

引用格式:李龙飞,王剑峰,刘欢,等. 硬件支持的多虚拟机数据交换及动态带宽分配方法[J]. 微电子学与计算机,2020,37(1):93-98. [LI L F, WANG J F, LIU H, et al. Hardware supported methods of multi-VM switching and dynamic bandwidth allocation[J]. Microelectronics & Computer, 2020, 37(1): 93-98.]

硬件支持的多虚拟机数据交换及动态带宽分配方法

李龙飞,王剑峰,刘欢,史阳春

(西安微电子技术研究所 集成电路设计部,陕西 西安 710065)

摘 要: 针对同一宿主计算机上多虚拟机之间数据交换开销大且带宽分配不灵活的问题,提出了一种硬件支持的多虚拟机数据交换及动态带宽分配方法,并进行建模和实验. 该方法采用 IO 虚拟化的思想,面向以太网控制器的硬件架构进行改进与优化. 通过对虚拟机的发送数据进行解析,同时扩展发送引擎对接收 BD 环的访问权限,实现数据从发送引擎直接向目的虚拟机接收队列的交换过程;通过对虚拟机中接收队列的数据信息进行统计与分析,实现对各个虚拟机的带宽进行动态分配与调整. 以自主研发的千兆以太网控制器为原型搭建测试平台进行实验. 结果表明,本文提出的方法不仅减小了多虚拟机之间数据交换和带宽分配的 CPU 开销,而且对以太网控制器和虚拟机管理程序均保持了兼容.

关键词: NIC 架构;IO 虚拟化;多虚拟机;动态带宽分配

中图分类号: TP393

文献标识码: A

文章编号: 1000—7180(2020)01—0093—06

Hardware supported methods of multi-VM switching and dynamic bandwidth allocation

LI Long-fei, WANG Jian-feng, LIU Huan, SHI Yang-chun

(Department of Integrated Circuit Design, Xi'an Microelectronics Technology Institute, Xi'an 710065, China)

Abstract: Aiming at the problem of high data exchange overhead and inflexible bandwidth allocation between multi-virtual machines (VMs) on the same host, hardware supported methods of multi-VMs switching and dynamic bandwidth allocation are proposed. Based on the concept of IO virtualization, this method focuses on improving and optimizing the hardware architecture of Ethernet controller. By parsing the sending data of VM and extending the access authority of the sending engine to the receiving BD ring, the switching process from the sending engine to the destination VM is realized directly. Besides, through statistics and analysis of data of receiving queue in VM, the bandwidth of each VM can be dynamically allocated and adjusted. The experimental results show that the proposed methods not only reduce the CPU overhead of data exchange and bandwidth allocation between multi-VMs, but also maintains compatibility with Ethernet controllers and hypervisors.

Key words: NIC architecture; IO virtualization; Multi-VM; dynamic bandwidth allocation

1 引言

为了提高底层硬件设备的利用率,增加上层应用的灵活性,虚拟机(VM, Virtual Machine)技术目

前在各类云计算平台中得到了广泛的使用. 通过对计算机的硬件资源进行抽象和模拟,虚拟机可以被看作是一台独立的计算机,且可以多个虚拟机同时运行在同一宿主计算机上^[1]. 然而,与本地的物理设

备相比,虚拟机中虚拟设备的性能可能会受到影响,例如 CPU、存储以及 I/O 设备^[2]. I/O 虚拟化是一种将单个物理网络控制器(Network Interface Controller, NIC)设备抽象为多个虚拟 NIC 设备的技术,其可以使每个虚拟机拥有专用的虚拟 NIC. 近年来, I/O 虚拟化技术得到了快速发展,不断有新的技术提出以缩小虚拟机和宿主计算机之间的性能差距. 然而,在当前千兆以太网或万兆以太网不断普及的环境下,提高高速网络设备的 I/O 虚拟化性能仍然具有很大的挑战^[3-4].

