

基于飞腾平台 TOE 协议栈的设计与实现

张志宏, 吴庆波, 邵立松, 谭郁松, 刘 刚
(国防科学技术大学 计算机学院, 湖南 长沙 410073)

摘要:传统 TCP/IP 协议栈要占用大量计算和访存资源,主要表现在中断上下文切换、协议处理和数据拷贝三方面。为减轻飞腾处理器计算负载,逐步采用软硬件一体化即协议卸载引擎(TCP/IP Offload Engine)技术,用硬件部分或全部实现 TCP/IP 协议处理。因飞腾平台处理器频率较低,网络负载较重时容易成为网络 I/O 瓶颈。文中对 TCP/IP 卸载引擎(TOE)技术及其相关原理进行研究,设计并实现了飞腾平台 TOE 协议卸载引擎的驱动,利用 TOE 对飞腾平台的网络性能进行优化。测试表明:飞腾平台使用 TCP/IP 卸载引擎能提高网络吞吐量并减少 CPU 利用率。

关键词:协议卸载引擎;网络负载;网络吞吐量;CPU 利用率

中图分类号:TP31

文献标识码:A

文章编号:1673-629X(2014)07-0001-04

doi:10.3969/j.issn.1673-629X.2014.07.001

Design and Implementation of TCP/IP Offload Engine Protocol Stack Based on FT Platform

ZHANG Zhi-hong, WU Qing-bo, SHAO Li-song, TAN Yu-song, LIU Gang
(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: Traditional TCP/IP protocol stack takes a lot of computation and memory access resources, mainly in the aspects of interrupting context switch, protocol processing and data copying. In order to reduce the computational load of processor, phased integration of hardware and software that is protocol offload engines (TCP/IP Offload Engine) technology, use some or all of the hardware to achieve TCP/IP protocol processing. FT processor's frequency is low, when the network load becomes heavy, it easily becomes a network I/O bottlenecks. Introduce the principle of TOE, and design and implement the TOE driver on domestic platform, optimizing network performance on FT platform by TOE. Tests show use of the TCP/IP offload engine significantly improves network throughput and reduces CPU utilization.

Key words: TCP/IP Offload Engine; network load; network throughput; CPU utilization

0 引言

在飞腾(FT)平台,传统 TCP/IP 协议栈需要占用大量计算和访存资源,因 CPU 处理能力有限,网络 I/O 容易成为瓶颈。通常情况,传输 1 bit 的数据需要消耗 1 Hz 的 CPU,在高速网络中 CPU 需要投入大量周期处理 TCP/IP 协议,容易造成实时性应用得不到及时处理,使得整体 I/O 性能降低。为减轻飞腾处理器在 TCP/IP 协议处理中的开销,文中引入 TCP/IP 协议卸载引擎(TCP/IP Offload Engine)技术,将 TCP/IP 协议处理卸载到硬件引擎。

1 相关研究

1.1 TCP/IP 卸载引擎技术

TCP/IP Offload Engine (TOE)的基本原理就是将 TCP/IP 协议处理部分或者全部转移到 TOE 中进行,简化了 TCP/IP 协议的处理路径,同时也能减轻 CPU 的协议处理开销,从而尽可能的消除主机端的网络 IO 瓶颈。

传统 TCP/IP 软件栈的协议处理开销主要分为中断上下文切换、数据拷贝和协议处理^[1-4]这三个方面。传统网卡将接收到的网络数据包最终传到用户空间的过程中通常需要经历多次中断处理过程(对 CPU 的硬

收稿日期:2013-09-05

修回日期:2013-12-18

网络出版时间:2014-04-24

基金项目:国家“核高基”项目(2012zx01040001)

作者简介:张志宏(1987-),男,硕士研究生,研究方向为计算机软件与理论;吴庆波,研究员,研究方向为计算机软件与理论;邵立松,副研究员,研究方向为操作系统;谭郁松,副研究员,CCF 会员,研究方向为操作系统。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140424.0812.053.html>

中断和中断下半部对接收进程的软中断)和多次数据拷贝(通过 DMA 方式先将接收的数据包传到内核空间,然后应用程序通过系统调用将数据再次拷贝到用户空间进行处理),传统的 TCP/IP 协议数据处理过程比较繁琐,需要消耗大量的 CPU 资源。而利用 TOE 技术可以有效地从这三个方面减少协议处理开销。

