容器和第二容器处于同一个宿主机上。通过该方式，本发明能够迅速、准确地判断第一容器和第二容器所处的位置并在两者处于同一宿主机的情况下采用本发明的共享内存的方式进行通信，以提高通信效率。

[0013] 根据一个优选实施方式，所述建立一个不同于基于TCP协议的第一信道的第二信道的处理包括：在确认第一容器和第二容器处于同一个宿主机上之后，先等待第一进程与第二进程之间完成建立基于TCP协议的所述第一信道，然后根据第一信道的TCP连接的第一语义信息建立一个能用于对所述通信数据进行传输的第二信道。通过该方式，本发明可以基于Socket原有的语义来建立第二信道，不需要修改第一进程和第二进程的源代码，具有良好地兼容性和实用性。

[0014] 根据一个优选实施方式，所述方法还包括：根据所述接收数据的确认消息确定第一进程与第二进程之间是否存在进一步的数据交换，若是，则在确认了第一容器的第一IP地址和第二容器的第二IP地址均未发生变化的情况下，继续使用所述共享内存区域、已建立的所述第一信道和所述第二信道按照后续通信数据通过第二信道传输且后续通信数据的元数据通过第一信道传输的方式完成第一进程和第二进程的数据通信；若否，则先协商释放第二信道，并在第二信道释放后才释放第一信道。通过该方式，可以提高后续数据通信的效率，而且，先释放第二信道可以使得在第二信道释放期间或者释放完成而第一信道还未释放，但第一进程和第二进程又需要进行通信连接的情况下，快速地基于第一信道建立第二信道，提高通信的效率。

[0015] 根据一个优选实施方式，所述第一容器将通信数据发送到第一容器和/或第二容器在宿主机上申请的共享内存区域的处理包括：识别第一进程通过Socket编程接口将所述通信数据发送到内核的状态，然后由驱动接口先将所述通信数据从内核拷贝到共享内存区域的。通过该方式，可以避免改写第一进程的源代码，提高本发明的兼容性和实用性。

[0016] 根据一个优选实施方式，所述通过第二信道将所述通信数据传递给第二容器是通过调用驱动接口将所述通信数据拷贝到第二进程的进程空间来实现的。通过该方式，可以避免改写第二进程的源代码，提高本发明的兼容性和实用性。

[0017] 根据一个优选实施方式，在通过第二信道传递通信数据之前，第二进程通过加密的第一信道向第一进程传输对称密钥，第一进程先使用所述对称密钥将所述通信数据加密后再通过第二信道将所述通信数据传递给第二容器。通过该方式，可以在少量增加计算开

销的情况下提高本发明的安全性。

[0018] 根据一个优选实施方式，所述对称密钥是由随机算法产生的。通过该方式，可以进一步提高本发明的安全性。

[0019] 根据一个优选实施方式，本发明还公开了一种面向并行应用的容器通信系统，该系统包括：至少一个处理器以及至少一个计算机可读存储介质，被配置为执行以下操作：在第一容器中的第一进程需要和第二容器中的第二进程进行通信且第一和第二容器处于同一个宿主机上的情况下，由所述宿主机在所述第一容器和第二容器之间建立一个不同于基于TCP协议的第一信道的第二信道，所述第一容器将所述第一进程与所述第二进程进行通信的通信数据发送到第一容器和/或第二容器在宿主机上申请的共享内存区域并将通信数据的元数据通过第一信道发送给第二容器，在所述第二进程根据接收到的所述元数据确认接收所述通信数据的情况下，所述第一容器通过所述第二信道将所述通信数据传递给第二容器且所述第二进程通过第一信道将接收数据的确认消息反馈给第一进程。

[0020] 根据一个优选实施方式，所述至少一个处理器以及至少一个计算机可读存储介质还被配置为执行以下操作：对所述第一容器和所述第二容器所处的位置进行判断，其包括：在所述第一进程和所述第二进程需要进行通信的情况下，所述第一进程尝试通过调用操作系统的Socket编程接口与所述第二进程建立通信，当所述第一进程和所述第二进程通过Socket编程接口实现消息传递时，获取第一容器的第一IP地址和第二容器的第二IP地址并结合对应的子网掩码计算第一容器的第一网络号和第二容器的第二网络号，在第一网络号和第二网络号相同的情况下确认第一容器和第二容器处于同一个宿主机上。

附图说明

[0021] 图1是一种常见的一个宿主机下同时运行并行应用的运行环境示意图；

[0022] 图2是本发明的系统的一个可选的实施方式的架构示意图；

[0023] 图3是本发明的系统的另一个可选的实施方式的架构示意图；和

[0024] 图4是本发明的方法的一个优选实施方式的流程示意图。

[0025] 附图标记列表

[0026] C1：第一容器 C2：第二容器 P1：第一进程

[0027] P2：第二进程 HM1：宿主机 10A：内核扩展模块

[0028] 10B：PCI扩展卡 11：路由模块 12：协议模块

[0029] 13：驱动模块 14：网络接口

具体实施方式

[0030] 下面结合附图进行详细说明。

[0031] 为了便于理解，在可能的情况下，使用相同附图标记来表示各附图中共同的相似元件。

[0032] 如在整篇本申请中所使用的那样，词语“可以”系容许含义(即，意味着有可能的)而不是强制性含义(即，意味着必须的)。类似地，词语“包括”意味着包括但不限于。

