计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20131182
52(1):83-93, 2015

基于单根 I/O 虚拟化的多根 I/O 资源池化方法

王 展[1,2] 曹 政[1] 刘小丽[1] 苏 勇[1,2] 李 强[1] 安学军[1] 孙凝晖[1]
[1](中国科学院计算技术研究所高性能计算机研究中心 北京 100190)
[2](中国科学院大学计算机与控制工程学院 北京 100190)
(wangzhan@ncic.ac.cn)

A Multi-Root I/O Resource Pooling Method Based on Single-Root I/O Virtualization

Wang Zhan[1,2], Cao Zheng[1], Liu Xiaoli[1], Su Yong[1,2], Li Qiang[1], An Xuejun[1], and Sun Ninghui[1]
[1](*High Performance Computer Research Center*, *Institute of Computing Technology*, *Chinese Academy of Sciences*, *Beijing* 100190)
[2](*Institute of Computer and Control Engineering*, *University of Chinese Academy of Sciences*, *Beijing* 100190)

Abstract Virtualization offers data center with efficient server consolidation and flexible application deployment, but it requires data center servers improve their I/O devices to get with the needs of virtualization, and to make up the performance degradation brought by device virtual sharing between virtual machines. These changes bring the redundancy of I/O devices for each server under current I/O architecture, increase the cost of data center infrastructure and add more I/O cables between servers. To solve these problems, we design and implement a SRIOV-based multi-root I/O resource pooling method. Through a hardware-based PCIe ID remapping and address remapping technology, virtual functions in the same SR-IOV I/O devices can be shared among different physical servers, which efficiently reduces the redundancy of I/O resources under virtualization environment. We also adopt a hotplug-based virtual I/O device allocation method to dynamically adjust resources between servers for increasing resource utilization. Experiments prove our design does can provides functions mentioned above and maintain server I/O performance as it using directly-attached devices.

Key words data center; server; I/O virtualization; I/O resource pooling; PCIe compatibility

摘 要 虚拟化技术在为现代数据中心提供高效的服务器整合能力和灵活的应用部署能力的同时,也对数据中心服务器的 I/O 系统设计提出了新的需求,现有 I/O 资源与服务器紧密绑定的 I/O 体系架构将产生成本上升、资源冗余、I/O 连线复杂化等一系列问题.针对上述问题,提出了一种基于单根 I/O 虚拟化协议(single root I/O virtualization, SR-IOV)的多根 I/O 资源池化方法:基于硬件的多根域间地址和 ID 映射机制,实现了多个物理服务器对同一 I/O 设备的共享复用,有效减少单体服务器所需的设备数量和连线数量,并进一步提高服务器密度;同时提出虚拟 I/O 设备热插拔技术和多根共享管理机制,实现了虚拟 I/O 资源在服务器间的实时动态分配,提高资源的利用效率.提出的方法在可编程逻辑器件(fieid-programmable gate array, FPGA)原型系统中进行了验证,其评测表明,方法能够在实现多根 I/O 虚拟化共享的同时,保证各个根节点服务器获得近乎本地直连设备的 I/O 性能.

关键词 数据中心;服务器;I/O 虚拟化;I/O 资源池;PCIe 兼容

中图法分类号 TP334

收稿日期:2013-08-17;**修回日期**:2014-05-09
基金项目:国家自然科学基金青年科学基金项目(61100014)
通信作者:曹 政(cz@ncic.ac.cn)

虚拟化技术在现代数据中心基础设施构建中发挥着重要作用.借助于单计算机系统虚拟化,数据中心管理者可以将原来的多个物理服务器整合为运行于单个物理服务器之上的多个虚拟机,从而提高硬件资源利用率,降低基础设施成本、能耗以及空间占用[1].同时,数据中心利用虚拟资源动态分配和虚拟机在线迁移等虚拟化相关特性,还能实现服务器间负载均衡,加速应用部署,提高整个基础设施的可用性[2].

然而,虚拟化技术的深入使用对数据中心服务器的 I/O 系统设计提出新需求:

1) 性能.虚拟机作为物理服务器的替代,当其通过软件模拟的方式[3]使用宿主服务器的物理 I/O 设备时,软件处理带来的 I/O 性能损失[4-5]需要通过物理 I/O 设备数量或性能的提升来弥补[6].