1.2 TCP/IP 卸载引擎硬件实现机制

TOE 网卡根据卸载的程度可分为数据路径卸载(Data-path offload)和全卸载(Full-offload)^[5-6]。数据路径卸载过程是把 TCP/IP 的数据发送和接收功能用网卡上的专用集成电路(ASIC)芯片进行处理,但 TCP 连接的建立和释放以及出错处理过程是由主机 CPU 来完成。全卸载则将协议处理相关的过程都交给网卡上的协处理器^[7-8]。全卸载方式便于特殊应用的二次开发,缺点是网卡上的协处理器容易成为网络瓶颈,没有能够从体系结构上解决瓶颈问题。文中基于飞腾平台采用数据路径卸载策略的 TOE 网卡。

文中研究 Chelsio 公司的 TOE 网卡 T420 - CR^[9-11],并基于飞腾平台在现有 Linux 网络软件栈基础上设计并实现该 TOE 网卡的驱动模块。通过 TOE 驱动模块,内核空间和用户空间的应用程序都可以直接与 TOE 网卡通讯,TOE 网卡接收到数据后,可以自动完成协议处理而不需要 CPU 的干预,这样省去 CPU 将数据拷贝到内核缓冲区和内核缓冲区到用户缓冲区的过程,避免网卡和主机间不必要的数据拷贝,减轻 CPU 负担,也极大减少了应用程序处理延时。

2 硬件功能结构

T420-CR 万兆网卡主要包括四部硬件逻辑:主机接口逻辑、通用控制处理逻辑、协议加速引擎、光收发模块,如图 1 所示。

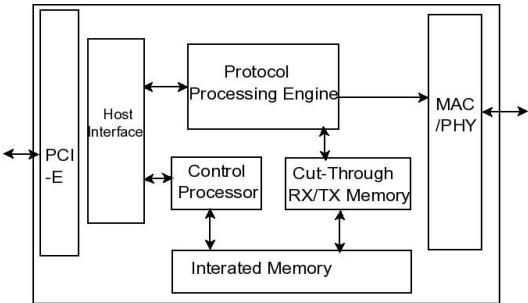


图 1 TOE 网卡结构图

主机接口逻辑:包括 DMA 引擎和门铃寄存器组。DMA 引擎用来完成主机端和网卡的数据传输,门铃寄存器组用来完成主机端和网卡的控制命令交互。

协议加速引擎:位于系统主机接口逻辑和以太网接口之间,是实现 TCP/IP 协议加速处理的核心逻辑部件。用于处理连接管理,校验和,路由表查找,拥塞控

制等任务。

通用控制处理逻辑:用来完成主机之间的命令交互,控制各个功能模块的执行。控制处理器连接着网卡片上内存,它用来保存每个卸载连接的 TCP 的状态信息,和第三层路由表及其相关结构和要处理的中间数据。光收发模块用来进行光电转换便于远距离传输。

该网卡具有普通模式(NIC)和协议加速(TOE)模式,普通模式下网络数据包还是经过传统的 TCP/IP 协议栈处理。

3 软件设计与实现

3.1 软件结构框架

现有 Linux 内核中没有对 TOE 技术的支持,需要设计卸载引擎驱动模块和应用层接口模块来与 TOE 网卡交互。为了保持应用程序的通用性,对内核中的 TCP/IP 协议栈进行部分改写,添加了 TCP/IP 卸载模块^[12-13],这个模块通过增加一个把应用层数据路由到协议卸载引擎来进行协议处理的钩子模块,实现 TOE 卸载功能^[14],如图 2 所示。



图 2 FT-TOE 驱动框架

3.2 BSD 套接字层

这一部分属于协议无关层,位于应用程序和协议栈之间,对应用程序屏蔽了协议相关实现的具体细节,将应用程序发送的协议无关的请求映射到与协议相关的实现,由此为应用程序提供一个访问网络和进程间通信的通用接口,每个套接字在内核中以 struct socket 结构体描述,该结构体中有一个 struct proto_ops 结构指针,该结构对应于传输层协议操作,正是这个结构实现了套接口层函数到传输层函数的映射。

3.3 协议卸载模块

网卡有两种工作模式,用户可以根据配置类型来决定在 TCP 建立连接时是否启用卸载连接;这里配置为启用卸载连接。通过将 inet socket 的 sk->ops 定向到 offload_inet_stream_ops 操作接口和 sk->sk_prot 和成员中的相关操作接口(建立协议卸载连接,发送,接收和释放连接等)重定向到 FT-TOE 核心层接口,然后