[0033] 短语“至少一个”、“一个或多个”以及“和/或”系开放式表达，它们涵盖操作中的关联与分离两者。例如，表述“A、B和C中的至少一个”、“A、B或C中的至少一个”、“A、B和C中的一

个或更多个”、“A、B或C”和“A、B和/或C”中的每个分别指单独A、单独B、单独C、A和B一起、A和C一起、B和C一起或A、B和C一起。

[0034] 术语“一种”或“一个”实体指的是该实体中的一个或多个。这样，术语“一”(或“一”)、“一个或多个”以及“至少一个”在本文中可以交换地使用。还应该注意，术语“包括”、“包含”和“具有”可以交换地使用。

[0035] 如本文中所使用的那样，术语“自动的”及其变型是指当执行过程或操作时在没有实质性人工输入的情况下完成的任何过程或操作。然而，如果在执行该过程或操作之前接收到该输入，则该过程或操作可以是自动的，即使该过程或操作的执行使用了实质性或非实质性的人工输入。如果这样的输入影响该过程或操作的执行方式，则该人工输入被认为是实质性的。准予执行该过程或操作的人工输入不被视为“实质性的”。

[0036] 并行应用程序是采用并行框架编写且以单机多线程和/或多机多进程形式运行的应用程序。优选地，并行框架例如是多线程、共享内存或者消息传递等并行框架。

[0037] 容器，类似于虚拟机，是一种软件沙箱，也是一种安全机制，主要为运行中的程序提供的隔离环境，严格控制容器中的程序所能访问的资源。Linux Namespaces机制为实现基于容器的虚拟化技术提供了很好的基础，容器就是利用这一特性实现了资源的隔离，不同容器内的进程属于不同的Namespace，彼此透明，互不干扰。容器是操作系统级别的轻量级虚拟化技术，而且它底层依赖的技术Linux命名空间(Namespace)、Linux控制组(Control Group，C Group)完全是内核特性，没有任何中间层开销，对于资源的利用率极高，性能接近物理机。优选地，本发明中的第一容器和/或第二容器例如是基于操作系统虚拟化技术在宿主机上建立的具有隔离环境的软件沙箱。

[0038] 元数据(Metadata)，又称中介数据、中继数据，为描述数据的数据(data about data)，主要是描述数据属性的信息，用来支持如指示存储位置、历史数据、资源查找、文件记录等功能。

[0039] 实施例1

[0040] 本实施例公开了一种面向并行应用的容器通信方法。在不造成冲突或者矛盾的情况下，其他实施例的优选实施方式的整体和/或部分内容可以作为本实施例的补充。

[0041] 根据一个优选实施方式，参见图1，该方法可以包括：在第一容器C1中的第一进程P1需要和第二容器C2中的第二进程P2进行通信且第一和第二容器C2处于同一个宿主机HM1上的情况下，建立一个不同于基于TCP协议的第一信道的第二信道，第一容器C1将通信数据发送到第一容器C1在宿主机HM1上申请的共享内存区域并将通信数据的元数据通过第一信道发送给第二容器C2，在第二进程P2根据接收到的元数据确认接收通信数据的情况下，通过第二信道将通信数据传递给第二容器C2且通过第一信道将接收数据的确认消息反馈给第一进程P1。优选地，第二信道可以基于成熟的现有通信协议。本领域技术人员可以根据需要从现有通信协议中进行选取。

[0042] 根据一个优选实施方式，该方法还可以包括：在第一容器C1将通信数据发送到共享内存区域之前，判断第一容器C1和第二容器C2是否在预设的可信容器列表之中，若第一容器C1和第二容器C2均在该预设的可信容器名单中，则开启该第一容器C1和该第二容器C2的通信权限，否则，第一容器C1和/或第二容器C2请求用户赋予通信权限。

[0043] 根据一个优选实施方式，第一信道采用第一加密机制，第二信道采用不同于第一

加密机制的第二加密机制。

[0044] 根据一个优选实施方式，该方法还可以包括：由第一容器C1根据通信数据的重要程度划分安全等级且在通信数据的安全等级超过预设安全阈值之时，第二信道采用不同于第二加密机制的第三加密机制，其中，第二加密机制采用三重对称加密，第三加密机制采用非对称加密。优选地，重要程度越高所对应的安全等级越高。

[0045] 根据一个可选的实施方式，该方法还可以包括：在第二信道采用不同于第二加密机制的第三加密机制之前，由宿主机分析当前操作系统的第一运行环境的第一安全情况，在第一运行环境异常之时，第二信道才采用第三加密机制，在第一运行环境安全之时，第二信道仍采用第二加密机制。

[0046] 根据一个可选的实施方式，该方法还可以包括：在第二信道采用不同于第二加密机制的第三加密机制之前，由宿主机获取当前操作系统的第一运行环境的第一安全情况、第一容器C1的隔离的第二运行环境的第二安全情况和第二容器C2的隔离的第三运行环境的第三安全情况，在第一运行环境、第二运行环境和第三运行环境中的至少一个存在异常之时，第二信道才采用第三加密机制，在第一运行环境、第二运行环境和第三运行环境全为安全状态之时，第二信道仍采用第二加密机制。

[0047] 优选地，在通过第二信道传递通信数据之前，第二进程P2通过加密的第一信道向第一进程P1传输对称密钥，第一进程P1先使用对称密钥将通信数据加密后再通过第二信道将通信数据传递给第二容器C2。