2) 扩展性.直接 I/O 虚拟化技术(pass-through mode)[7-8]允许虚拟机通过独占宿主机物理设备来提高 I/O 访问效率.但该技术需要 I/O 设备的数量与虚拟机个数匹配,即对服务器的 I/O 系统扩展性提出需求[9].

3) 设备多样性.虚拟化技术允许多种应用整合部署在单个物理服务器之上,这就要求宿主物理服务器配备更多类型的 I/O 设备,以支持多种应用所对应的不同的存储、计算以及通信协议[9].

现有数据中心服务器采用 I/O 资源与物理主机绑定的紧耦合架构,只能以增加单体主机内物理设备的性能、数量以及种类来应对上述需求,这无疑会增加单体服务器的 I/O 资源冗余,提高服务器的购置成本和管理成本,同时单机 I/O 数量和种类的增加还会导致整个数据中心网络连线数量的增加和网络多样化,增加网络部署的复杂度.

目前工业界和学术界针对新型数据中心服务器 I/O 体系结构的研究主要从以下 3 个方面展开:

1) 聚合.通过设备本身的硬件辅助虚拟化技术,将直接 I/O 虚拟化所需的多个物理 I/O 设备聚合转换为单个物理设备内的多个虚拟功能,以减少单个物理服务器所需配备的 I/O 设备数量,例如 PCI-SIG 提出的单根虚拟化技术(single root I/O virtualization, SR-IOV)[10].

2) 融合.通过设备本身的硬件辅助协议转换,实现单个物理 I/O 设备支持多种协议数据包的处理和传输,从而降低数据中心所需配备的 I/O 设备和互连网络的多样性,例如以太网光纤通道技术(fiber channel over Ethernet, FCoE)[11].

3) 共享.将 I/O 设备的共享范围从单个物理服务器内的多个虚拟机之间扩大到多个物理服务器之间,以缩减单机所需的 I/O 设备的数量和数据中心内部网络连线的数量,同时通过共享复用提高 I/O 资源的利用率,例如 NEC ExpEther[12]和 NextIO vNet[13]等,此部分工作将在后面详细介绍.

更为全面的解决方案应该兼顾上述 3 个方面,并进行有效的整合和拓展.本文正是基于这种思想,设计并实现了一种基于单根 I/O 虚拟化 SR-IOV 协议的多根 I/O 虚拟资源池化方法,在实现物理服务器多根之间 I/O 虚拟动态共享的同时,兼顾 PCIe SR-IOV 技术的 I/O 聚合特性.

如图 1 所示,本文方法提出了一种 PCIe 多根互连交换架构,以此为基础,通过硬件层次的多根域间 ID 映射、地址映射以及虚拟设备功能模拟,构建出由虚拟设备实例为资源分配实例的多根 I/O 共享资源池,缩减多根环境下所需 I/O 设备的数量;同时本文提出并实现了以虚拟 I/O 设备热插拔为基础的多根 I/O 资源动态管理机制,使作为 PCIe 根节点的物理服务器(包括其上运行的虚拟机)可以以按需的方式,从资源池中动态地获取和释放虚拟 I/O 设备,提高系统中 I/O 设备的使用效率.值得强调的是,本文提出的方法无需对 I/O 设备及其驱动进行任何修改,对上层软件完全透明.

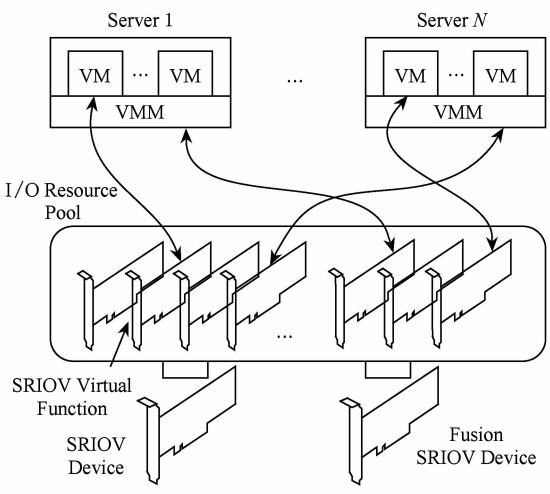


Fig. 1 Our multi-root I/O resource pooling architecture.

