计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(1):53−60 [doi: 10.15888/j.cnki.csa.006746]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

IPoIB 在国产并行系统上的实现与优化①

李　祎, 陈淑平, 何王全

(江南计算技术研究所, 无锡 214083)
通讯作者: 李　祎, E-mail: liyi_csnt@163.com

摘　要: IPoIB 是一种在 InfiniBand 网络上支持 IP 的协议, 使 IP 应用程序可以运行在 InfiniBand 网络上. 我们在国产并行系统上实现了 IPoIB, 通过乱序处理、内存拷贝优化、网络参数调优和避免应答延迟的优化手段, 实现了 IPoIB 在国产并行系统上的性能提升. 实验结果表明, IPoIB 在国产并行系统上正确运行, 网络带宽与优化前相比提高近 6 倍, 与 10GbE 万兆以太网相比, IPoIB 更具优势, 乱序处理机制减少乱序效果明显.

关键词: IPoIB; 网络优化; 网络参数; 内存拷贝; 带宽

引用格式: 李祎,陈淑平,何王全.IPoIB 在国产并行系统上的实现与优化.计算机系统应用,2019,28(1):53–60. http://www.c-s-a.org.cn/1003-3254/6746.html

Realization and Optimization of IPoIB on Domestic Parallel System

LI Yi, CHEN Shu-Ping, HE Wang-Quan

(Jiangnan Institute of Computing Technology, Wuxi 214083, China)

Abstract: IPoIB is a protocol that supports traditional Ethernet over InfiniBand networks, allowing IP applications to run on InfiniBand networks. We realize IPoIB on the domestic parallel system to improve the performance of IPoIB on the system, we also propose four optimization methods, namely, reordering packets, optimizing memory copy, tuning network parameters, and avoiding delayed acknowledgement. Practice shows IPoIB runs correctly on the domestic parallel system, and under the methods of optimum, IPoIB's network bandwidth performance is nearly 6x higher than the not-tuned version. IPoIB has more advantages compared with 10 GbE, and the reordering method shows obvious effects.

Key words: IPoIB; network optimizing; network parameters; memory copy; bandwidth

最新公布的 HPC top 500 显示, 有 207 台超级计算机使用以太网作为互连网络, 这表明 TCP/IP 是广泛使用的网络协议. 但随着众多学科领域对网络性能需求的不断提升, TCP/IP 协议中频繁数据拷贝、复杂协议处理机制和中断上下文切换逐渐成为数据传输瓶颈. 而 IB[1](InfiniBand) 相比传统以太网具有高带宽低延迟的通信性能优势, 更能满足上层应用需求. 将 IB 网络与传统以太网结合, 可以同时利用两者优势以满足不同的网络需求, 这是当今网络融合趋势下的一种研究方向, 其中 IPoIB[2](IP over InfiniBand) 和 SDP 实现了在 IB 网络之上对 TCP/IP 协议的支持, iWarp 和 RoCE 实现了在以太网上对 RDMA (Remote Direct Memory Access) 传输技术的支持. 这几种技术和传统的 TCP/IP 协议相比, 均能获得更高的网络性能, 且各有优势, 使用者可以根据具体应用场景选择合适的技术.

IB 是一种高性能、低延迟的基于通道的高速互连结构标准, 支持 RDMA 技术, 具有零拷贝以及 CPU 负载卸载的特点, 能够有效减少系统 CPU 和内存的开销,

① 基金项目: 国家重点研发计划"高性能计算"重点专项 (2016YFB0200502)
Foundation item: National Key Research and Development Program of China（2016YFB0200502）
收稿时间: 2018-07-24; 修改时间: 2018-08-21; 采用时间: 2018-08-29; csa 在线出版时间: 2018-12-07

提高网络吞吐量并降低网络延迟. IPoIB 是一种在 IB 网络之上构建 TCP/IP 的技术, 隐藏 IB 网络的复杂性, 使得 TCP/IP 应用程序可以不加修改地在以 IB 协议为基础的网络之上运行, 同时还能利用 IB 网络特有的优势以获得更好的网络传输性能[3]. 目前 IPoIB 主要应用于商用 x86 集群服务器中, 官方维护组织 OFED 不断地从数据处理、网络管理、服务支持等角度对 IPoIB 进行优化更新, 但尚未有将 IPoIB 应用于国产众核服务器系统中的实例.

众核处理器具有计算能力强、性能功耗比高等突出优点, 异构众核架构已成为当前超级计算机体系结构的重要发展方向. 本文基于“国产异构众核并行系统”展开, 对 IPoIB 进行移植, 并在已有优化方法上研究进一步提升 IPoIB 网络性能的手段, 对于支持 TCP/IP 应用具有重要的意义.