[0048] 根据一个优选实施方式，该方法还可以包括：对第一容器C1和第二容器C2所处的位置进行判断，其可以包括：在第一进程P1和第二进程P2需要进行通信的情况下，第一进程P1尝试通过调用操作系统的Socket编程接口与第二进程P2建立通信；当第一进程P1和第二进程P2通过Socket编程接口实现消息传递时，获取第一容器C1的第一IP地址和第二容器C2的第二IP地址并结合对应的子网掩码计算第一容器C1的第一网络号和第二容器C2的第二网络号，在第一网络号和第二网络号相同的情况下确认第一容器C1和第二容器C2处于同一个宿主机HM1上。

[0049] 根据一个优选实施方式，建立一个不同于基于TCP协议的第一信道的第二信道的处理包括：在确认第一容器C1和第二容器C2处于同一个宿主机HM1上之后，先等待第一进程P1与第二进程P2之间完成建立基于TCP协议的第一信道，然后根据第一信道的TCP连接的第一语义信息建立一个能用于对通信数据进行传输的第二信道。

[0050] 根据一个优选实施方式，该方法还可以包括：根据接收数据的确认消息确定第一进程P1与第二进程P2之间是否存在进一步的数据交换，若是，则在确认了第一容器C1的第一IP地址和第二容器C2的第二IP地址均未发生变化的情况下，继续使用共享内存区域、已建立的第一信道和第二信道按照后续通信数据通过第二信道传输且后续通信数据的元数据通过第一信道传输的方式完成第一进程P1和第二进程P2的数据通信；若否，则先协商释放第二信道，并在第二信道释放后才释放第一信道。

[0051] 实施例2

[0052] 根据一个优选实施方式，本发明公开了一种面向并行应用的容器通信系统，该系统适于执行本发明记载的各个方法步骤，以达到预期的技术效果。

[0053] 本实施例可以是对实施例1的进一步改进和/或补充，重复的内容不再赘述。在不

造成冲突或者矛盾的情况下，其他实施例的优选实施方式的整体和/或部分内容可以作为本实施例的补充。

[0054] 根据一个优选实施方式，该系统可以包括：至少一个处理器以及存储若干指令的至少一个计算机可读存储介质，若干指令可以包括在由至少一个处理器执行时执行以下操作的至少一条指令：在第一容器C1中的第一进程P1需要和第二容器C2中的第二进程P2进行通信且第一和第二容器C2处于同一个宿主机HM1上的情况下，建立一个不同于基于TCP协议的第一信道的第二信道，第一容器C1将通信数据发送到第一容器C1和/或第二容器C2在宿主机HM1上申请的共享内存区域并将通信数据的元数据通过第一信道发送给第二容器C2，在第二进程P2根据接收到的元数据确认接收通信数据的情况下，通过第二信道将通信数据传递给第二容器C2且通过第一信道将接收数据的确认消息反馈给第一进程P1。

[0055] 根据一个优选实施方式，若干指令还可以包括在由至少一个处理器执行时执行以下操作的至少一条指令：对第一容器C1和第二容器C2所处的位置进行判断，其可以包括：在第一进程P1和第二进程P2需要进行通信的情况下，第一进程P1尝试通过调用操作系统的Socket编程接口与第二进程P2建立通信，当第一进程P1和第二进程P2通过Socket编程接口实现消息传递时，获取第一容器C1的第一IP地址和第二容器C2的第二IP地址并结合对应的子网掩码计算第一容器C1的第一网络号和第二容器C2的第二网络号，在第一网络号和第二网络号相同的情况下确认第一容器C1和第二容器C2处于同一个宿主机HM1上。

[0056] 根据一个优选实施方式，建立一个不同于基于TCP协议的第一信道的第二信道的处理可以包括：在确认第一容器C1和第二容器C2处于同一个宿主机HM1上之后，先等待第一进程P1与第二进程P2之间完成建立基于TCP协议的第一信道，然后根据第一信道的TCP连接的第一语义信息建立一个能用于对通信数据进行传输的第二信道。

[0057] 根据一个优选实施方式，计算机程序指令和/或若干指令还可以包括在由至少一个处理器执行时执行以下操作的至少一条指令：根据接收数据的确认消息确定第一进程P1与第二进程P2之间是否存在进一步的数据交换，若是，则在确认了第一容器C1的第一IP地址和第二容器C2的第二IP地址均未发生变化的情况下，继续使用共享内存区域、已建立的第一信道和第二信道按照后续通信数据通过第二信道传输且后续通信数据的元数据通过第一信道传输的方式完成第一进程P1和第二进程P2的数据通信；若否，则先协商释放第二信道，并在第二信道释放后才释放第一信道。

[0058] 根据一个优选实施方式，第一容器C1将通信数据发送到第一容器C1和/或第二容器C2在宿主机HM1上申请的共享内存区域的处理可以包括：识别第一进程P1通过Socket编程接口将通信数据发送到内核的状态，然后由驱动接口先将通信数据从内核拷贝到共享内存区域的。

[0059] 根据一个优选实施方式，通过第二信道将通信数据传递给第二容器C2是通过调用驱动接口将通信数据拷贝到第二进程P2的进程空间来实现的。

[0060] 根据一个优选实施方式，计算机程序指令和/或若干指令还可以包括在由至少一个处理器执行时执行以下操作的至少一条指令：在通过第二信道传递通信数据之前，第二进程P2通过加密的第一信道向第一进程P1传输对称密钥，第一进程P1先使用对称密钥将通信数据加密后再通过第二信道将通信数据传递给第二容器C2。