在高速网络环境下同一宿主计算机上多虚拟机之间的快速数据交换是十分重要的,尤其是在数据关联性较强的云计算环境. 除此之外,对于复杂的分布应用来说,虚拟机之间的合理带宽分配同样也应该受到重视. 作为目前主流的基于硬件的 I/O 虚拟化技术,尽管 VMDQ 和 SR-IOV 在很大程度上缓解了 CPU 对虚拟机数据收发处理的负担,但是它们对于虚拟机之间的数据交换和带宽分配未做优化和处理,且仍是通过操作系统中的虚拟机管理程序来实现^[5-6]. 随着网络带宽的提高和 CPU 计算资源的限制,这种基于软件的方式势必会受到越来越大的压力. 事实上,在 NIC 中采用专用硬件的方式来解决这一问题是具有明显优势的. 一方面,硬件与生俱来的高速性会加快虚拟机之间的数据交换;另一方面,所有的操作和处理都不会占用任何额外的 CPU 资源^[7-8]. 就目前国内外所公开的文献来看,虽然有不少研究人员对如何加速多虚拟机之间的数据交换进行了研究,但绝大多数都处于理论研究阶段,并未有实际的原型实验.

针对上述问题,本文面向自主研发的千兆以太网控制器,提出一种硬件支持的多虚拟机数据交换及动态宽带分配方法.

2 相关研究工作

2.1 Vswitch 技术

随着分布式计算、云计算等技术的发展,内部虚拟机之间的数据交换变得越来越频繁. 所谓内部虚拟机,是指在同一宿主计算机上构建的若干个虚拟机. 内部虚拟机之间的数据交换如果采用传统物理交换的方式实现,那么将会违背交换技术中的生成树协议,因为数据的收发过程总是通过同一个交换端口. 为了解决这个问题,IEEE 制定了相关的 802.1Qbg 和 802.1Qbh 协议,即数据中心边缘虚拟交换技术,以使交换机能够有效支持内部虚拟机的数据

传输^[9]. 然而,由于该项技术需要对交换机硬件设备和虚拟机管理软件同时进行升级,实现成本较大,不利于在已部署的网络环境中应用,因此这项技术并未得到广泛的使用.

为了更简便地解决这个问题,研究人员寻求从虚拟机管理软件的层面入手,并提出了一种称之为虚拟交换(Vswitch, Virtual Switching)的技术^[10-11]. Vswitch 技术是在虚拟机管理软件的层面,通过相应的软件算法,判断各个 VM 所发送数据的目的地址,并判断是否需要进行内部转发,从而实现内部虚拟机的数据交换.

2.2 带宽分配策略

理论上讲,同一宿主计算机上的多个虚拟机总是会平均共享物理网络带宽. 然而,由于各类应用和数据需求的不同,虚拟机通常都会被分配一个固定的带宽以满足其业务需要,保证性能稳定. 通常来说,带宽的分配都是由系统管理员决定的,一旦配置完成,各个虚拟机的带宽也就确定.

事实上,在不同的应用场景下,虚拟机的实际带宽总是处于动态的变化之中,分配带宽与实际带宽之间的差距往往会造成资源的浪费或性能的下降,因此,静态的带宽分配策略并不能满足当前复杂应用环境下虚拟机对带宽的动态需求.

3 改进的方法

本文所提出的多虚拟机数据交换及动态带宽分配方法均采用了基于硬件的思想,选取以太网控制器层面作为解决问题的切入点. 因此为了更好地理解本文所提出的方法,这里首先对改进后的以太网控制器硬件架构进行描述与分析.

3.1 以太网控制器系统架构

基于自主研发的千兆以太网控制器架构,同时结合改进方法的需求,提出了一种对虚拟机友好的以太网控制器系统架构(VFN, VM-Friendly Architecture),如图 1 所示.

VNF 在保持对传统以太网控制器架构兼容的基础上,对接收引擎和发送引擎进行了拓展,为多虚拟机数据交换及动态带宽分配方法的实现提供了底层的硬件支撑.