文章的后续部分组织如下: 第 1 节介绍国产并行系统平台环境, 第 2 节给出相关工作介绍, 第 3 节简要介绍 IPoIB 的实现, 第 4 节详细阐述 IPoIB 在国产并行系统上的优化手段, 第 5 节是实验结果, 最后对本文进行总结.

1 国产异构众核系统

“国产并行系统”[4]运算单元采用面向高性能计算的众核处理器, 包括 Intel 的 MIC、Nvidia 和 AMD 的 GPU、Godson-T、申威众核处理器等. 申威众核处理器包含 4 个 Core-Groups (CGs), 每个 CG 包含一个 MPE (Management Processing Element, 主核)、一个 8*8 的 Computing Processing Element (CPE, 从核) cluster 和一个 Memory Controller (MC), 4 个 CG 通过片上网络 (NoC) 互连, 处理器通过 System interface (SI) 连接外部设备. 申威众核处理器的主核和从核共享 Memory, 从核采用轻量级的核心设计, 配备由软件管理的高速存储器 SPM (Scratch Pad Memory), 支持通过 DMA (Direct Memory Access) 方式在 Memory和 SPM 间批量传输数据.

运算系统采用申威众核处理器构建, 通过中心交换网络和管理网络与存储系统和管理系统连接, 系统的登陆界面和存储空间采用单一映像组织, 为用户提供统一的视图.

2 相关工作

数据中心网络一般采用 TCP/IP 协议并基于以太网技术搭建, 近年来, 为了满足数据中心网络对高带宽、低延迟、低能耗的需求, 减少 TCP/IP 协议处理对 CPU 产生的负担, 业内研究者重点着眼于 TCP/IP 协议卸载技术. TOE (TCP 卸载引擎) 技术利用硬件分担 CPU 对 TCP/IP 协议处理所造成的负担, 将协议的处理放到网卡专用硬件中, 使 CPU 占用率大幅下降, 带宽性能也有一定提升, 但由于网卡和主机接口的不兼容性和延迟增加的额外消耗, 其发展难以维系.

研究者又从新型高速网络 (IB、FC 等) 与传统 TCP/IP 协议结合以提升网络性能的角度出发, 提出 iWARP、RoCE、SDP 等网络融合技术: iWARP (Internet Wide Area RDMA Protocol)[5]是一种在 TCP/IP 协议栈之上实现 RDMA 的技术, 实现了远程数据的直接存取, 数据可以直接放入上层协议的接收缓存区, 避免了不必要的内存拷贝, 大大提升了时延和带宽性能, 可用于广域网间 RDMA 通信; RoCE (RDMA over Converged Ethernet)[6]是一种在以太网数据链路层之上实现 RDMA 的技术, 一般建立在无损以太网之上, 和 iWARP 相比降低了复杂性和部署难度, 简化了管理, 且通信性能比 iWARP 略好[7], 但造价更高; SDP (Socket Direct Protocol)[8]是一种在 InfiniBand 可靠连接之上实现字节流传输的技术, 利用 IB 网络中 send/receive 和 RDMA 等操作, 为应用程序提供 socket 套接字接口调用, 与传统 TCP/IP 相比, 拥有协议卸载、旁路核心、零拷贝的特点, 与 IPoIB 相比, 更好地利用了 IB 网络的高速通信能力, 但由于 SDP 需要另外搭建 socket 库且难以进行网络管理, 已逐渐被淘汰.

IPoIB 作为在 IB 中兼容 TCP/IP 应用的主要技术, 其性能优化一直是研究热点, 研究人员从 IPoIB 的多个方面对其进行了数据处理方面的优化探索: 包括支持 LRO、aRFS、RSS、TSS 等 CPU 负载卸载技术、优化中断处理流程、实现隧道卸载、利用多个 pkeys 实现 VLAN 等, 实现了 IPoIB 在多数据流多处理器下的性能提升.

上述工作都是关于网络融合技术在商用平台上的研究进展, 目前在国产系统上尚未有网络融合技术的应用. 本文做了一种新的研究尝试, 将 IPoIB 协议应用到国产并行系统中, 对移植后的 IPoIB 进行性能测试和评估, 并且使用一系列优化方法来提高 IPoIB 在国产并行系统中的通信性能.

3 IPoIB 在国产并行系统上的实现

IB 是 IBTA 提出的一种基于通道的高速互连结构标准, 可提供低延迟、高带宽的数据传输能力, 在 HPC 领域具有广泛的应用. 它采用 RDMA 编程语义, 为用户提供 IB verbs 编程接口, 但其语义和编程方法与 Socket 编程语义有非常大的差异, 传统的 TCP/IP 应用不能在其上直接运行. IPoIB 解决了该问题, 它将 HCA 卡虚拟成网卡设备, 使通用的 TCP/IP 应用程序不加修改地在 IB 网络中运行, 拓展了 IB 应用领域及范围.

