中国科学技术大学

研究生学位论文开题报告

论文题目 用户空间协议栈在通用多核服务器上的并行化和性能优化

学生姓名 别体伟

学生学号 SA13011047

指导教师 华 蓓

所在院系 计算机科学与技术

学科专业 计算机系统结构

研究方向 计算机网络

填表日期 2014-01-20

中国科学技术大学研究生院培养办公室

二零零四年五月制表

说　明

1. 抓好研究生学位论文开题报告工作是保证学位论文质量的一个重要环节。为加强对研究生培养的过程管理，规范研究生学位论文的开题报告，特印发此表。

2. 研究生一般应在课程学习结束之后的第一个学期内主动与导师协商，完成学位论文的开题报告。

3. 研究生需在学科点内报告，听取意见，进行论文开题论证。

4. 研究生论文开题论证通过后，在本表末签名后将此表交所在学院教学办公室备查。

一．选题依据

１．阐述该选题的研究意义，分析该研究课题国内外研究的概况和发展趋势。

随着硬件技术的提高，万兆网卡逐渐普及。相对于普通网卡，万兆网卡不仅在速度上有显著提升，还拥有很多普通网卡不具备的新特性，其中最突出的是多队列支持。一个物理网卡可以设置多个接收队列，网卡使用一个哈希函数（目前使用的是微软提出的哈希算法 RSS[1]）对数据包的某些域计算哈希值，然后根据哈希值将包分配到某个队列上。如果将每个队列绑定到一个固定的 CPU 核上，就可以将属于同一个流的包（如同一个 TCP 连接的包）分配到同一个 CPU 核上处理，易于实现网络程序的并行化。

然而，传统的网络软件（如网卡驱动和协议栈）在设计之初并未预见到多队列网卡的出现，因而这些软件无法充分利用万兆网卡的新特性以及多核处理器的并行处理能力。近几年，人们为充分发挥万兆网卡的新特性开发了一些高速的 raw packet I/O 软件，如 netmap[2]、Packet-IO-Engine[3]、DPDK[4]等。这些软件的共同特点是绕过内核，直接把网卡用于包传输的数据结构暴露给用户程序。这样做一方面避免了修改内核，另一方面避免了应用程序与内核之间数据拷贝带来的巨大开销。

目前这些技术已经十分成熟，如 DPDK 已经是 Intel 公司推出的商业产品。但是这些软件都只提供了 raw packet I/O 功能，并未提供 Internet 上广泛使用的 TCP/IP 协议栈。为利用高速的 raw packet I/O 软件，需要在用户空间实现协议栈。用户空间协议栈并非是一个新名词，很早以前就已提出来，并且已有许多开源的作品，如 uptcpip[6]、libuinet[7]、lwip[8]等。但是这些用户空间协议栈的提出都是为了方便编程或者用于嵌入式系统等，并非为了在多核系统中充分利用高端网卡的新特性实现高性能。

第一个基于万兆网卡（Packet-IO-Engine 软件）提出的用户空间 TCP/IP 协议栈实现是 mtcp[5]，但是 mtcp 只是一个演示性质的 TCP/IP 协议栈，实现得十分简陋[9]，不仅可靠性无法保证，很多十分重要的功能（如 IP 分片重组）也未实现。因此，mtcp 并不是一个能够实用的用户空间协议栈。此前实验室的江盛杰师兄在多核处理器上优化了 lwip 用户空间协议栈，以利用万兆网卡的特性。但是 lwip 是为嵌入式系统而开发，很多功能只是对协议的最基础实现，不如现在广泛使用的内核协议栈那么完备。并且即使在性能优化方面，也未做到在每个 CPU 核上运行一个完全独立无关的协议栈实例，即未能充分发挥万兆网卡的多队列优势。

综上所述，到目前为止尚没有可以实用的、能够充分利用万兆网卡新特性的高性能协议栈实现。

实现一个实用的、高性能的用户空间协议栈可以从两个方向着手：（1）以一个现有的协议栈为蓝本，对其进行并行化和性能优化；（2）全新实现一个协议栈，以并行化和高性能为目标进行开发。显然，后者不仅需要十分高的人力代价完成开发工作，还需要很高的时间代价来进行足够充分的测试以证明其可靠性。所以，直接以一个现有的协议栈为蓝本，对其进行适当的修改做到并行化是更切合实际的选择，故而本论文采用第一种办法。

目前协议栈虽然有很多，但最完善、最可靠的协议栈是目前广泛使用的 Linux 和 FreeBSD 等 UNIX 系统的内核协议栈，因特网上绝大部分的终端在使用这些协议栈实现。Linux 协议栈的代码模块性不好，实现较混乱；而 FreeBSD 协议栈的代码传承自最早的 BSD 协议栈，模块性很好。此外，FreeBSD 是 BSD 众多分支中以性能和稳定性见长的，其协议栈的性能和稳定性极高。Libuinet 是移植自 FreeBSD 9.1-RELEASE 内核的用户空间协议栈，保留了内核协议栈的完整功能，只是通过在用户空间实现协议栈所依赖的功能完成移植工作，所以可靠性高、功能完善。其最大的不足是性能不高，甚至不如原始的内核协议栈性能。