3.2 多虚拟机数据交换方法

基于硬件的多虚拟机数据交换方法在整个交换过程中不需要 CPU 的参与,因此不会给 CPU 带来任何额外的负担. 在 VFN 的架构中,内部虚拟机之间的数据交换直接在发送引擎和接收引擎之间进行. 由于

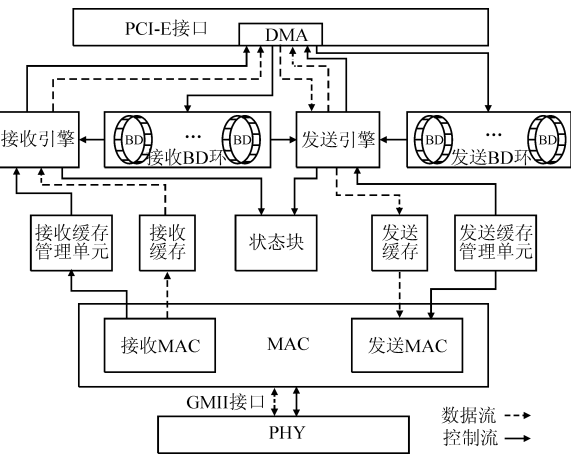


图 1 VFN 总体架构

发送引擎可以访问接收 BD 环,因此当数据帧的目的 MAC 地址等于宿主计算机上其他虚拟机的 MAC 地址时,发送引擎从接收 BD 环中获取 BD,并根据 BD 所描述的内存区间地址直接将数据帧写入对应的内存区间,从而在 NIC 的内部完成了多虚拟机之间数据的交换.为了更好的理解本文所提出的方法,下面结合 VFN 架构来具体描述数据帧是如何在多虚拟机之间完成交换的,其整个处理流程如图 2 所示.

(1)虚拟机 1 准备向虚拟机 2 发送数据帧 A. 通过调用系统驱动,从其对应的 BD 环中选取 BD 对

数据帧 A 进行描述.若采用“分散-聚集”技术,存在多个发送 BD 与数据帧 A 映射.

(2)虚拟机 1 更新发送 BD 环指针信息. NIC 通过更新后的指针信息,判断接下来哪些 BD 需要进行处理.

(3)虚拟机 1 的发送 BD 环通过 DMA 由主机搬运到 NIC. 值得注意的是,接收 BD 环在这个过程中也会进行同步,因此在 NIC 中也保存着虚拟机 2 的接收 BD 环镜像.

(4)发送引擎通过解析发送 BD,获取到数据帧在内存中的地址,并将数据帧由主机搬运到 NIC 中的帧过滤模块.

帧过滤模块会根据数据帧的目的 MAC 地址对其进行过滤,并将其存储在不同的缓存中.如果目的 MAC 地址等于宿主计算机上其他虚拟机的 MAC 地址,那么将数据帧存放在缓存 1 中;否则,存放在缓存 2 中.

(5)发送引擎向接收 BD 环发出请求,从而获取到虚拟机 2 所对应的接收 BD.

(6)解析获取到的接收 BD,并根据 BD 所描述的地址将数据帧通过 DMA 传输到所对应的内存空间.