图 1 本文多根 I/O 资源池化技术架构

1 相关工作

学术界和工业界都已经开展了一些研究工作,致力于实现多服务器间的 I/O 共享.NEC 的 ExpEther[12]使用以太网将传统 PCIe I/O 设备的共享范

围从单个服务器内部扩展到多个服务器之间,以太网的两端使用“PCIe←→以太网”协议转换桥分别连接服务器和 I/O 设备,该协议转换桥负责实现多个服务器对同一设备的共享使用. 文献[14]在 ExpEther 的基础之上,支持以 SR-IOV 设备的虚拟功能作为多服务器间共享环境下的资源分配实例,兼顾了 SR-IOV 技术的 I/O 聚合特性. 文献[12,14]事实上是设备网络化技术(device over Ethernet),由于服务器与 I/O 设备之间的数据交换需要经过 PCIe 和以太网之间的协议转换,增加了系统 I/O 操作的开销和延时. 文献[15]使用软件方式实现 I/O 设备的多服务器间共享,其使用一台配有 I/O 设备的物理主机作为多机 I/O 共享的服务器端,通过网络向客户端提供通用的 I/O 接口服务,客户端使用该接口进行 I/O 操作. 然而,这种集中式的 I/O 共享服务提供模式使服务器端主机成为整个系统的性能瓶颈,此外,由于经过了较多的软件层次导致客户端主机的 I/O 性能较低.

在工业界,PCI-SIG 提出了 MR-IOV(多根虚拟化)技术[16],通过对标准 PCIe 协议修改和扩展,使得支持 MR-IOV 协议的处理器和设备可原生支持 I/O 的多服务器间共享. 但该技术要求计算机 I/O 系统的各个部分(包括芯片组和设备)都按照 MR-IOV 协议进行修改实现,较大的系统改动导致其未得到普遍应用. NextIO 公司关注于数据中心单机架内所有服务器间的 I/O 设备共享,其设计的 vNET I/O[13]将单个机架所需的以太网卡和 HBA 卡整合部署在一个 I/O 箱(I/O Box)内,该 I/O 箱可将单个物理设备虚拟为多个虚拟设备,供多个服务器共享使用. 服务器只需使用普通的 PCIe 扩展适配卡连接到该箱,无需修改上层操作系统和设备驱动. vNet I/O 可以简化数据中心 IO 设备部署和内部以太网及光纤网络的连线数量,提高系统的 I/O 管理效率. 但是,vNet IO 没有利用 SR-IOV 设备本身即可提供多个虚拟设备的特性,技术设计存在的冗余. 其他如 Xsigo[17], Virtensys[18] 公司也有类似的产品,但要么存在类似文献[12,14]的协议转换开销问题,要么限定可共享 I/O 设备种类,无法实现设备多样性需求.

本文提出的基于 SR-IOV 的多根间 I/O 资源池化方法克服了上述问题,利用 SR-IOV 设备自身的虚拟化特性优化了整个 I/O 资源池化共享框架,而且不引入任何协议转换开销,不限制 I/O 设备的功能类型.

2　相关背景和关键问题

2.1　相关背景

PCIe SR-IOV(单根 I/O 虚拟化)协议[10]是标准 PCIe 总线互连协议的扩展,是本文的设计基础. PCIe SR-IOV 主要目标是通过 I/O 设备自身的硬件虚拟化,将单个物理设备呈现为一个物理功能(physical function)和若干虚拟功能(virtual function). 物理功能用于设备基本功能的配置和实现,而虚拟功能由上层系统配置相应物理功能的 PCIe 配置空间所产生,在 I/O 设备内拥有与上层接口交互数据所需的独立资源(如操作队列),因此可以作为独立 I/O 设备使用. SR-IOV 协议服务于支持直接 I/O 虚拟化的单计算机系统,系统上运行的每个虚拟机都可以直接拥有独立的 I/O 设备(物理功能或虚拟功能).