再调用 FT-TOE 设备驱动提供的 FT-TOE 发送、接收函数把数据路由到 FT-TOE 网卡中进行协议处理。

3.4 TOE 核心层

该层是上层网络协议栈和 FT-TOE 网卡驱动之间的接口,主要提供了类似 Linux 传统协议栈中网络核心层的功能。FT-TOE 网卡设备驱动程序将其设备底层操作方法注册到 FT-TOE 设备核心层来供上层的协议卸载模块调用。该层还可以向 Linux 传统的网络协议栈提交网卡设备驱动程序非卸载连接的传统报文、控制报文和 UDP 报文。

FT-TOE 设备核心层通过对内核软件协议栈进行修改,来为上层卸载模块提供相关卸载服务接口,并为底层 FT-TOE 设备驱动提供卸载操作接口,正如内核为驱动提供一个通用接口 struct net_device 一样,核心层提供一个 struct toedev 接口和操作函数。这一层实现了应用程序的数据路由与 FT-TOE 网卡之间的过渡。

3.5 TOE 设备驱动

FT-TOE 驱动程序直接负责管理协议卸载引擎及其相关的资源。FT-TOE 设备驱动不仅包括传统的网卡驱动程序所具备的设备注册注销、数据的发送接收、中断处理、硬件操作接口,还包括硬件连接管理、FT-TOE 设备数据读写操作、提供建立协议卸载连接、连接释放和 FT-TOE 设备的数据发送接收功能。Linux 内核的软件协议栈负责处理 UDP 协议报文和没有建立协议卸载连接的报文。为尽可能减少冗余的数据拷贝操作,该驱动在数据传输过程采用零拷贝机制来完成数据与 FT-TOE 网卡间的数据传输。

4 TOE 中的数据零拷贝传输机制

零拷贝机制(zero-copy)的基本思想是^[12]:网络数据包从网卡到用户空间传输的过程中,通过减少数据的拷贝次数,实现 CPU 的零参与,消除 CPU 在这方面的负载。零拷贝用到的最主要技术是内存区域映射和 DMA 机制。传统的网络数据包处理过程需要经过网卡缓冲区到内核空间,内核空间到用户空间这两次拷贝,同时还需要经历用户向系统发出的系统调用。零拷贝技术则首先利用 DMA 技术将网络数据包直接传递到内核预先分配的地址空间中,期间不需要 CPU 的参与;同时将内核缓存数据包的内存区域映射到用户空间。用户程序直接可以对这块内存进行访问,从而减少内核向用户空间的内存拷贝,同时减少系统调用的开销,实现真正的“零拷贝”。

4.1 协议卸载的零拷贝传输设计与实现

文中采用另外一种实现方式,首先用户程序需要分配发送/接收缓冲区,然后通过标准套接字把地址传

到内核中,接着把用户缓冲区映射到物理页中,并把映射到的物理页,偏移和长度保存到离散聚集表中,物理页在数据传输完成之前是不能被换出的,把离散聚集表首地址写到网卡寄存器中,来告诉网卡发送/接收的数据在主机中的地址。然后通过 DMA 来完成网络设备和用户空间之间的数据传输。实现过程如图 3 所示。

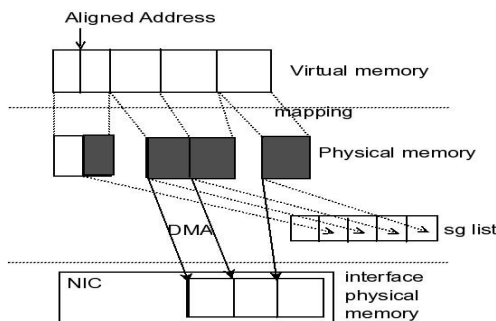


图 3 零拷贝过程图解

4.2 协议卸载的传输过程

发送过程:应用程序准备好待发送数据,协议卸载模块首先会判断其是否已经建立协议卸载连接。对于已卸载连接,用户进程把数据缓冲区交给内核来处理,通过 FT-TOE 核心层的操作接口调用驱动的 FT-TOE 数据发送接口进行发送。内核首先需要把该缓冲区映射到连续的物理页上,将数据缓冲区的虚拟地址翻译成能够进行 DMA 操作的总线地址。然后 FT-TOE 网卡驱动将发送的数据组织成各个数据块的 DMA 传输事件,并将这些事件以发送描述符的形式交付给 FT-TOE 网卡。这样网卡将用户数据直接 DMA 到网卡的发送缓冲区并依次为其封装 TCP/IP/MAC 报文头,最后经过 TCP 发送窗口控制发送,将数据移到物理层的发送缓冲进行发送。