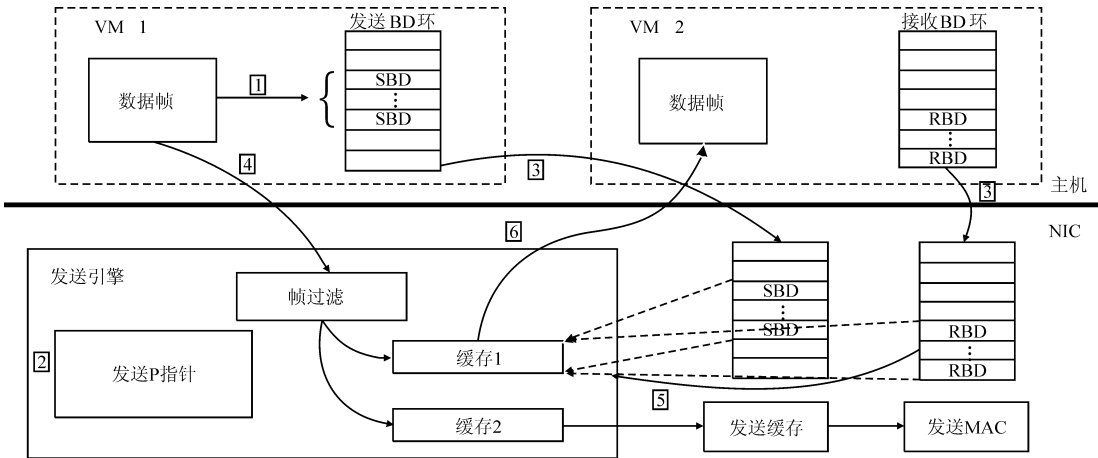


图 2 基于硬件的交换流程示意图

通过上述 6 个步骤,虚拟机 1 与虚拟机 2 实现了数据帧的交换,且整个交换过程中所有的操作都在发送引擎中完成.由于 VFN 架构的支持,数据帧由“发送”状态到“接收”状态的转换过程不涉及 PCI-E 的参与,因此该方法对整个 PCI-E 总线保持了透明,便于在 SR-IOV 技术中进行扩展.

对于非内部交换的数据帧,发送引擎中的帧过滤模块会对其进行识别,并将其存储在另外一块缓存中按正常的发送处理方式进行处理.因此,该方法对传统的数据收发处理流程保持了很好的兼容,对数据收发的处理速率不会带来任何的影响.

3.3 动态带宽分配方法

对于接收过程来说,接收 MAC 在将数据帧存放在接收缓存的同时,都会对每个数据帧进行解析,提取数据帧中关键的信息,例如目的 MAC、VLAN ID 等.在将接收到的数据帧传输到主机内存之前,接收引擎都会根据 MAC 提取的关键信息来决定该数据帧属于哪一个虚拟机,并从对应的接收 BD 环中获取 BD.因此,接收过程可以被视为是由接收到的数据帧进行驱动的.而对于发送过程来说,发送引擎从哪个发送 BD 环中获取 BD 就决定了接下来要发送哪个虚拟机的数据帧.换句话说,获取 BD 的先后顺序就决定了每个虚拟机的带宽.事实上,作为运行着多台虚拟机的宿主计算机来说,虚拟机的发送带宽是衡量是其性能优劣的主要指标.因此,本节主要研究针对虚拟机的发送带宽分配方法.

轮询算法(RR, Round-Robin)是目前在 NIC 中广泛使用的多 BD 环调度算法,其可以保证各个虚拟机的 BD 环依次被发送引擎访问,从而实现各个虚拟机发送带宽的平等^[13].然而,轮询算法虽然保证了虚拟机之间的公平,但统一的、固定的处理方式并不能满足某些情况下虚拟机对带宽的需求,例如瞬时的流量激增、延时敏感型数据的处理等.在这种情况下,本节基于 VFN 架构,提出了一种动态的带宽分配方法.

通过对数据的业务类型以及网络传输层协议进行分析,发现接收数据流的带宽在某种程度上可以反映出每个虚拟机对带宽的需求.这是因为目前几乎所有的网络应用都是基于 TCP/IP 协议实现的,而 TCP 协议又是一种面向连接的传输方式,其采用确认应答的机制来保证传输的可靠性.这样一来,虚拟机在单位时间内接收到的 ACK 报文数量就可以体现出其对发送带宽的相对需求.基于这样的思想,考虑到带权重的轮询算法(WRR, Weighted Round-Robin)在调度时可以体现出个体之间的相对优先级,因此对每个虚拟机的接收带宽进行统计,从而可以将统计的结果作为权重的划分依据.具体到 VFN 架构,在接收引擎中增加相应的硬件单元对每个虚拟机的接收速率进行统计,并将统计的结果转换为发送引擎中每个 BD 环在轮询时的权重值.当轮询过程中所有的调度计数器都等于零时,发送引擎会再次更新每个 BD 环的权重值,同时接收引擎又会开始新一轮的统计.