3.1 IPoIB 协议原理

图 1 给出了 IPoIB 协议架构. IPoIB 协议位于内核层, 处于 TCP/IP 协议栈之下、IB 传输层之上. 用户层应用程序调用 socket 套接字接口将数据送进内核层 TCP/IP 协议栈, IPoIB 通过注册 net_device 结构以及一系列设备操作函数为上层 TCP/IP 协议或 ARP 协议提供网络设备传输接口, 设备操作函数利用 IB 的诸多 verbs: QP (Queue Pair) 队列对、CQ (Completion Queue) 完成队列以及 MR (Memory Regions) 地址空间注册信息等与远程节点交互, 通过 IB 的 send/receive 操作实现数据传送接收及处理. 可以将 IPoIB 看作为以太网的数据链路层.

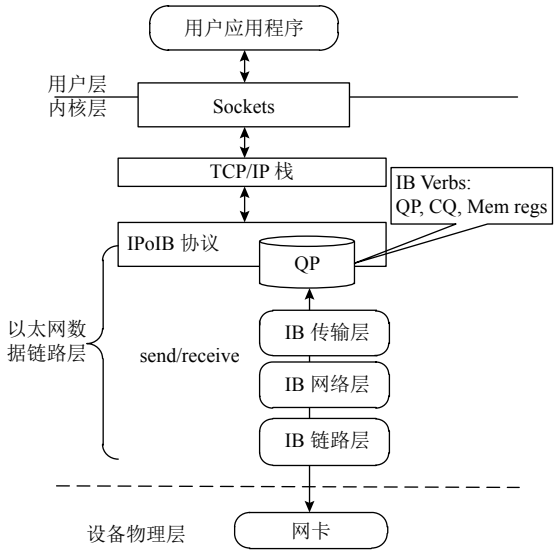

图 1 IPoIB 协议架构图

IPoIB 分别利用两种 IB 服务进行数据传输, 其中 RC (reliable connected) 是面向连接的可靠传输服务, 最大可发送 2 GB 大小的数据; UD (Unreliable Datagram) 是不可靠的数据报服务, 一次最多发送 4 KB 数据.

IPoIB 主要实现下列功能: (1) 地址解析: 将 IB 地址信息编码进 MAC 地址, 通过 ARP 获取目的方 MAC 地址, 进而获得 QPN (Queue Pair Number)、LID 等 IB 地址信息; (2) 报文封装: IPoIB 为 IP 报文添加 4 字节的链路层包头, 并封装成一条 IB 消息, 通过 IB 网络发送给远程节点, 完整的 IPoIB 数据包如图 2 所示; (3) 多播支持: IB 多播在 IPoIB 中起关键作用, ARP 协议必须通过 IB 多播实现. IPoIB 定义了多播 GID 和对应的多播组, 并启动多播组任务处理组播列表的变化、响应多播请求, 使节点自由加入多播组或从多播组中删除.

3.2 IPoIB 协议实现方式

图 3 为 IPoIB 数据包在协议栈中的处理流程[9], 包括数据发送和接收过程.

发送过程: 应用程序通过 sys_socketcall 系统调用进入 socket 内核层, socket 层再通过 sock_sendmsg 函数调用进入 inet_sendmsg 函数, 然后调用 TCP 层发送函数 tcp_sendmsg, 该函数在准备好 sk_buff 结构后调用 skb_do_copy_data_nocache 将用户数据拷贝到内核层, 然后数据包依次通过 IP 层、设备驱动层, 最后利用 IPoIB 驱动中的 ipoib_hard_header 函数为 ip 报文添加 4 字节报文头, 然后调用 ipoib_start_xmit 函数, 将数据内存地址进行 DMA 映射, 并通过 IB 的 send 操作将数据发送出去.

接收过程: 应用进程调用 sys_socketcall 系统调用进入内核层, 再通过 socket 层的接收函数 sock_recvmsg 进入到 TCP 层, TCP 调用 tcp_recvmsg 函数接收数据, 当没有数据包到来时, 用户接收进程会休眠. IPoIB 驱动程序通过 NAPI 轮询函数 ipoib_poll 检查是否有数据到达, 并对数据包进行正确性检验, 去掉 IPoIB 层链路头, 然后通过 netif_receive_skb 函数将数据依次移交给 IP 层、TCP 层处理. TCP 层再通过 skb_copy_datagram_iovec 将内核层的数据拷贝到用户层缓冲区.