本论文拟以 libuinet 作为用户空间协议栈蓝本，将其移植到配置有万兆网卡的多核平台上，以实现一个实用的、高性能的用户空间协议栈。

２．国内外主要参考文献（列出作者、论文名称、期刊名称、出版年月）。

[1] Introduction to Receive Side Scaling, http://msdn.microsoft.com/en-us/library/windows/hardware/ff556942(v=vs.85).aspx
[2] Luigi Rizzo, netmap: a novel framework for fast packet I/O, Best Paper award at Usenix ATC'12, June 2012
[3] Sangjin Han, Keon Jang, KyoungSoo Park, Sue Moon, PacketShader: A GPU-Accelerated Software Router, SIGCOMM 2010, 195-206
[4] DPDK, http://dpdk.org/
[5] EunYoung Jeong, Shinae Woo, Muhammad Jamshed, and Haewon Jeong, Sunghwan Ihm, Dongsu Han and KyoungSoo Park, mTCP: a Highly Scalable User-level TCP Stack for Multicore Systems, NSDI 2014
[6] up-TCP/IP Stack Library, http://parsec.cs.princeton.edu/parsec3-doc.htm
[7] Patrick Kelsey, Userspace Networking with libuinet A portable and performant TCP/IP stack-in-a-box, BSDCan2014
[8] lwIP - A Lightweight TCP/IP stack, http://www.nongnu.org/lwip/
[9] mtcp source code, https://github.com/eunyoung14/mtcp

二．已取得的与论文研究内容相关的成果

已发表或被接收发表的文章目录或其它相关研究成果。
无

三．研究内容和研究方法

主要研究内容及预期成果，拟采用的研究方法、技术路线、实验方案的可行性分析。

1. 主要研究内容

（1） 在 Intel 多核平台上并行化协议栈，使其能够充分利用万兆网卡的多队列特性和多核 CPU 的并行计算能力。

（2） 移植一到两个应用程序到协议栈上，验证协议栈的实用性和性能。

2. 预期成果

以开源软件的形式提供一个可在通用多核服务器上运行的高性能用户空间协议栈及应用案例，并发表学术论文。

3. 研究方法及技术路线

（1）协议栈并行化和性能优化

协议栈并行化和性能优化需要解决以下问题：1）并行编程模型的选择； 2）全局数据结构转化为 per-core 的数据结构；3）协议栈接收路径的加速。

对于第一个问题，拟采用任务并行的编程模型。为万兆网卡的每个队列创建一个虚拟网卡，为每个虚拟网卡创建一个协议栈实例，将每个协议栈实例的线程绑定到一个 CPU 核上，利用网卡的 RSS 功能将属于同一个连接的包分配到同一个 CPU 核上处理。采用以上并行化方法之后，每条 TCP 连接的信息仅保存在一个 CPU 核上，不同的 CPU 核间不共享连接信息，各个核独立地进行 TCP 连接的处理。

对于第二个问题，协议栈中包含很多全局的数据结构来保存协议栈的全局信息，比如保存所有 TCP 连接信息的 TCB 表。由于协议栈在实现网络虚拟化的时候已经提供了相关机制，可以直接建立一个新的虚拟协议栈实例，其中包括协议栈所需的所有全局数据结构。这里可以利用这些工作成果，直接为每个核心建立一个虚拟协议栈实例，从而做到将原本所有 CPU 核共享的全局数据结构转化为每个 CPU 核独占的 per-core 的数据结构。

对于第三个问题，为了避免从网卡的缓冲区拷贝每个数据包的巨大开销，拟直接把保存数据包的网卡缓冲区打包成协议栈所能使用的格式交付协议栈处理。为了避免网卡缓冲区被持续占用，导致缓冲区满，网卡无法接收新的数据包，需要对协议栈接收路径进行加速，使其尽快释放所占用的缓冲区。加速手段包括以 RTC（Run-To-Complete）的模式完成对每个数据包的处理工作，以及将处理包时产生的 send 需求移

交 timer 线程来做。

（2）应用案例

拟在协议栈上移植两个网络应用，一个是自主开发的 SDN 控制器，另一个是应用代理服务器。

（3）实验方案

实验拟采用三台多核服务器，一台服务器运行服务端（即与用户空间协议栈绑定在一起的服务器程序），另外两台服务器运行客户端（即 benchmark 工具），服务器之间通过万兆网卡连接。

拟移植的第一个应用软件是实验室为华为的 POF 交换机开发的 SDN 控制器软件，将采用业界标准的 SDN benchmark 工具 cbench 对系统性能进行测试。

拟移植的第二个应用软件是一个开源的应用代理服务器，并与基于内核协议栈运行的应用代理服务器进行性能比较。

4. 可行性分析

到目前为止，已经完成了用户空间协议栈的并行化工作，并基于新的协议栈编写了简单的应用，实际运行测试表明以上技术方案是可行的。

四．课题研究的创新之处

研究内容、拟采用的研究方法、技术路线等方面有哪些创新之处。

1. 本论文系首次对一个开源、实用的用户空间协议栈在通用多核服务器上进行优化，做到能够充分利用万兆网卡的多队列特性和多核处理器的并行处理能力获得高性能。

五．研究工作进度安排

2014.09--2015.1　已完成了用户空间协议栈的并行化

2015.01--2015.02　搭建网络测试环境

2015.03--2015.07　进行初步的性能测试工作，并根据结果进行优化

2015.08--2015.11　移植 SDN 控制器，并进行性能测试和优化

2015.12--2015.01　移植应用代理服务器，并进行性能测试和优化

2016.02--2016.03　分析实验结果，得出结论

2016.04--2016.05　撰写毕业论文和准备答辩

研究生本人签名：别体伟

2014 年　01 月　20 日