接收过程:协议建立卸载连接后,报文不再经过内核原始协议栈处理,而是先 DMA 到网卡的缓存进行硬件协议处理。当从网络上接收数据包时,根据源和目的 IP 地址、源和目的端口号确定报文所属连接是否已经启用协议卸载。对于已建立协议卸载的连接,FT-TOE 网卡将数据移至接收缓冲区,对乱序报文进行处理。所有协议处理过程完成后,DMA 引擎将数据送至主机内存并通过中断通告 FT-TOE 网卡驱动,驱动程序通知协议卸载模块接收数据并最终直接交给应用程序。这期间 CPU 没有参与数据从应用空间到内核空间的拷贝过程。

5 性能测试

文中在飞腾平台进行性能测试。飞腾体系结构为 sparc 架构,大端字节序,有 8 个核,每个核有一个独立的执行单元,而且每个核有 4 个硬件线程,相当于有

32 个逻辑 CPU,主频 1 GHz,内存 8 G,使用 1 G 光纤模块进行直连测试。测试使用麒麟操作系统,内核版本为 2.6.32,使用 Iperf 对该网卡在传统模式和协议卸载模式下进行测试,分别统计网络吞吐量,同时利用 Top 命令分别记录网卡在传统模式和协议卸载模式下的 CPU 利用率,测试结果如图 4 和图 5 所示。

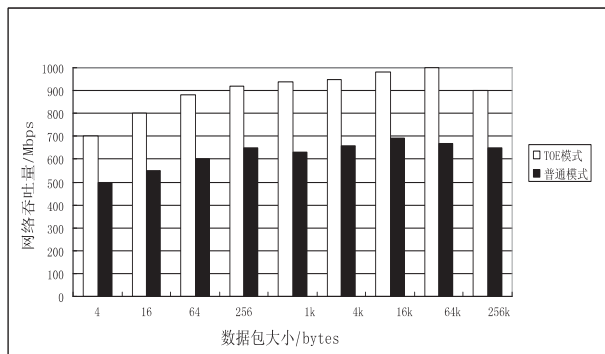


图 4 网络吞吐量对比测试图

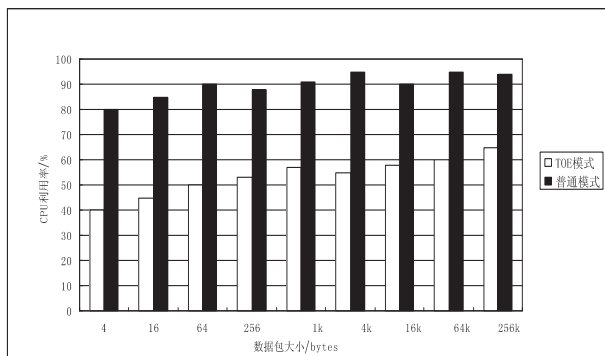


图 5 CPU 利用率对比测试图

测试表明,在相同 MTU 的情况下,TOE 网卡的网络吞吐量明显高于普通网卡,而且 CPU 利用率也会低很多。在进行数据传输时,采用 FT-TOE 模式可以达到比 NIC 模式更高的吞吐率,而且 CPU 利用率也明显低于 NIC 模式。在进行大数据流传输时,尤其是数据包大小到 1 k 以后,采用 TOE 卸载模式可以维持比普通模式更高的吞吐率,并且 TOE 模式下的 CPU 占用率也明显低于普通模式。在最好的情况下 TOE 模式可以比普通模式降低 50% 的 CPU 利用率。此次测试使用的光纤模块的吞吐量极限是 1 G 左右,这是此次实验的一个局限,但不影响最终对测试结果的分析。

实验表明,在飞腾服务器应用领域,如果用 TCP 卸载引擎网卡代替普通网卡,网络报文处理效率即可提高,一台具有 TOE 加速卡的服务器相当于若干台部署传统网卡的服务器,这在数据中心领域可以通过减少服务器的数量来显著降低能耗和成本。

6 结束语

文中通过对 TOE 技术进行研究,在飞腾平台部署

TOE 引擎,设计并实现了卸载引擎驱动和接口模块。测试表明:在飞腾平台使用 TOE 协议卸载技术可以有效地降低飞腾主机的 CPU 利用率,增加网络吞吐量,提高传输效率。

参考文献:

- [1] 王 圣,苏金树. TCP 加速技术研究综述[J]. 软件学报, 2004,14(11):1689-1699.
- [2] 赵 欣,时向泉,吴纯青. 支持 TCP/IP 卸载引擎的协议栈的设计与实现[J]. 微电子学与计算机,2006,23(Sup):132-134.
- [3] 项 敏,王学军. TCP/IP 协议栈在嵌入式芯片上的实现[J]. 电子设计应用,2004(5):67-68.
- [4] 周敬利,王志华,徐 漾,等. 基于 TCP/IP 卸载引擎的千兆网卡[J]. 计算机工程,2004,30(4):86-87.
- [5] 陈代寿. TOE:加大千兆以太网卡容量[N]. 中国计算机报, 2004.
- [6] 陈 聪. TOE 技术以及 TCP 网卡的工作原理[N]. 中国计算机报,2003.
- [7] 方捷磊,朱 杰. 在嵌入式网络应用中实现 TCP/IP 协议[J]. 微电子学与计算机,2002(5):28-30.
- [8] Binkert N L, Hsu L R, Saidi A G, et al. Performance analysis of system overheads in TCP/IP workloads[C]//Proceedings of the 14th international conference on parallel architectures and compilation techniques. [s. l.]: IEEE, 2005:218-228.
- [9] Jang H, Chung S H, Oh S C. Implementation of a hybrid TCP/IP offload engine prototype[C]//Proc of 10th Asia-Pacific conf on advances in computer system architecture. Berlin: Springer-Verlag, 2005:464-477.
- [10] Wang Wen-Fong, Wang Jun-Yau, Li Jin-Jie. Study on enhanced strategies for TCP/TP offload engines[C]//Proc of 11th international conference on parallel and distributed systems. [s. l.]: IEEE, 2005:398-404.
- [11] Chelsio T4 architecture white paper[EB/OL]. 2012. <http://line-provider.com/whitepapers/tcpip-offload-engine-toe/>.
- [12] Oh Soo-Cheol, Kim Seong-Woon. An efficient Linux kernel module supporting TCP/IP offload engine on grid[C]//Proceedings of the fifth international conference on grid and cooperative computing. Hunan: IEEE, 2006:228-235.
- [13] Kang Dong-Jae, Kim Chei-Yol, Kim Kang-Ho, et al. Design and implementation of kernel S/W for TCP/IP offload engine (TOE)[C]//Proc of the 7th international conference on advanced communication technology. Phoenix Park: IEEE, 2005: 706-709.
- [14] Kim Hyong-Youb, Rixner S. TCP offload through connection handoff[C]//Proceedings of the 1st ACM SIGOPS/EuroSys European conference on computer systems. New York, NY, USA: ACM, 2006:279-290.

作者: 张志宏, 吴庆波, 邵立松, 谭郁松, 刘刚, ZHANG Zhi-hong, WU Qing-bo, SHAO Li-song, TAN Yu-song, LIU Gang
作者单位: 国防科学技术大学 计算机学院, 湖南 长沙, 410073
刊名: 计算机技术与发展 
英文刊名: Computer Technology and Development
年, 卷(期): 2014(7)

参考文献(14条)

1. 王圣;苏金树 TCP加速技术研究综述 2004(11)
2. 赵欣;时向泉;吴纯青 支持TCP/IP卸载引擎的协议栈的设计与实现 2006(Sup)
3. 项敏;王学军 TCP/IP协议栈在嵌入式芯片上的实现 2004(05)
4. 周敬利;王志华;徐漾 基于 TCP/IP卸载引擎的千兆网卡 2004(04)
5. 陈代寿 TOE:加大千兆以太网卡容量 2004
6. 陈聪 TOE技术以及TCP网卡的工作原理 2003
7. 方捷磊;朱杰 在嵌入式网络应用中实现TCP/IP协议 2002(05)
8. Binkert N L;Hsu L R;Saidi A G Performance analysis of system overheads in TCP/IP workloads 2005
9. Jang H;Chung S H;Oh S C Implementation of a hybrid TCP/IP offload engine prototype 2005
10. Wang Wen-Fong;Wang Jun-Yau;Li Jin-Jie Study on en-hanced strategies for TCP/TP offload engines 2005
11. Chelsio T4 architecture white paper 2012
12. Oh Soo-Cheol;Kim Seong-Woon An efficient Linux kernel module supporting TCP/IP offload engine on grid 2006
13. Kang Dong-Jae;Kim Chei-Yol;Kim Kang-Ho Design and implementation of kernel S/W for TCP/IP offload engine(TOE) 2005
14. Kim Hyong-Youb;Rixner S TCP offload through connection handoff 2006

引用本文格式: 张志宏, 吴庆波, 邵立松, 谭郁松, 刘刚, ZHANG Zhi-hong, WU Qing-bo, SHAO Li-song, TAN Yu-song, LIU Gang 基于飞腾平台TOE协议栈的设计与实现 [期刊论文] - 计算机技术与发展 2014(7)