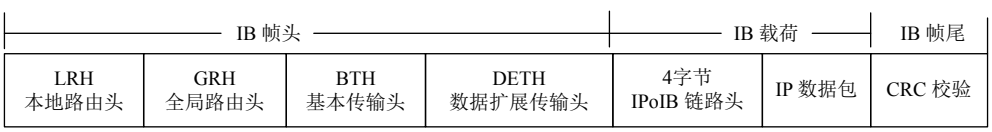

图 2 IPoIB 数据包格式图

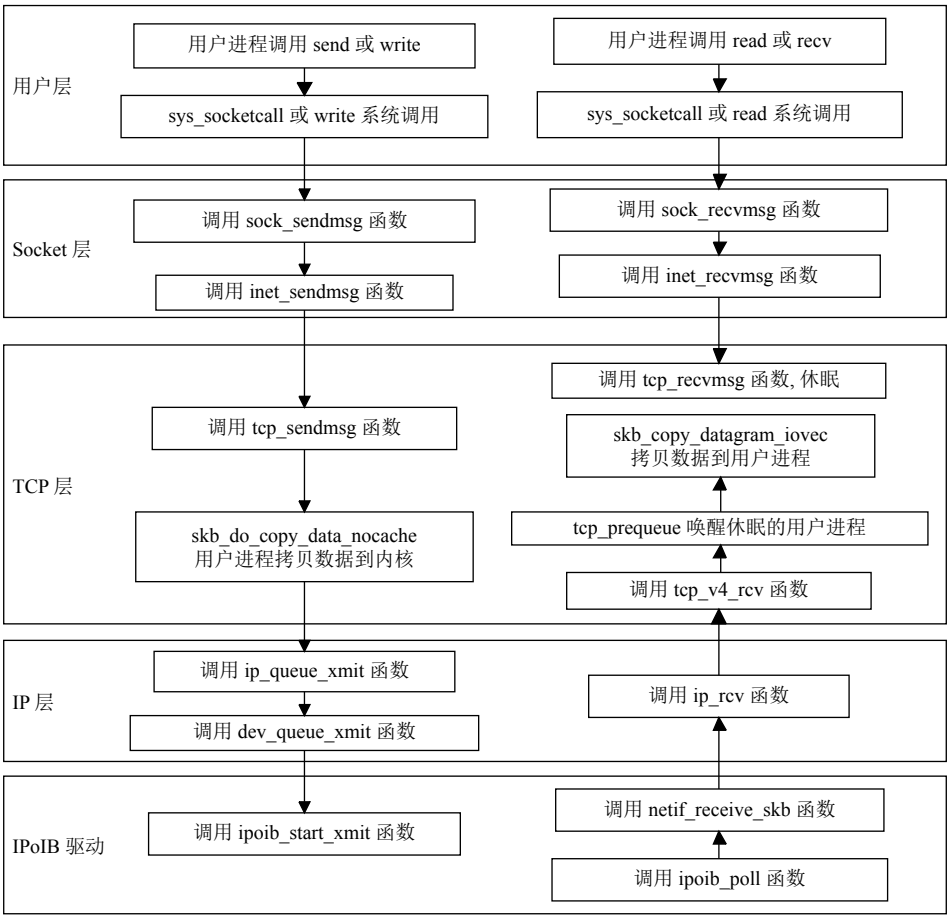

图 3 IPoIB 数据包协议处理流程图

4 IPoIB 在国产并行系统上的优化

IPoIB 数据包在 TCP/IP 协议栈处理过程中会占用较多的 CPU 时间和内存资源, 主要包括数据拷贝、协议处理机制、延迟应答等 3 个方面. 数据拷贝包括数据包在网卡和内核空间以及内核空间与用户空间之间的拷贝; 协议处理机制包括复杂的拥塞控制[10]、超时重发机制[11]、TCP 协议完整性校验、数据包分发和整合等; 延迟应答可以推迟 TCP ACK 的发送, 以便使 ACK 与数据包一起发送. 本章针对国产并行系统上 IPoIB 的性能瓶颈点, 对这三个方面的处理流程进行了优化.

4.1 乱序处理

非确定性路由中数据包沿不同的路径到达接收端以及数据包在发送单元中调度顺序与到达顺序不一致等因素都会导致接收端产生乱序数据包[12]. 在高速网络中, 发送端的发送窗口很大, 频繁的乱序数据会导致重发大量的数据包, 致使性能大幅下降. 针对乱序数据包对网络性能产生的影响, 提出一种重排序的解决方案: 每对连接发送的数据包都携带一个 16 位的序列号 *msg_id*, 接收端按照序列号进行重排序, 如果数据包是顺序的, 则直接将该数据包交上层协议栈处理; 如果是乱序的, 则进行缓存. 这种方案的缺点在于若网络中乱序情况严重, 顺序的数据包迟迟不到达, 接收端长时间缓存乱序数据包引起超时, 进而导致上层 TCP 重传数据, 反而致使性能下降更严重, 得不偿失, 且缓存大量乱序包会消耗过多的系统内存. 为了防止上述现象出现, 采用“尽可能”保序的手段, 规定接收端对数据包进行重排序的窗口长度大小限制在 W, 对不在窗口内的数据包不进行乱序处理, 预期序列号 *next_id* 始终指向窗口左边界的位置. 具体见如图 4 所示的乱序处理方法.