[0061] 根据一个优选实施方式，对于本发明中提及的任何方法实施方式，计算机程序指

令和/或若干指令中均可以包括在由至少一个处理器执行时执行与该方法对应的操作的至少一条指令。

[0062] 根据一个优选实施方式，对于本发明中提及的任何方法实施方式，至少一个处理器以及至少一个计算机可读存储介质均可被配置为执行与之对应的操作。

[0063] 实施例3

[0064] 本实施例可以是对实施例1、2或者其结合的进一步改进和/或补充，重复的内容不再赘述。在不造成冲突或者矛盾的情况下，其他实施例的优选实施方式的整体和/或部分内容可以作为本实施例的补充。

[0065] 根据一个优选实施方式，本实施例公开了一种面向并行应用的容器通信方法。

[0066] 优选地，该方法可以包括：在运行着并行应用的容器集群之中，当第一容器C1中的第一应用对应的第一进程P1需要与第二容器C2中的第二应用对应的第二进程P2进行通信时，第一进程P1可以通过尝试调用操作系统的Socket编程接口与第二进程P2建立通信连接。优选地，本发明提及的操作系统可以是UNIX、XENIX、LINUX、Windows、Netware和Mac中的至少一种。优选地，比如，在操作系统是Linux的情况下，第一容器和/或第二容器可以是基于Docker技术、Singularity技术和LXC技术中的至少一个建立的。其中，LXC是指Linux Container。

[0067] 根据一个可选的实施方式，参见图2，本发明的方法可以通过加载在系统内核上的内核扩展模块来实现，该内核扩展模块可以包括路由模块11、协议模块12和驱动模块13。比如，在各宿主机的操作系统的系统内核上各加载该内核扩展模块，在第一容器和第二容器需要进行通信的情况下，由路由模块11获取第一容器的第一IP地址和第二容器的第二IP地址；在内核扩展模块根据第一IP地址和第二IP地址判断第一容器和第二容器均处于宿主机的情况下，路由模块11根据TCP连接的语义信息建立一个能用于对通信数据进行传输的第二信道；第一进程通过Socket编程接口将通信数据发送到内核，由路由模块11将用户数据拷贝到第一进程和第二进程均能访问的共享内存区域，并将与通信数据对应的元数据通过TCP连接发送到第二进程；当第二进程根据元数据确认接收通信数据的情况下，路由模块11将通信数据从元数据中指定的内存地址通过第二信道拷贝到第二进程的进程空间。优选地，第二信道是根据协议模块12建立的。优选地，第一容器和第二容器都运行在加载了内核扩展模块的系统内核上。

[0068] 根据另一个可选的实施方式，参见图3，本发明的方法可以通过PCI扩展卡10B实现。优选地，该方法可以包括：使用至少一块外接的PCI扩展卡10B，该PCI扩展卡10B通过宿主机的PCI插槽与宿主机连接，其中，该PCI扩展卡10B可以包括路由模块11、协议模块12和驱动模块13。优选地，路由模块11、协议模块12和驱动模块13可以是专用集成电路(ASIC)、FPGA、或者任何其他硬件等同物。优选地，第一容器和第二容器都运行在连接了该PCI扩展卡10B的系统内核上。

[0069] 优选地，该方法还可以包括：当第一进程P1和第二进程P2通过Socket编程接口建立通信时，第一容器C1所处的宿主机HM1上的路由模块11获取第一进程P1的第一IP地址和第二进程P2的第二IP地址。

[0070] 优选地，该方法还可以包括：宿主机根据第一IP地址和第二IP地址判断第一容器C1和第二容器C2所处的位置，并在第一容器C1和第二容器C2均处于宿主机HM1上的情况下，

由路由模块11采用共享内存的形式协助第一进程P1和第二进程P2进行通信数据传输。

[0071] 根据一个优选实施方式，路由模块11采用共享内存的形式协助第一进程P1和第二进程P2之间进行通信数据传输之时，协议模块12根据TCP连接的语义信息建立一个能用于对通信数据进行传输的第二信道。

[0072] 根据一个优选实施方式，当第一进程P1将通信数据通过Socket编程接口发送到内核时，由路由模块11将通信数据拷贝到申请的内存区域，并通过TCP连接将与通信数据对应的元数据发送到第二进程P2。通过该方式，本发明将通信数据和其元数据进行分开传输，仅仅将数据传输通过共享内存的形式完成，而将数据的元数据通过TCP连接来传输，不仅保证了传输过程的稳定性，还提高了传输过程的安全性。

[0073] 根据一个优选实施方式，当第二进程P2收到元数据后，第二进程P2通知路由模块11将通信数据从元数据中指定的内存地址拷贝到第二进程P2的进程空间，并通过TCP连接传送接收数据的确认消息。

[0074] 实施例4

[0075] 本实施例可以是对实施例1、2、3或者其结合的进一步改进和/或补充，重复的内容不再赘述。在不造成冲突或者矛盾的情况下，其他实施例的优选实施方式的整体和/或部分内容可以作为本实施例的补充。