由于 BD 环与虚拟机是一一对应的,因此具有较高权重的 BD 环在轮询过程中会被多次访问,从而使对应的虚拟机获得相对更大的带宽.此外,在整个接收和发送的过程中,每个虚拟机的发送带宽总

是会根据接收带宽的变化而不断地调整,从而实现了多个虚拟机动态的带宽分配.动态带宽分配方法的伪代码描述如下.

```

For N VMs,  $0 \leq i \leq N$ ;
while(RX engine starts) then
    initial all weight value(i) and RX_counter(i);
loop:
    start statistics timer;
    if(RX engine get a BD from RX BD ring(i))
        if(RX_counter(i) is not full)
            RX_counter(i) = RX_counter(i) + 1;
        if(RX_counter(i) is full or statistics timer is up) begin
            output weight value(i) for each VM based on RX_counter(i);
            RX_counter(i) = 0;
        back to loop;
    end
while(TX engine starts) then
loop:
    config sheduler_counter(i) = weight value(i);
    start round-robin from VM 0 to VM N;
    for( $i=0; i \leq N; i++$ )
        if(scheduler_counter(i) != 0) begin
            use a send BD from send BD ring(i);
            sheduler_counter(i) = sheduler_counter(i) - 1;
        end
    else step to next BD ring;
    if(all scheduler_counter = 0)
        back to loop;

```

4 性能评估

在 Stratix V FPGA(Device: 5SGSMD4E2H29I3)上对 VFN 架构进行了实现,并通过 PCI-E 接口与测试机主板相连,作为该测试机的 NIC.为了对性能进行横向对比,实验中选择了一台与测试机具有相同硬件配置的 PC 机,且该 PC 机的 NIC 采用传统方式实现.具体来说,二者均搭载 Intel I7 处理器,8GB DDR3 内存,运行 Xen-4.2.0 虚拟机管理软件.实验中对每个虚拟机均分配 1GB 内存,一个 CPU 核的计算资源.此外,每个虚拟机均运行 Linux 操作系统.

首先对 VFN 架构下多虚拟机之间数据交换的性能进行评估.实验中选择同一宿主计算机上的两个虚拟机进行数据的传输,数据大小为 100 MB,并分别记录不同数据帧长度下的传输耗时,如图 3 所示.从图中可以明显地看出,无论数据帧长度如何变化,采用 VNF 架构的测试机均花费了较小的时间完成了数据传输.

在采用传统 Xen 方案时,其传输耗时随着数据帧长度的增加而持续地降低,且当帧长达到 1 518 字节时,其传输耗时基本与 VFN 方案持平.这主要是因为当传输数据总量一定时,较大的帧长会降低中断产生的次数,从而会减小数据传输的总体耗时.

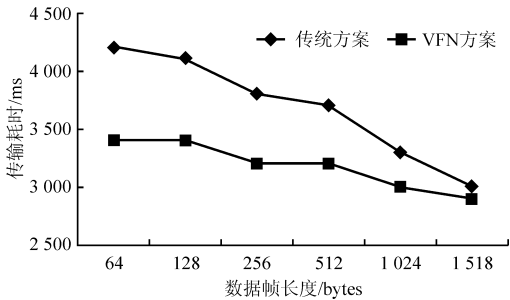


图 3 不同数据帧长度下的传输耗时对比

评估虚拟机之间的交换速率对 CPU 性能的影响.图 4 示出了不同交换速率下 CPU 占用率的统计结果.实验结果表明,在使用了 VFN 架构后,CPU 的占用率最大减小了 12%,且随着交换速率的提升,VFN 相对传统方案所带来的性能优势就越发明显.从图中可以看出,当交换速率从 200 Mbps 提高到 500 Mbps 时,传统方案下的 CPU 占用率从 48% 突增至将近 60%.然而,在采用 VFN 的情况下,CPU 的利用率并不同有随着交换速率的增大而激烈变化,而是保持了相对的平衡.这主要是因为 VFN 架构下的交换过程主要通过硬件的方式完成,从而对 CPU 保持透明.