窗口长度 W 的大小设置原则以在一次网卡硬件收包中断发生时, NAPI 设备轮询函数所能处理的报文数量上限 NAPI_POLL_WEIGHT 为参考, 一般为 32 或 64. 这种设置原则是有依据的, 在一些中断处理速度较慢的网卡设备中, 若是窗口长度 W 超过了一次轮询

报文数量上限, 乱序较重情况下, 乱序数据包很可能被缓存较长时间, 直到下一次中断触发处理新到来的数据包, 从而造成网络性能下降.

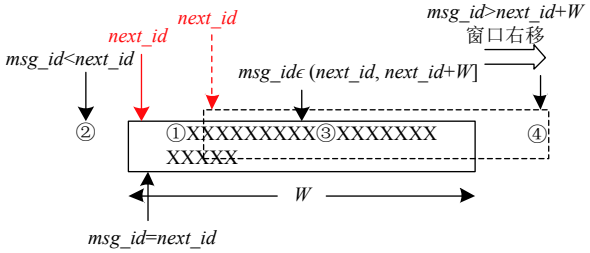

图 4 乱序处理示意图

详细算法描述如算法 1.

算法 1. 乱序处理算法

```
INPUT:
到达接收端的数据包序号 msg_id
预期序列号 next_id←0
BEGIN
IF(msg_id=next_id)
BEGIN
DO{
将数据包转交给上层;
next_id← next_id+1;
}WHILE(next_id 位置已缓存数据包)
END
ELSE IF(msg_id<next_id)
将数据包转交给上层;
ELSE IF(next_id<msg_id≤next_id+W)
暂时缓存数据包;
ELSE IF(msg_id>next_id+W)
DO{
IF(next_id 位置已缓存数据包)
BEGIN
将该数据包转交给上层;
END
next_id← next_id+1;
}WHILE(msg_id>next_id+W)
END
```

4.2 优化内存拷贝

优化内存拷贝是提高 IPoIB 性能的重要手段. 在图 3 的 IPoIB 数据包处理流程走向图的指导下, 逐一查看每个功能模块函数调用的局部时间, 发现 RC 模式下用户层到内核层数据拷贝速率较慢. 分析发现 TCP 协议在 skb_do_copy_data_nocache 函数将用户层数据拷贝进内核层时, 需要对数据进行正确性校验, 这会对网络性能造成损耗. 而不计算数据校验和时, 内存拷贝速率明显提高. 一般来讲服务器均已实现拷贝内存的“纠单错、报双错”机制, 同时 IB 也实现了对数据进行校验的功能, 因此本文将 RC 模式下计算数据校验和的工作由 TCP 协议转交给 IB 网卡硬件完成, 从而减少 TCP 协议计算检验和的工作, 减轻 CPU 负担, 实现了 IPoIB 用户层数据到内核层拷贝的优化.

4.3 通过调整网络参数优化协议栈处理流程

网络参数调优是提高 IPoIB 性能的手段之一, 通过调整内核参数达到提升 IPoIB 性能的目的.

图 5 给出了网络参数优化模型. 网络参数按照功能分为以下几类: (1) 资源分配: 诸如信道分配、队列大小、缓冲区空间等, 科学合理配置资源可以提高网络的利用率, 有助于 IPoIB 运行时吞吐量的显著提升; (2) 任务调度: 包括中断缓和、传输队列轮询等, 网络任务调度直接影响网络系统的负载均衡, 对 IPoIB 的平均时延有一定影响; (3) 协议运行时设置: ACK 应答机制、延时机制、时间戳等, 优化 IPoIB 的多种协议运行机制, 可以不同程度上提高网络带宽以及网络系统整体性能.