[0076] 根据一个优选实施方式，本发明公开了一种面向并行应用的容器通信系统。该系统可以包括至少一个处理器以及存储计算机程序指令的至少一个计算机可读存储介质，计算机程序指令包括在由至少一个处理器执行时执行以下操作的指令：通过PCI插槽调用PCI扩展卡10B，在运行着并行应用的容器集群之中，当第一容器的第一进程需要与第二容器中的第二进程进行通信时，使第一进程尝试通过调用其所处操作系统的Socket编程接口与第二进程建立通信连接；当第一进程和第二进程通过Socket编程接口实现消息传递时，通过PCI扩展卡10B的路由模块11获取第一进程的第一IP地址和第二进程的第二IP地址；

[0077] 然后根据第一IP地址和第二IP地址判断第一容器和第二容器是否均处于宿主机，若是，由路由模块11采用共享内存的形式协助第一进程和第二进程之间进行数据传输，若否，则通过默认的虚拟网桥通信方式实现第一进程和第二进程之间的数据传输。所有现有使用TCP协议进行通信的应用程序都可以在不加任何修改的情况下，使用本发明提供的方法进行加速，实用性、操作性更强，更易于被开发者所接受。当容器中运行的是非高性能并行应用时，PCI扩展卡10B也能检测到同主机容器间通信，转而采用PCI扩展卡10B提供的内存通信方式进行通信，从而优化进程间的通信效率。因此本发明不仅可以优化同一主机中不同容器里的高性能并行应用的通信效率，也可以优化普通应用如Web服务、数据库服务等的通信效率。

[0078] 根据一个优选实施方式，当第一进程将用户数据通过Socket编程接口发送到内核时，可以由路由模块11将用户数据拷贝到申请的内存区域，并将与用户数据对应的元数据通过TCP连接发送到第二进程。本发明采用了一种基于TCP连接的第一信道和非TCP连接的第二信道相结合的方式进行数据传输方式，而非直接替换掉TCP连接的方法。因此本发明提供的方法具有更好的兼容性，可以兼容现有的大部分程序，另外，采用TCP连接进行数据控制信息的传输，在一定程度上可以提高信息传输的可靠性，保证整个数据传输过程的正确性，因此本发明相比于现有的方法，稳定性和可靠性具有更好的保证。

[0079] 根据一个优选实施方式，当第二进程收到元数据后，路由模块11将用户数据从元数据中指定的内存地址拷贝到用户的进程空间，并通过TCP连接传送数据确认消息。

[0080] 实施例5

[0081] 本实施例可以是对实施例1、2、3、4或者其结合的进一步改进和/或补充，重复的内容不再赘述。在不造成冲突或者矛盾的情况下，其他实施例的优选实施方式的整体和/或部分内容可以作为本实施例的补充。

[0082] 优选地，本实施例公开了一种面向并行应用的容器通信方法。

[0083] 根据一个优选实施方式，参见图4，该方法可以包括以下步骤中的至少一个：

[0084] 步骤S1：对于运行着并行应用的容器集群，当其中的一个本地进程需要与另一个目标进程进行通信时，该本地进程将会尝试通过调用Linux操作系统的Socket编程接口进行通信连接的建立；

[0085] 步骤S2：当该本地进程和目标进程通过Socket通信信道进行消息传递时，路由模块11获取Socket连接的本地进程的IP地址和目标进程的IP地址，并由此判断目标进程所在容器的所在位置，若本地进程和目标进程的容器处于同一宿主机内，则PCI扩展卡10B看将会采用共享内存的形式进行数据的传输；

[0086] 步骤S3：采用共享内存方式进行数据传输时，协议模块12会根据TCP连接的语义信息建立一个第二信道，该第二信道主要用于用户数据的传输；

[0087] 步骤S4：当该本地进程将用户数据通过Socket接口发送到内核时，PCI扩展卡10B将会负责将数据拷贝到申请的内存区域，并将数据的元信息通过TCP连接发送到目标进程；

[0088] 步骤S5：当目标进程通过TCP连接收到元信息后，PCI扩展卡10B将用户数据从元信息中指示的内存地址拷贝到目标进程所在的进程空间，并通过TCP连接传送数据确认消息。

[0089] 根据一个优选实施方式，步骤S1可以包括如下子步骤中的至少一个：

[0090] 对于每一个运行高性能并行应用的容器，都运行在加载了PCI扩展卡10B的Linux内核上，优选地，各高性能并行应用可以是基于MPI标准编写的；

[0091] 当一个容器中的高性能并行应用(High Performance Parallel Application，以下简称HPPA)的进程需要与另一个容器中的进程进行通信时，由于不同容器均配置了自己的网络命名空间，不同容器的TCP/IP网络协议栈进行了隔离，因此不同容器中的HPPA的进程将会通过Socket方式进行通信，本地HPPA的进程将会调用Socket API进程进行Socket连接的创建。

[0092] 根据一个优选实施方式，步骤S2可以包括如下子步骤中的至少一个：

[0093] 当本地HPPA的进程尝试创建Socket连接进行数据传输时，路由模块11将至少获取本地进程的IP地址、目标进程的IP地址和端口信息，通过分析本地进程的IP地址和目标进程的IP地址，判断两个进程所在的容器是否位于同一宿主机，优选地，宿主机可以是物理机和/或虚拟机；