可见,SR-IOV 技术可以降低系统所需要的 I/O 设备数量,但它仅支持将单个设备的多个 SR-IOV 虚拟功能分配给单计算机系统中的不同虚拟机使用,无法实现资源在大规模系统范围内的聚合和有效共享. 将若干支持 SR-IOV 的设备聚合,构建 I/O 设备资源池,实现资源池对多个物理服务器的按需 I/O 服务,这是本文的研究目标.

2.2　关键问题

现有数据中心服务器都以单机资源绑定的模式使用物理 I/O 设备,其整个架构从底层硬件到上层操作系统和设备驱动都是为分立隔离的单机系统而设计,即 I/O 资源仅能在单个物理计算机之内共享使用,在此基础之上实现多物理服务器间 I/O 资源池化共享,现面临如下 3 个问题:

1) 多根 I/O 互连. I/O 总线是信息在处理器与设备之间流动的数据通路,目前主流服务器的 I/O 系统采用单根互连结构,即以处理器为根、设备为叶子,设备只与其唯一的根处理器有连接通路,从而在物理结构上限定了设备的使用范围. 为了实现多服务器间的 I/O 共享,首先就要破除这种单根互连结构,提供单个 I/O 设备到多个根处理器的互连通路,构建多根 I/O 互连拓扑结构. 同时为了兼容现有处理器和 I/O 设备,多根互连结构需要兼容现有单根协议数据包的传输.

2) SR-IOV 设备的多根共享. PCIe SR-IOV 协议作为标准 PCIe 协议的扩展,依然是单根协议,SR-IOV 设备及其虚拟功能都只能在单根环境中被

发现、配置和使用. 在本文所构建的多根环境下,多个服务器都作为根节点对被共享的 SR-IOV 设备资源发起配置访问和 I/O 操作,会相互覆盖引发混乱,导致 I/O 设备不能正常工作,甚至可能导致服务器崩溃. 因此,需要在服务器与设备之间引入 SRIOV 设备的多根共享管理,建立 SR-IOV 设备的多个虚拟功能与不同的服务器根节点间的配置映射关系,在不修改根节点操作系统和设备驱动的前提下,实现 SR-IOV 设备在多个根节点间互不干扰的共享使用.

3) 共享资源分配. 资源按需分配技术是共享环境下实现资源合理高效使用的必备手段. 在本文所讨论的 I/O 共享环境下,可以使用以下两种方法实现 I/O 资源的分配:①通过集中式的设备共享管理,对各个服务器 I/O 操作被发往共享设备的优先级进行调度,属于细粒度的资源使用划分;②通过静态映射或动态插拔的方式,控制共享服务器可支配的 SR-IOV 设备虚拟功能数量,属于粗粒度的资源调配. 为了适应于不同的应用需求场景,两种分配方法都应在设计实现中予以支持.

本文工作解决了多根 I/O 互连和 SR-IOV 设备多根共享问题,并实现了以方法②为主的共享资源分配.

3 系统设计

本文基于 SRIOV 的多服务器间 I/O 共享系统设计如图 2 所示,对应 2.2 节所述的 3 个关键问题,该系统主要由 3 部分组成:PCIe 多根 I/O 互连交换机、SR-IOV I/O 共享管理引擎以及热插拔控制器. 下面对这 3 部分分别进行具体阐述.

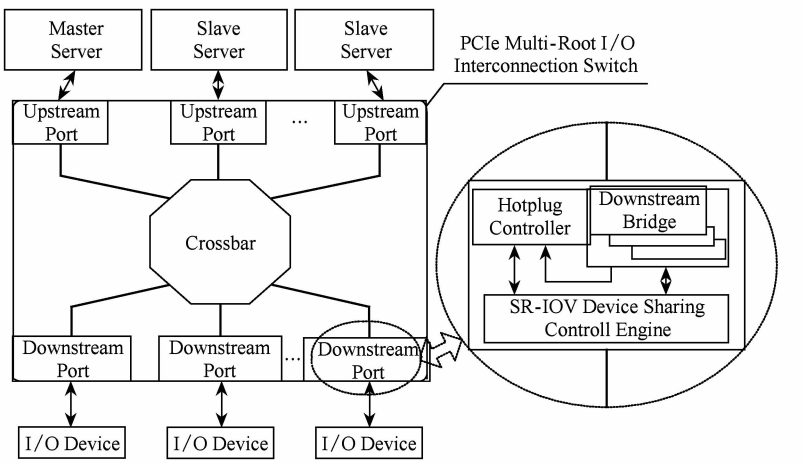


Fig. 2 System design of our architecture.

