软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(Suppl.(2)):18−24

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于异构网络处理平台的可配置并行路由查表算法研究*

严锦立, 吕高锋, 唐 路, 李 韬, 孙志刚

(国防科学技术大学 计算机学院,湖南 长沙 410073)

通讯作者: 严锦立, E-mail: yan_jinli@126.com

摘 要: 基于通用多核的网络转发性能难以满足高速网络流量线速处理的需求.软硬件结合的异构网络处理平台以其较高的性能和灵活性在网络处理领域得到广泛应用,但是如何基于异构平台实现高效的路由查表算法仍需进行深入研究,多核资源利用率低、共享冲突严重和访存次数多的问题是制约传统路由查表算法在异构网络处理平台实现性能提升的主要问题.为此,基于异构网络处理平台(network processing platform,简称 NPP)提出一种可配置并行路由查表机制(configurable parallel lookup,简称 CPL).CPL 中的多线程并行查找和路由表的多副本存储技术在提高多核资源利用率的同时,实现了零冲突访问路由表项.此外,考虑到不同场景下路由前缀分布的差异,CPL 支持通过配置对多级路由表的组织结构进行调整,从而有效地减少了路由表访问次数.最后在 NPP 上,对 CPL 和传统的查表算法进行性能测试和对比,验证了 CPL 的可用性和高效性.

关键词: 多核;并行;零冲突;可配置

中文引用格式: 严锦立,吕高锋 唐路,李韬,孙志刚.基于异构网络处理平台的可配置并行路由查表算法研究.软件学报, 2016,27(Suppl.(2)):18−24. http://www.jos.org.cn/1000-9825/16014.htm

英文引用格式: Yan JL, Lü GF, Tang L, Li T, Sun ZG. Research on reconfigurable parallel routing lookup algorithm based on heterogeneous network processing platform. Ruan Jian Xue Bao/Journal of Software, 2016,27(Suppl.(2)):18−24 (in Chinese). http://www.jos.org.cn/1000-9825/16014.htm

Research on Reconfigurable Parallel Routing Lookup Algorithm Based on Heterogeneous Network Processing Platform

YAN Jin-Li, LÜ Gao-Feng, TANG Lu, LI Tao, SUN Zhi-Gang

(School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: The forwarding performance of general-purpose multi-core server cannot satisfy the demand of wire-speed processing of high-speed traffic very well. Although the heterogeneous network processing platforms combining software and hardware have been applied widely in network processing field for their high performance and flexibility, the problem of designing high-efficiency routing table lookup algorithm in this platform still needs further study. The main problems restricting the performance improvement of heterogeneous network processing platform include low utilization of multi-core resources, serious shared-resources access conflict and large amounts of memory access. Thus, this paper proposes a routing lookup mechanism named CPL (Configurable Parallel Lookup) based on heterogeneous network processing platform NPP (Network processing Platform). The technologies of multi-thread parallel lookup and multi-copy management of routing tables in CPL improve the utilization of multi-core resources and accessing routing tables with zero-conflict. In addition, given that the distribution differences of routing prefix in different scenarios, CPL regulates the structure of multi-level routing table by configuration, which reduces the number of routing table accesses efficiently. At last, after making the performance comparison of CPL and traditional lookup algorithm, the experiments demonstrate the availability and high-efficiency.

Key words: multi-core; parallel; zero-conflict; configurable

* 基金项目: 国家高技术研究发展计划(863)(2015AA016103); 高性能计算协同创新中心优秀研究生创新资助项目

Foundation item: National High-Tech R&D Program of China (863) (2015AA016103); Excellent Graduate Student Innovation Funding Program of High-Performance Computing Coordination Innovation Center

收稿时间: 2015-05-31; 采用时间: 2016-01-05

目前,网络数据流量正在飞速增长,核心路由器需要处理 30 万条路由,而且路由更新频率非常快.这些因素使得通用服务器的报文处理性能不能很好地满足人们的需要[1].软硬件结合的异构网络处理平台以其较高的性能和灵活性在网络处理领域得到广泛应用[2],但是目前基于异构网络平台设计路由查表算法仍需进行深入研究.因此,在异构网络处理平台上实现一种高效并行的路由查找算法对提升网络处理的性能具有非常重要的意义.