[0094] 若目标HPC进程的容器和本地进程的容器处于同一个宿主机时，路由模块11将会采第二信道来进行MPI消息的传递；

[0095] 根据一个优选实施方式，步骤S3可以包括如下子步骤中的至少一个：

[0096] 当确定采用PCI扩展卡10B提供的共享内存通信方式时，路由模块11先等待本地进程与目标进程的TCP连接完成；

[0097] 路由模块11检测到本地进程与目标进程的TCP建立完成后，便会调用协议模块12的相关接口建立一个第二信道，该连接将会用于实际的数据传输；

[0098] 本地进程将数据通过Socket相关接口发送后，PCI扩展卡10B根据目标进程的IP地址，检测是否存在该目标进程的IP地址的第二信道，如果是，则将用户数据通过第二信道进行发送，如果否，则建立相应的第二信道；

[0099] 目标容器所在的驱动模块13会申请一块本地进程所在容器和目标进程所在容器均可以访问的内存区域，并将本地进程的数据拷贝到该内存区域中，并将数据的元信息通过TCP连接发送到目标进程，优选地，元信息包括数据的位置和大小；

[0100] 根据一个优选实施方式，步骤S4可以包括如下子步骤中的至少一个：

[0101] 目标进程通过TCP连接接收到来自本地进程发送的元信息；

[0102] 目标进程的所在宿主机的路由模块11检测到数据包的源IP地址信息，发现该数据包的IP地址与目标进程IP处于同一宿主机上，于是将TCP连接中的元信息取出，并将其提交到该宿主机的PCI扩展卡10B的协议模块12；

[0103] 协议模块12根据TCP连接接收到的元信息解析出的语义信息，至少得到数据所在内存空间的地址、数据的大小和数据的类型；

[0104] 协议模块12根据元信息，调用驱动接口将对应的数据拷贝到用户程序的数据空间；

[0105] 拷贝结束后，并通过TCP连接发送数据确认信息给本地进程。

[0106] 根据一个优选实施方式，步骤S5可以包括如下子步骤：

[0107] 本地进程的TCP连接接收到目标进程的确认信息后，路由模块11获取目标进程的IP地址信息；

[0108] 若目标进程的IP地址和本地进程的IP地址处于同一宿主机中，那么将检测该确认信息，判断是否存在进一步的数据交换，若是，则重复S2至S4中的步骤进行下一步的数据传输工作，若否，则协商进行释放第二信道，当第二信道释放后，TCP连接也随之释放。

[0109] 实施例6

[0110] 本实施例可以是对实施例1、2、3、4、5或者其结合的进一步改进和/或补充，重复的内容不再赘述。在不造成冲突或者矛盾的情况下，其他实施例的优选实施方式的整体和/或部分内容可以作为本实施例的补充。

[0111] 根据一个优选实施方式，本实施例公开了一种面向高性能并行应用的容器通信系统。

[0112] 根据一个优选实施方式，该系统可以包括PCI扩展卡10B，该PCI扩展卡10B可以包括如下几个子模块：路由模块11，该模块主要用于判断源进程所在容器和目标进程所在容器的相对位置信息；协议模块12，当确定源进程与目标进程处于同一宿主机时，根据实际的TCP连接语义转换为用于建立第二信道的协议语义，提供相关接口调用；驱动模块13，为协议模块12提供接口实现，具体实现了第二信道的建立、释放、数据传输和确认等接口；和网络接口14。优选地，网络接口14可以是虚拟网络接口，用于进行网络数据包的接收和发送。

[0113] 根据一个优选实施方式，路由模块11可以包括：路由表：该路由表主要用于记录源进程与目标进程的IP地址信息和本地化信息，用于后续进行通信方式的选择，路由表由路由判断模块来维护；和/或路由判断模块：该模块是系统的关键模块之一，主要负责路由表

的维护工作，当源进行与目标进行通信，尝试建立TCP连接时，该路由模块11会获取到相应的源IP地址和目标IP地址信息，并查询路由表中是否存在相关的信息，如果不存在源IP至目标IP和目标IP至源IP地址的记录，那么路由判断模块根据IP地址和子网掩码信息判断两个IP地址是否处于同一宿主机，若两者处于同一宿主机中，那么将这条记录添加到路由表中，如果不处于同一宿主机当中，那么默认不做任何处理，每次添加完路由表记录后，进行一次无效路由记录的清理工作。

[0114] 根据一个优选实施方式，协议模块12可以包括：语义解析模块：主要负责根据TCP连接的状态信息来进行第二信道的创建和释放，数据的传输和接收工作；和/或数据结构模块：该协议模块12主要负责提供第二信道相关的基本数据结构。

[0115] 根据一个优选实施方式，驱动模块13可以包括：数据传输子模块，负责实现协议模块12中数据传输相关接口，提供具体的数据拷贝、数据获取、数据传输等工作；和/或内存管理子模块，提供了容器间共享内存区域的申请、释放、管理等接口调用。优选地，语义解析模块、路由判断模块、数据结构模块和内存管理子模块可以是专用集成电路(ASIC)、FPGA、通用计算机或者任何其他硬件等同物。

[0116] 虽然已经详细描述了本发明，但是在本发明的精神和范围内的修改对于本领域技术人员将是显而易见的。这样的修改也被认为是本公开的一部分。鉴于前面的讨论、本领域的相关知识以及上面结合背景讨论的参考或信息(均通过引用并入本文)，进一步的描述被认为是不必要的。此外，应该理解，本发明的各个方面和各个实施例的各部分均可以整体或部分地组合或互换。而且，本领域的普通技术人员将会理解，前面的描述仅仅是作为示例，并不意图限制本发明。