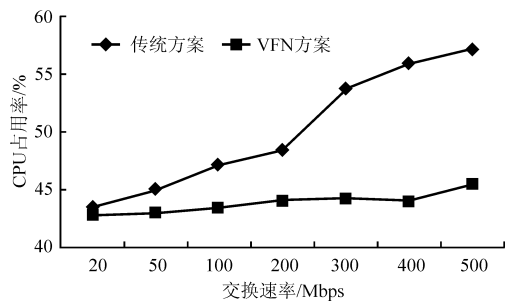


图 4 不同交换速率下 CPU 占用率对比

对动态带宽分配方法的性能进行评测.类似地,两台 PC 机上均分别运行 4 个虚拟机,每个虚拟机具有相同的设置,但被分配不同的接收带宽,以此来模拟虚拟机对网络带宽的不同需求.实验中,每个虚拟机均运行 FTP 服务上传 1 GB 的数据,同时对每个虚拟机发送不同带宽的数据,以实现虚拟机之间实际接收带宽的差异.实验进行中对目标虚拟机的发送带宽进行监测和统计,得到实验结果如图 5 所示.图中横

坐标为目标虚拟机的接收带宽在总带宽中的占比,纵坐标为目标虚拟机的实际发送带宽.从图中可以得到,在目标虚拟机的接收带宽较小时,动态带宽分配方法给目标虚拟机分配了较小的发送带宽,所以其发送带宽小于采用 Xen 方案下的带宽.然而,随着接收带宽的不断增大,尤其是在占比超过了 30% 以后,动态带宽分配方法给目标虚拟机提供了持续增大的带宽,且最高达 600 Mbps,明显高于传统方案下的 240 Mbps.对于 Xen 下的传统方案,虚拟机的最大发送带宽由系统静态分配决定,而不会受到接收带宽变化的影响,因此保持了相对稳定的状态.

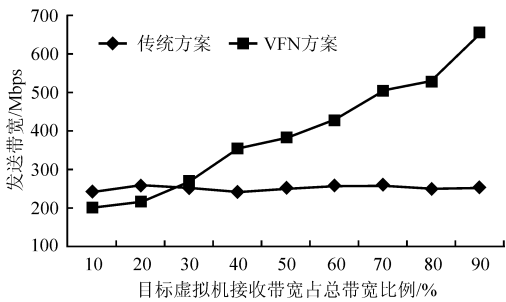


图 5 不同接收带宽下的发送带宽分配结果

最后对虚拟机在相同发送带宽下 CPU 占用率的情况进行实验.实验中保持两种方案下虚拟机的发送带宽处于一致的水平,并对 CPU 的占用率进行统计,得到结果如图 7 所示.在采用动态带宽分配方法时,其 CPU 的占用率相对传统方案较小,且随着发送带宽的增大而优势趋于明显.对数据进一步挖掘可以发现,在采用动态带宽分配方法的情况下,CPU 占用率随着带宽增加而有一定程度的增大,并保持最终的稳定.这主要是因为 VFN 架构下,不断增加的带宽会涉及到越来越多的 BD 管理和维护,因此 CPU 的负载会有所增加.

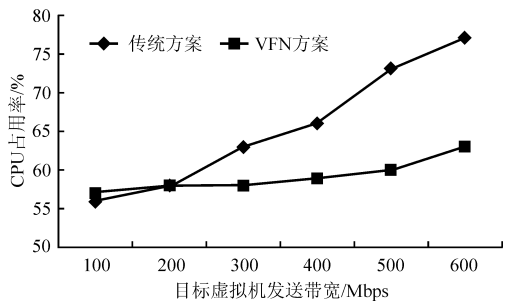


图 6 不同发送带宽下的 CPU 利用率

5 结束语

本文立足于对虚拟机网络性能的提高,针对多虚

拟机数据交换和带宽分配中存在的不足,从硬件层面出发对以太网控制器的系统架构进行研究,并提出了对虚拟机友好的 VFN 架构. 基于此,提出了硬件支持的多虚拟机数据交换方法以及动态的带宽分配方法. 实验结果表明,本文提出的方法不仅有效减小了多虚拟机之间数据交换和带宽分配的 CPU 开销,而且整个处理过程对上层的虚拟机管理程序保持透明.

参考文献:

[1] 金顺福,郝闪闪,王宝帅. 融合双速率和工作休眠的虚拟机调度策略及参数优化[J]. 通信学报, 2017, 38 (12):10-20.

[2] 叶可江,吴朝晖,姜晓红,等. 虚拟化云计算平台的能耗管理[J]. 计算机学报, 2012, 35(6):1262-1285.

[3] ADAMS K, AGESEN O. A comparison of software and hardware techniques for x86 virtualization [J]. ACM SIGOPS Operating Systems Review, 2006, 40 (5): 2-13.