为了实现并行路由查表算法,目前主要的方法是在用户空间使用 MPI 消息传递库、POSIX 多线程库以及一些新的语言结构对路由查表算法在应用层进行并行编码,同时在编译阶段对算法进行优化加速[3].虽然这些方法能够提升路由查表算法的性能,但是,由于多线程管理难度大经常造成不能充分利用多核资源的问题.此外,报文从内核态到用户态的数据拷贝和网卡的中断处理会占用大量的资源[4].因此,提升多核资源的利用率对提升报文处理性能具有重要意义.

在并行路由查表过程中,多个进程或线程会同时访问路由表来获取目的端口,路由表是非常关键的共享变量.对于单个内存控制单元(memory controller unit,简称 MCU),一块内存空间同一时间只能被一个线程进行访问,因此当多个线程访问同一条路由表项时会发生共享冲突,只有获得路由表访问权的线程访问完毕后,下一个进程才可以再次访问该资源,这种排队访问造成较大的访问延迟[5].所以减少对共享变量的访问延迟是提高路由查找效率的关键.

NPP(network processing platform)是一种基于 CPU/FPGA 异构网络处理平台.NPP 通过 FPGA 硬件实现缓冲区的分配和释放,能够有效减小缓冲区管理开销.NPDK(network processing development kit)是适配 NPP 的软件开发套件,通过旁路内核提供了一个零中断和零拷贝的数据面环境[6].通过缓冲区管理和轮询模式驱动等核心技术有效减少了传统网卡驱动处理数据的开销,从而有效地提高了报文的处理性能.另外,NPDK 提供了报文处理开发环境,支持多种网络应用的开发.

本文基于异构网络处理平台 NPP 提出了一种可配置并行路由查表(configurable parallel lookup,简称 CPL)算法.CPL 在内核驱动中直接创建多个线程,每个线程在轮询状态下进行 Run to Completion 模式的处理,从而有效提升了多核资源利用率.CPL 通过多副本技术[7]进行路由表的存储管理,将路由表复制多份,为每个处理线程分配一个路由表副本,通过管理线程统一下发更新信息的方式解决多副本的一致性的问题,以有效提高访存速率.此外,CPL 是一种对 DIR-24-8 路由查找算法改进的可配置两级查表算法.DIR-24-8 路由查表算法基于当时核心路由器中大部分的路由前缀不超过 24 的背景而设计的[8,9].但是,随着网络规模的不断扩大和路由聚合技术的出现,核心和边缘路由器中的路由前缀分布差异较大,使用 DIR-24-8 算法会产生大量的两次访存[10].为了使路由表组织结构更好地适配不同的应用场景,CPL 通过配置对路由表的组织结构进行调整,从而减小访存次数.

为了验证本文提出的可配置并行路由查表算法的高效性,本文在 NPP 异构网络处理平台上对 CPL 查表算法与传统 Hash 查表算法的性能进行了对比分析,实验结果表明,CPL 查表算法可以达到 520Mb/s,而 Hash 查表算法带宽只能达到 180Mb/s.

本文第 1 节对 NPP 平台进行介绍,包括 NPP 的软硬件架构和 NPDK 多核并行处理框架.第 2 节对可配置并行路由查表算法进行介绍.第 3 节在 NPP 平台中对 CPL 路由查表算法和传统 Hash 查表算法的转发性能进行测试和对比分析.最后是结束语.