[0117] 已经出于示例和描述的目的给出了本公开的前述讨论。这并不意图将本公开限制于本文公开的形式。在前述的具体实施方式中，例如，为了简化本公开的目的，本公开的各种特征在一个或多个实施例、配置或方面中被组合在一起。实施例、配置或方面的特征可以以除上面讨论的那些之外的替代实施例、配置或方面组合。本公开的该方法不应被解释为反映本公开需要比每个权利要求中明确记载的更多特征的意图。相反，如以下权利要求所反映的，创造性方面在于少于单个前述公开的实施例、配置或方面的所有特征。因此，以下权利要求由此被并入本具体实施方式中，其中每个权利要求其自身作为本公开的单独实施例。

[0118] 而且，虽然本公开的描述已经包括对一个或多个实施例、配置或方面以及某些变型和修改的描述，但是其他变型、组合和修改也在本公开的范围内，例如在本领域技术人员的技能和知识范围内，在理解了本公开之后。旨在获得在允许的程度上包括替代实施例、配置或方面的权利，所述权利包括那些要求保护的替代的、可互换的和/或等效的结构、功能、范围或步骤的权利，无论这种替代的、可互换的和/或等效的结构、功能、范围或步骤是否在本文中公开，并且无意公开奉献任何可专利的主题。