网络参数调优对内核环境影响较小, 但是网络性能提升效果却较为明显. 需要注意的是, 内核参数多种多样, 参数的设置受具体的服务器系统环境影响, 需要根据实际网络应用负载对参数进行适应性调整, 在寻找最优参数中必须分清主次, 突出改善主要参数, 使网络尽可能得到有效利用. 针对具体的输入数据流, 本文利用 sysctl 命令设置网络参数, 试验了网络参数优化模型中不同的参数变量值和不同的参数组合, 找出能为 IPoIB 带来性能提升的参数组合及其数值. 根据试验结果, 给出令 IPoIB 通信性能最优的配置策略: (1) 启用 tcp_low_latency, 要求 TCP/IP 栈在高吞吐量的情况下尽量保持低延时; (2) 将 rmem_max 和 wmem_max 设为 16 MB, 增大 TCP 连接在发送和接收 ACK 期间所能处理的最大数据包, 从而减少传输等量数据过程中处理 ACK 所用时间; (3) 将 tcp_rmem 和 tcp_wmem 中的 max 值设为 16 MB. 目的在于增大 socket 发送和接收缓冲区内存空间, 使一块内存空间存放尽量多的数据, 减少为数据分配多个内存块的管理开销, 优化数据传输流.

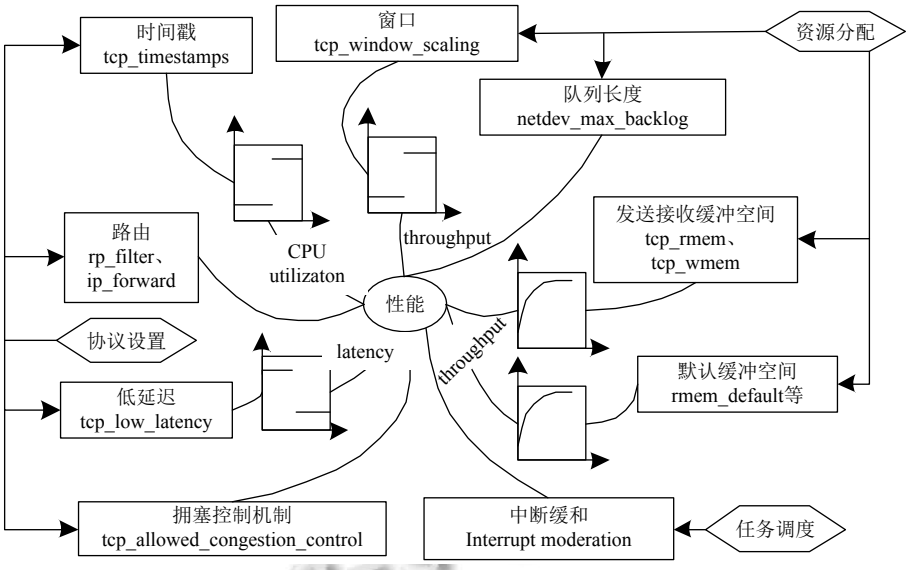

图 5 网络参数优化模型图

4.4 避免应答延迟 (Delayed ACK)

TCP 采用应答延迟机制时, 如果当前时间与最近一次接收数据包的时间间隔小于延迟应答超时时间, 则会推迟 ACK 应答的发送, 积攒多个应答并将它们结合成一个响应包, 与需要沿该方向发送的数据一起发送, 从而减少协议开销. 然而, 在应用程序进行交互处理时, 延迟 ACK 应答时间过长可能会降低应用程序的效率. TCP 协议中利用宏定义 TCP_DELACK_MIN 控制最小延迟确认时间, 一般默认值为 (HZ/25), 也就是 40 ms. 本文在不改变其它参数的情况下, 逐一试验在 1 ms~40 ms 范围内不同的 TCP_DELACK_MIN 值, 并测试网络最大带宽, 发现最小延迟应答时间设为 5 ms 左右时, 网络带宽可以达到最大, 既能维持较低协议开销, 又可以减少 TCP 传输中 ACK 的等待时间, 使得网络带宽最大化, 提升内存的利用率. 最佳的延迟应答时间受服务器系统环境和网络应用场景的影响, 本文所得结果在其它集群系统中可能不是最优, 但其试验方法具有普适性.

5 实验结果

本文在国产异构众核系统上对 IPoIB 进行测试, 配备 32 GB 内存, 节点间采用 40 GB/s 的 Infiniband EDR 网络连接. 网络性能测试工具选用 Netperf-2.4.5 和 Iperf-2.0.2. Netperf 主要用于记录两对联节点间 TCP 单连接带宽、延迟、CPU 利用率、内存等资源的占用情况, Iperf 用于记录两对联节点之间 TCP 多连接带宽.

5.1 优化效果分析

5.1.1 不同消息大小带宽对比

利用 Netperf 工具测试两对联节点间单个 TCP 连接优化前后不同消息大小带来的带宽变化, 结果如图 6 所示.