1 异构网络处理平台 NPP

1.1 软硬件架构

NPP 开发平台主要由 SDB 分组处理硬件、NPDK 开发环境以及并行网络处理应用 3 部分组成,如图 1 所示.其中,SDB 逻辑功能可以划分为 3 个主要部分:Ingress 处理、Egress 处理以及管理控制.在 Ingress 端,主要实现网络接口报文接收、报文解析与分类、报文多线程分派以及报文 I/O 接收等功能;在 Egress 端,则主要实现网络接口报文发送、网络流量整形、报文 I/O 发送等功能;管理控制部分主要包括报文描述符管理、协处理器

加速、控制访问总线以及摘要及管控接口等部分.NPDK 开发环境是网络处理器的应用软件开发套件,支持 Intel 和 FT 等多种通用多核处理器,完成了网络处理器收发报文的驱动和应用适配,同时提供了系统级和用户级的报文处理开发环境;并行查表应用是指在 NPDK 开发环境中利用多线程技术实现的路由查表算法,可分内核态和用户态两种模式进行实现.

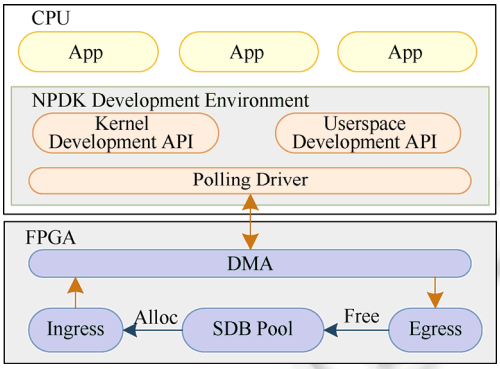


Fig.1 Architecture of NPP

图 1 NPP 架构

1.2 NPDK 多核并行处理框架

通过研究 I/O 加速软件 NPDK,发现其所提供的多核并行处理框架可以将路由查找算法并行化.如图 2 所示,NPDK 软件支持多线程,在软硬件初始化完成后,可以通过 Thread Management 模块创建多个线程并分别与 CPU 核绑定,每个线程绑定 Run to Completion 轮询处理函数.网络接口将接收到的报文通过 DMA 控制器分发到各个报文接收队列上,报文缓冲区可分为控制区和数据区,控制区存放 MetaData 信息,数据区存放原始报文数据.与队列绑定的报文处理线程上的轮询处理函数包括接收报文、路由查找、发送报文 3 个过程.从而实现了路由查找算法的并行化.

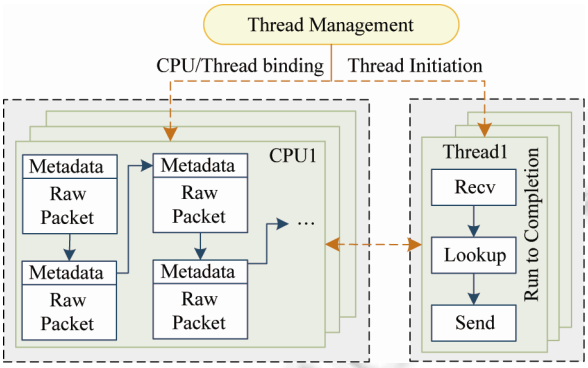


Fig.2 Multi-Core parallel processing architecture of NPDK

图 2 NPDK 多核并行处理框架

2 可配置并行路由查表算法 CPL

本文基于软硬件结合的异构网络处理平台 NPP,提出了一种可配置并行路由查表算法 CPL.一方面,CPL 算法采用了多核并行处理和多副本技术,解决了传统机制中多核资源利用率低以及共享变量访问延迟高的问题.另一方面,根据不同场景下路由表前缀分布的差异,CPL 通过配置对路由表的组织结构进行调整,有效地减少了访存次数,从而提高了查表效率.

2.1 算法思想

CPL 基于 NPDK 提供的多核并行处理框架,在 Linux 内核驱动中创建多个线程与 CPU 核进行绑定,每个线程在轮询状态下进行 Run to Completion 模式的处理.这种处理方式消除了协议栈中数据拷贝、报文格式转换和中断处理所带来的开销,从而有效地提升了多核资源利用率.