[4] GUAN H, MA R, LI J. Workload-aware credit scheduler for improving network I/O performance in virtualization environment [J]. IEEE transactions on cloud computing, 2014, 2(2): 130-142.

[5] SHEA R, LIU J. Network interface virtualization: challenges and solutions[J]. IEEE Network, 2012, 26 (5): 28-34.

[6] DONG Y, YANG X, LI J, et al. High performance network virtualization with SR-IOV [J]. Journal of Parallel and Distributed Computing, 2012, 72 (11): 1471-1480.

[7] FEI Z, LIU G, FU X, et al. A Survey on Virtual Machine Migration: Challenges, Techniques and Open Issues[J]. IEEE Communications Surveys & Tutorials, 2018(99):1.

[8] GUMASTE A, DAS T, KHANDWALA K, et al. Network Hardware Virtualization for Application Provisioning in Core Networks[J]. IEEE Communications Magazine, 2017, 55(2):152-159.

[9] AZODOLMOLKY S, WIEDER P, YAHYAPOUR R. Cloud computing networking: challenges and opportunities for innovations[J]. IEEE Communications Magazine, 2013, 51(7): 54-62.

[10] 张洪素. 基于 Xen 的虚拟交换技术的研究与设计[D]. 天津:天津大学, 2012.

[11] CHERKASOVA L, GARDNER R. Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor[C]//USENIX Annual Technical Conference, General Track, 2005: 50.

[12] 郭军,闫永明,马安香,等. 云环境下基于冷点虚拟机迁移的热点消除方法[J]. 清华大学学报(自然科学版), 2016(11):1232-1236.

[13] LIAO G, BHUYAN L. Performance measurement of an integrated nic architecture with 10gbe[C]// 17th IEEE Symposium on High Performance Interconnects. New York, NY, USA ,IEEE, 2009: 52-59.

作者简介:

李龙飞, 男,(1988-),博士,工程师. 研究方向为高性能网络芯片、网络硬件加速、网络安全等.
E-mail: longfeisos@163. com.

王剑峰 男,(1971-),研究员. 研究方向为高性能系统总线、计算机网络及控制、交换芯片等.

刘欢 女,(1983-),高级工程师. 研究方向为高性能系统总线及控制芯片等.

史阳春 男,(1983-),高级工程师. 研究方向为高性能网络控制、交换芯片等.

(上接第 92 页)

[14] SHI X Y, LIU Y Q, CHOU H R. 48-Mode Reconfigurable Design of SDF FFT Hardware Architecture Using Radix-32 and Radix-23 Design Approaches [J]. IEEE Transactions On Circuits And Systems-I: Regular Papers, 2017, 64 (4): 1456 - 1467. DOI: 10. 1109/ TCSI. 2017. 2654451.

[15] 杨学鹏. 异构多核 SoC 中大点数 FFT 加速单元的的实现[D]. 安徽:合肥工业大学, 2013.

[16] 禹霁阳,李阳,黄丹,等. 一种高性能单精度浮点基-3 蝶形运算单元的设计与实现[J]. 仪器仪表学报, 2010, 31(12): 2676-2681.

[17] 王靖琰,曹新民. 混合基 FFT 处理器的设计与 Verilog 实现[J]. 中国科技论文在线精品论文, 2008, 1(10):

1128-1135. <http://www. paper. edu. cn/releasepaper/content/200802-267>.

[18] 方波. 基于改进 CORDIC 算法的 FFT 处理器设计与研究[D]. 广州:华南理工大学, 2013.

作者简介:

宋宇鲲 男,(1975-),博士,副研究员. 研究方向为面向数据密集与计算机密集应用的 SoC/MPSoC 体系结构与实现.

曲双双 女,(1993-),硕士研究生. 研究方向为 SoC 设计.

徐礼晗 男,(1994-),硕士研究生. 研究方向为 SoC 设计.

张多利(通讯作者) 男(1975-),博士,研究员. 研究方向为多核处理器体系结构与设计方法、多媒体专用集成电路设计实现. E-mail: zhangduoli@hfut. edu. cn.