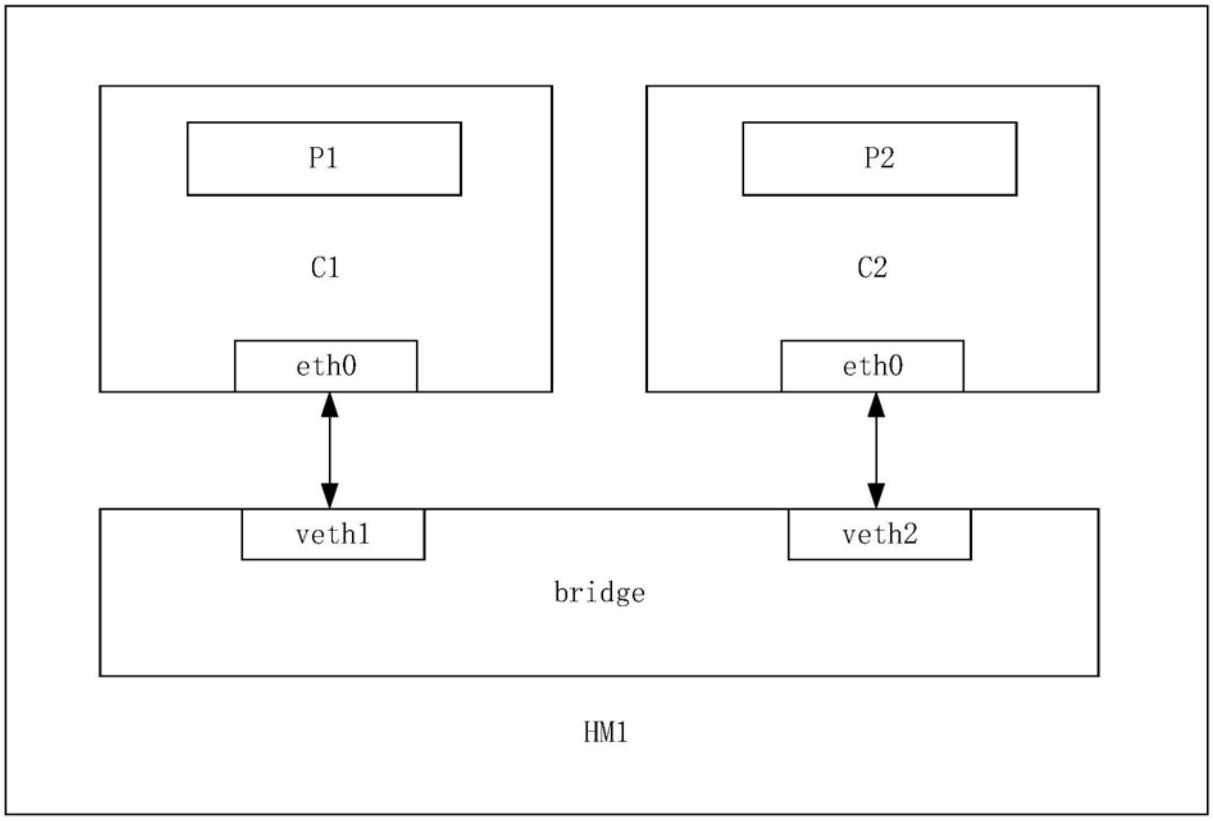


图1

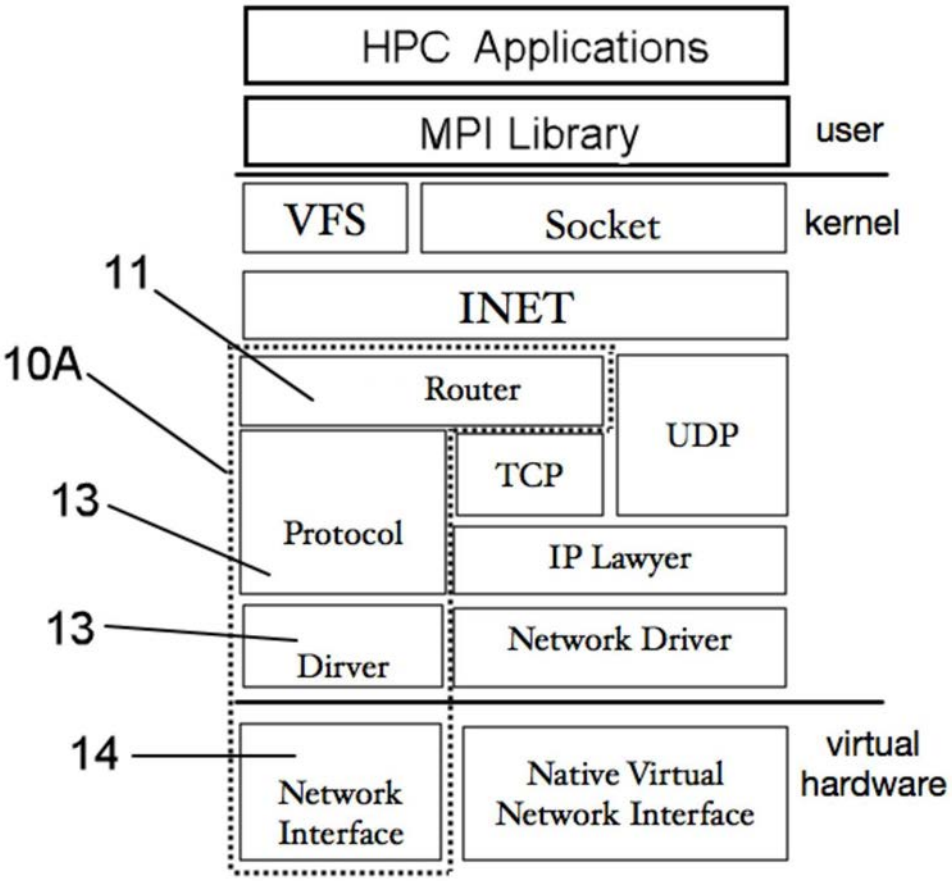

HPC Applications
MPI Library
user
VFS
Socket
kernel
INET
11
10A
Router
UDP
TCP
13
Protocol
IP Lawyer
13
Dirver
Network Driver
14
Network Interface
Native Virtual Network Interface
virtual hardware

图2

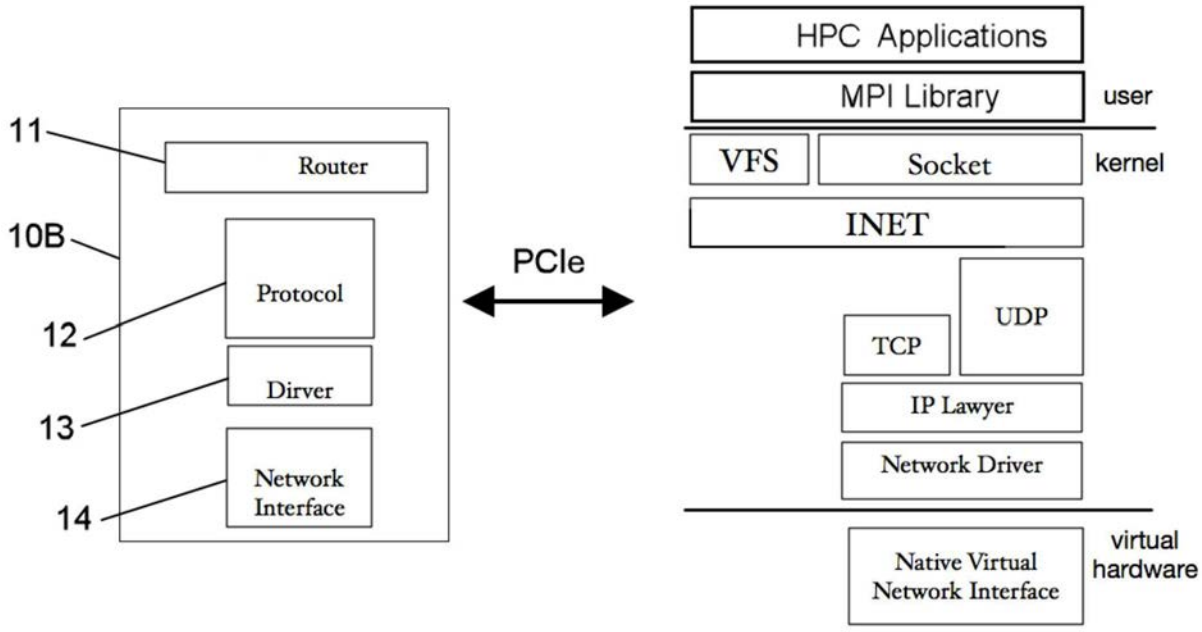

11
Router
10B
Protocol
12
Dirver
13
Network Interface
14
PCIe
HPC Applications
MPI Library
user
VFS
Socket
kernel
INET
UDP
TCP
IP Lawyer
Network Driver
Native Virtual Network Interface
virtual hardware

图3

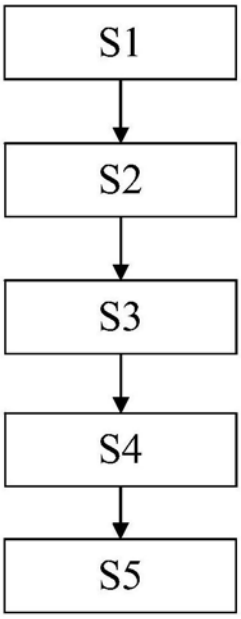


图4