解决共享变量访问的冲突是并行处理中的主要难点.传统机制中的路由表在一块共享的内存空间中,对于单个内存控制单元,路由表在同一时间内只能被单个线程访问,多个线程同时进行访问时会发生共享冲突,排队处理会产生较高的访问延迟.CPL 中采用多副本技术,将路由表复制多份,每个线程独享一份路由表,通过多个内存控制单元同时进行访问,可以有效解决共享冲突问题.但是,多副本技术会带来一致性问题.在更新过程中,需要确保每个副本中的路由信息是最新的.同时,cache 中也存储最新的路由信息.为此,CPL 在多线程处理框架的基础上,创建了对路由表进行统一更新的管理线程,具体如图 3 所示.管理线程获取到需要更新的路由表项后,将该表项复制多份,发送到每个线程上.然后,每个线程找到对应的路由表项进行替换.最后,通过 cache 监听总线上失效的地址,如果 cache 中存在该表项,则完成 cache-line 的写失效操作.

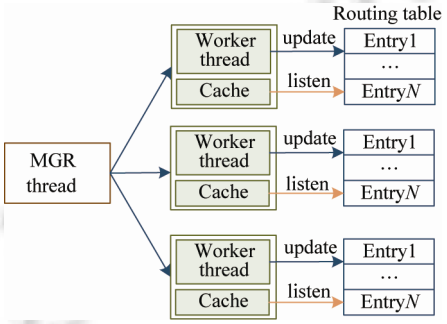


Fig.3 Management of routing table

图 3 路由表的管理

CPL 是一种对 DIR-24-8 路由查表算法进行改进的可配置两级树形查表算法.在 DIR-24-8 路由查表算法中,需要设计两级路由表保存路由信息,一级表保存前缀长度小于等于 24 的所有路由前缀信息,二级表保存前缀长度大于 24 的路由前缀信息.该算法的设计主要是考虑到当时核心路由器中大部分的路由前缀不超过 24.但是,随着网络规模的不断扩大和路由聚合技术的出现,核心路由器中很多路由前缀大于 24,并且边缘路由器中路由表项相对于核心路由器较少,路由前缀的分布也会有所不同,继续使用 DIR-24-8 算法会带来大量的二次访存.为了使路由表组织结构更好地适配不同的路由器,以减少访存次数,CPL 算法提供了配置接口,可以根据当前路由器中前缀的分布,对每一级路由表存储的路由前缀长度进行动态的配置,使大部分查表的访存次数为 1,从而提高了查表效率.

2.2 详细设计

2.2.1 关键数据结构

CPL 算法的关键数据结构包括:IP 地址、一级表 FirTable、二级表 SecTable,如图 4 所示.其中,IP 地址可以分为 3 部分.其中,FirRef 用于索引到一级表的首地址;FirBias 是一级表的偏移量,通过一级表的首地址和偏移量确定一级表项;SecBias 是二级表的偏移量,通过二级表的首地址和偏移量确定二级表项.

一级表 FirTable 的表项为 32bit,可以分为 5 部分.其中,SecRef 是二级表的索引,用于确定二级表的首地址;Outport 是输出端口;SecValid 用于判断是否有二级表;FirEmp 用于判断一级表是否为空;FirPrefix 是一级表的前缀长度.二级表 SecTable 的表项为 16bit,可以分为 3 部分.其中,SecPrefix 用于判断二级表的前缀长度;SecEmp 用于判断二级表项是否为空;Outport 是输出端口.

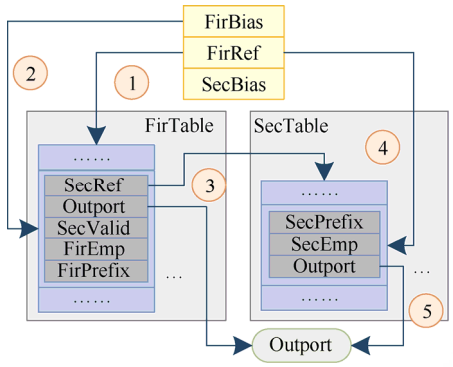


Fig.4 Routing lookup flow

图 4 查表流程

2.2.2 运行流程

CPL 算法的工作流程主要分为初始化和路由查表两个阶段,伪代码如算法 1 所示.