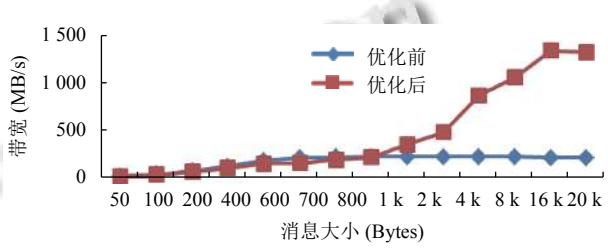

图 6 不同消息大小带宽对比图

从图 6 可以看出, IPoIB 通信性能优化效果良好, 优化后的 IPoIB 网络带宽明显高于优化前. 由测试数据可知, 优化后的峰值带宽达到 1340 MB/s, 对比优化前的 227 MB/s 提升近 6 倍. 可见针对 IPoIB 的优化对带宽具有较好提升效果, 使得系统的 IB 网络资源得到尽可能的使用, 提升了 IPoIB 在国产异构并行系统上的运行效果, 证明优化方法足够有效.

5.1.2 多连接带宽对比

使用 Iperf 测试两节点间多个 TCP 连接的网络最大带宽, 从 2 个连接测到 10 个连接, 如图 7 所示.

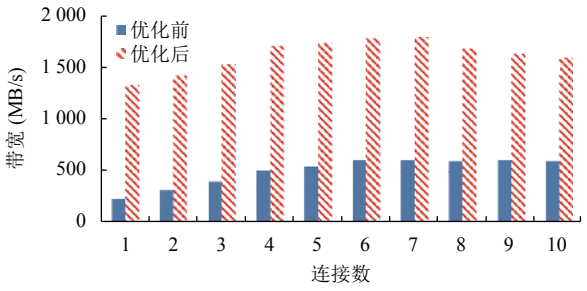

图 7 不同连接数带宽对比图

测试数据表明, 连接数为 6 或 7 时, 网络带宽达到最大. 优化后的网络带宽可以达到 1800 MB/s, 相比单连接最大带宽 1340 MB/s 提升 1.34 倍, 相比优化前多连接的网络最大带宽 615 MB/s, 提高近 3 倍. 可以看出, 在多个进程下网络带宽能有较大提升, 主要原因是: (1) 并行系统的 CPU 多核组得到了有效利用; (2) 通过在传输过程中对数据包进行抓取, 发现在其中某个 TCP 连接等待 ACK 不发送数据包时, 另一个 TCP 连接不用等待 ACK, 继续发包, 从而使得链路带宽得到有效利用.

5.1.3 CPU 利用率对比

在高速网络环境下, CPU 的处理能力很大程度上影响网络的性能. 通过测试发现, 优化前 IPoIB CPU 利用率为 33%, 优化后的 IPoIB CPU 利用率在 24% 左右, 优化后的 CPU 利用率明显低于优化前. 在最好的情况下, IPoIB 优化后的 CPU 利用率可以降低约 30%.

5.2 与万兆以太网的性能对比

利用 Netperf 测试优化后的 IPoIB 在国产并行系统上两节点间通信的带宽和延迟, 并与万兆以太网对比. 万兆以太网卡型号 T520, 10 GB/s, 理论带宽为 1250 MB/s, 万兆网卡测试环境采用国产中标麒麟服务器, 处理器型号为申威 1621. 测试结果如表 1 所示.

表 1 IPoIB 与万兆以太网性能对比

测试项	带宽 (MB/s)	延迟 (μs)
IPoIB	1340	39
万兆以太网	1024	30

从表 1 可以看出万兆以太网的稳定带宽在 1024 MB/s 左右, 而 IPoIB 优化后的带宽达 1340 MB/s, 是万兆以太网持续带宽的 1.31 倍; 万兆以太网延迟平均约为 30 μs, IPoIB 延迟平均 39 μs, 相比万兆以太网延迟高 9 μs. 基础性能对比结果表明, IPoIB 在国产异构众核并行系统上的通信性能相比基于 10 GbE 的万兆以太网通信性能要略显优势, 持续带宽优于万兆以太网, 延迟虽然比万兆以太网略大, 但实际应用中聚合带宽是主要考虑因素, 因此 IPoIB 相比 10 GbE 是更好的选择.

5.3 乱序处理效果分析

利用 Iperf 测试国产并行系统上两节点间通信处于不同流量负载时的乱序情况, 乱序处理窗口长度 W 设为 32, 为消除偶然因素影响, 每个负载下运行 10 次取平均值. 表 2 显示了在不同流量负载下乱序处理前后每秒发生的乱序次数并统计乱序减少比例. 可以看出, 乱序处理对于减少乱序数据包的作用效果十分明显: 网络流量较高时乱序情况比较严重, 而经过乱序处理后乱序次数明显减少, 乱序减少比例可达 95% 以上; 当网络负载较轻时, 经过乱序处理后网络不再有乱序包.