初始化阶段可以分为以下 3 个步骤:

a) 根据用户配置的一级表的最大长度 max_len 进行 FirTable 和 SecTable 组建;

b) 根据用户输入的线程数 *n* 进行线程初始化;

c) 根据用户线程数 *n* 进行路由表复制.

路由查表阶段可以分为以下两个步骤:

a) 根据目的 IP 地址的高 max_len 位在 FirTable 表中查找下一跳信息;

b) 若不能获取下一跳信息,则在二级表 SecTable 中继续查找.

算法 1. CPL.

输入: fir_len, thread_num, ip_addr;

输出: outport.

```
1.  FirTable=CPL_first_init(fir_len); //一级表初始化
2.  SecTable=CPL_sec_init(fir_len); //二级表初始化
3.  CPL_thread_init(thread_num); //多线程初始化
4.  CPL_table_copy(thread_num); //路由表拷贝
5.  outport=CPL_first_lookup(ip_addr,fir_len); //查找一级表
6.  IF outport!=–1 THEN
7.          RETURN outport;
8.  ELSE
9.          outport=CPL_second_lookup(ip_addr, fir_len); //查找二级表
10.         IF outport !=–1 THEN
11.                 RETURN outport;
12.         ELSE
13.                 RETURN –1
14. END IF
```

3 实验分析

本文在基于通用多核 FT1000A 处理器和 Altera FPGA 组成的异构网络处理平台 NPP 开发平台上,将 CPL 查表算法与传统 Linux 内核中的 Hash 路由查表算法进行性能测试.

3.1 测试方案

本文通过服务器将网段 1 中客户端 1 发送的报文转发到网络 2 中的客户端 2,实验拓扑如图 5 所示.在测试过程中,分别设置报文大小为 64B、128B、256B、512B、1 024B、1 500B,以测试客户端的报文发送带宽和服务

端的报文接收带宽.

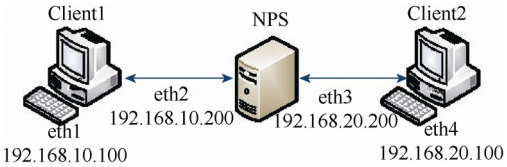


Fig.5 Topology of experiment

图 5 实验拓扑图

3.2 性能对比与分析

为了验证多副本并行路由查表机制的优越性,本文将 Hash 查表算法与 CPL 查表算法的转发性能进行对比分析.

如图 6 所示,在千兆环境下,当报文大小为 64B 时,发送带宽为 760Mb/s,CPL 查表算法的接收带宽为 520Mb/s,而 Hash 查表算法的接收带宽为 180Mb/s.此外,CPL 查表算法在报文大小为 128B 时可以达到线速转发,而 Hash 查表算法在报文大小为 512B 时才能达到线速转发.

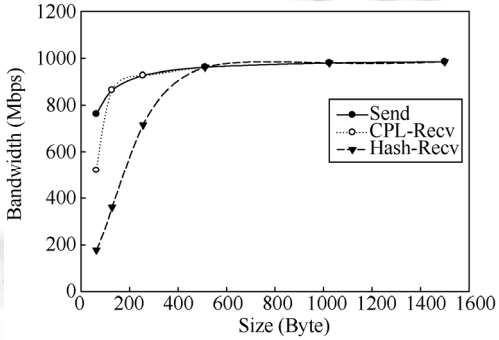


Fig.6 Performance comparison between CPL algorithm and Hash lookup algorithm

图 6 CPL 查找算法与 Hash 查表算法转发性能对比

通过分析,导致两者性能差距较大的原因主要是:(1) CPL 查表算法使用多个线程进行并行处理,而 Hash 查表算法使用单个线程进行转发处理;(2) CPL 查表算法在内核驱动中直接进行转发,不会产生协议栈处理的开销;而 Hash 查表算法在处理过程中报文会进入协议栈,有协议栈处理的开销;(3) CPL 查表转发在绝大多数情况下访存次数为 1 次,而 Hash 查表算法访存次数较多;(4) CPL 路由查表算法的复杂度为 $O(1)$,而 Hash 查表算法的算法复杂度为 $O(n)$.