表 2 不同流量负载下乱序处理效果

发送速率 (MB/s)	乱序处理前后乱序次数		乱序减少比例 (%)
	处理前 (次/s)	处理后 (次/s)	
1800	3815	184	95.2
800	1411	37	97.4
400	469	0	100

为了验证乱序较重时窗口长度 W 设置过大对性能造成的不利影响, 每隔一段时间让发送方故意丢弃一次数据包以模拟乱序较重的情况, 利用 Iperf 测试窗口长度分别为 32 和 80 的最大网络带宽, 结果如表 3 所示. 从表 3 可以看出, 乱序较重时, W 为 32 的最大带宽为 1782 MB/s, 乱序程度减少 95.8%; W 为 80 的最大带宽为 1624 MB/s, 即便乱序减少效果更好, 但带宽下降明显. 实验结果证明窗口长度不宜设置过大, 过大反而会造成带宽性能下降.

表 3 不同窗口长度的网络性能对比

窗口长度	乱序减少比例 (%)	最大网络带宽 (MB/s)
32	95.8	1782.32
80	98.3	1624.71

6 结束语

本文将 IPoIB 移植到国产异构众核并行系统上, 并进行了乱序处理、拷贝优化、网络参数调优以及应答延迟避免等一系列优化措施. 测试结果显示, 优化后 IPoIB 基础带宽峰值性能为 1340 MB/s, 比优化前 IPoIB 带宽提升近 6 倍, 也高于 10 GB 万兆以太网; 多连接下带宽达到 1800 MB/s, 相比单连接提升 1.34 倍;

CPU 利用率也有了显著降低; 乱序处理机制作用效果明显.

IPoIB 基于 IB 的 send/receive 异步消息机制实现, 而没有利用具有零拷贝、CPU 负载卸载优势的 RDMA 机制, 考虑到在一些特定的应用场景下利用 RDMA 实现 IPoIB 的通信效果可能会更好, 后续将制定以 RDMA 为底层通信机制的 IPoIB 实现策略, 以期进一步提高 IPoIB 通信性能.

参考文献

1 徐迪威, 余焯佳. InfiniBand 高速互连网络设计的研究. 电脑与电信, 2012, (7): 26–29. [doi: 10.3969/j.issn.1008-6609.2012.07.025]

2 刘爱华, 钱德沛, 董小社, 等. IPoIB 体系结构及其应用. 计算机科学, 2003, 30(9): 85–88. [doi: 10.3969/j.issn.1002-137X.2003.09.025]

3 朱叶青, 牛德姣, 蔡涛, 等. 不同网络环境下大数据系统的测试与分析. 江苏大学学报 (自然科学版), 2016, 37(4): 429–437. [doi: 10.3969/j.issn.1671-7775.2016.04.010]

4 何王全, 刘勇, 方燕飞, 等. 面向国产异构众核系统的 Parallel C 语言设计与实现. 软件学报, 2017, 28(4): 764–785. [doi: 10.13328/j.cnki.jos.005197]

5 Dalessandro D, Devulapalli A, Wyckoff P. Design and implementation of the IWARP protocol in software. Proceedings of International Conference on Parallel and Distributed Computing Systems. Phoenix, AZ, USA. 2005. 471–476.

6 Kaur G, Kumar M, Bala M. Comparing Ethernet and soft RoCE for MPI communication. IOSR Journal of Computer Engineering, 2014, 16(4): 52–58.

7 秦宣龙, 李大刚, 都政, 等. 面向数据中心网络的高速数据传输技术. 软件, 2016, 37(9): 1–7. [doi: 10.3969/j.issn.1003-6970.2016.09.001]

8 伍卫国, 杜哲君, 刘娟, 等. InfiniBand 结构中 SDP 协议分析. 微电子学与计算机, 2004, 21(9): 144–148. [doi: 10.3969/j.issn.1000-7180.2004.09.041]

9 Wright GR, Stevens WR. TCP/IP 详解卷 2: 实现. 陆雪莹, 蒋慧, 译. 北京: 机械工业出版社, 2000. 680–803.

10 孔金生, 任平英. TCP 网络拥塞控制研究. 计算机技术与发展, 2014, 24(1): 43–46.

11 王敏杰, 徐昌彪, 刘光明. 无线网络下 TCP 重传定时器研究. 计算机工程与应用, 2004, 40(36): 146–150. [doi: 10.3321/j.issn:1002-8331.2004.36.046]

12 胡晓峰, 孙志刚, 苏金树. 基于 NewReno 拥塞控制机制的 TCP 分组乱序影响分析. 计算机工程与科学, 2009, 31(5): 8–12. [doi: 10.3969/j.issn.1007-130X.2009.05.003]