4 结束语

本文从目前通用多核服务器的网络处理设备不能高效处理快速增长的流量的实际背景出发,基于异构网络处理平台 NPP,提出一种可配置并行路由查表算法 CPL.CPL 通过实现多线程并行查找和路由表项的多副本存储有效提升了多核资源的利用率,并实现了零冲突访问路由表.另外,CPL 通过提供配置多级路由表的组织结构,有效减小了路由表访存次数.性能测试结果表明,CPL 路由查找算法相比于 Linux 内核 Hash 查表,其转发性能有较大提升.综上所述,本文提出的可配置并行路由查表算法对于在异构网络处理平台下提升网络数据处理性能有非常大的现实意义.

References:

[1] Kumar S, Crowley P. Segmented hash: An efficient hash table implementation for high performance networking subsystems. In: Proc. of the 2005 ACM Symp. on Architecture for Networking and Communications Systems. ACM, 2005. 91–103.

[2] Gandhi R, Liu HH, Hu YC, *et al*. Duet: Cloud scale load balancing with hardware and software. ACM SIGCOMM Computer Communication Review, 2015,44(4):27–38.

[3] Zhou Q, Li Y. Isomorphic new parallel division methods and parallel algorithms for giant matrix transpose. Journal of Computers, 2010,5(2):169–177.

[4] Mogul JC, Ramakrishnan KK. Eliminating receive livelock in an interrupt-driven kernel. ACM Trans. on Computer Systems, 1997,15(3):217–252.

[5] Medardoni S, Ruggiero M, Bertozzi D, *et al*. Capturing the interaction of the communication, memory and I/O subsystems in memory-centric industrial MPSoC platforms. In: Proc. of the 2007 Design, Automation & Test in Europe Conf. & Exhibition. IEEE, 2007. 1–6.

[6] Tang L, Yan JL, Sun ZG, *et al*. Towards high-performance packet processing on commodity multi-cores: Current issues and future directions. Science China Information Sciences, 2015,58(12):1–16.

[7] Huang ZB, Zhu MF, Xiao LM. LvtPPP: Live-Time protected pseudopartitioning of multicore shared caches. IEEE Trans. on Parallel and Distributed Systems, 2013,24(8):1622–1632.

[8] Gupta P, Lin S, McKeown N. Routing lookups in hardware at memory access speeds. In: Proc. of the INFOCOM'98, the 17th Annual Joint Conf. of the IEEE Computer and Communications Societies. IEEE, 1998,3:1240–1247.

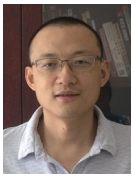
[9] Zec M, Rizzo L, Mikuc M. DXR: Towards a billion routing lookups per second in software. ACM SIGCOMM Computer Communication Review, 2012,42(5):29–36.

[10] Baboescu F, Singh S, Varghese G. Packet classification for core routers: Is there an alternative to CAMs? In: Proc. of the INFOCOM 2003, the 22nd Annual Joint Conf. of the IEEE Computer and Communications. IEEE Society, 2003,1:53–63.

严锦立(1993－),男,陕西咸阳人,硕士,主要研究领域为新型互联网体系结构,高性能路由与交换技术.

李韬(1983－),男,博士,CCF 专业会员,主要研究领域为计算机网络,网络处理器.

吕高锋(1980－),男,博士,主要研究领域为新型互联网体系结构,高性能路由与交换技术.



孙志刚(1973－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为计算机网络体系结构,高性能路由器.

唐路(1988－),男,博士,主要研究领域为高性能路由器,网络处理器.