图 2 本文多根 I/O 资源池化系统结构

3.1 PCIe 多根 I/O 互连交换机

PCIe 多根互连交换机用于连接多个共享物理服务器与 PCIe SR-IOV 设备,该交换机兼容标准 PCIe 协议,能够在不增加任何协议转换开销的基础上,实现以标准单根 PCIe 数据包为交互载体的、多个物理服务器与多个 I/O 设备之间的互连通信.

该多根交换机由上游端口、下游端口以及交叉开关 3 部分组成. 每个上游端口用于连接一个物理服务器,其内部实现为一个 PCIe 上游(upstream)桥(见 PCIe 规范[19]);每个下游端口用于连接一个 PCIe SRIOV 设备,如图 2 所示其内部实现为多个 PCIe 下游(downstream)桥(见 PCIe 规范[19]). 每个 Downstream 桥通过交叉开关与一个 Upstream 桥连接,形成一个标准单根 PCIe 交换机呈现给 Upstream 桥上游端口所连接的物理服务器. 这样,从该服务器的角度来看,多根交换机及其连接的 I/O 设备就可作为其单根 I/O 系统的自然拓展,服务器本身不需要作任何软硬件的修改就可以加入该多根互连. 同时,该多根交换机通过多个 Downstream 桥的实现,将不同的服务器对应于不同的单根 PCIe 交换机,为不同服务器 PCIe 域内的通信提供了隔离.

在标准单根 PCIe 数据包兼容方面,本文在进入该多根交换机的 PCIe 数据包事务层包头中加入多根路由控制位,以区分不同的物理服务器对同一 I/O 设备的访问,并控制各物理服务器能仅能访问到为其分配的 I/O 资源. 这些控制位会在数据包离开多根交换机时被清除或还原,不会对原数据包信息造成破坏,因此不影响现有处理器和 I/O 设备对数据包的正确识别和处理,从而在不增加任何协议转换开销的基础之上,实现了标准单根 PCIe 数据

包兼容传输.

3.2　SR-IOV 设备多根共享管理引擎

为了使单个 SR-IOV 设备能被多个根节点共享使用,且相互之间互不干扰,本文为每个 SR-IOV 设备都配备了一个完全硬件化实现的多根共享管理引擎,每个引擎位于 PCIe 多根互连交换机与物理 I/O 设备连接的下游端口中.而考虑到上层操作系统分别通过 PCIe 配置空间和 I/O 功能空间对设备实现的协议处理部分和 I/O 功能部分进行管理和控制,本文也将该引擎划分为 PCIe 配置空间共享管理和 I/O 功能空间共享管理两部分来实现.

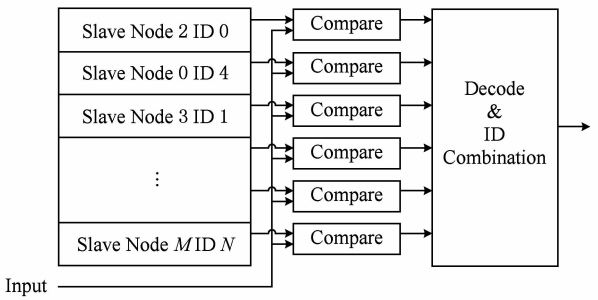
3.2.1　PCIe 配置空间共享管理

在本文所述的多根共享环境下,SR-IOV 设备的多个虚拟功能分别被分配给不同的物理服务器所使用,每个物理服务器在系统初始化时都为其所拥有的虚拟功能分配由总线号、设备号和功能号所构成的系统唯一标识,下面简称为 ID 号,系统后续使用该 ID 号对虚拟功能的 PCIe 配置空间发起读写访问操作.

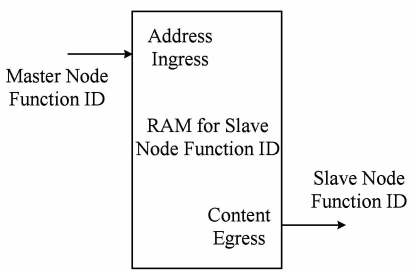
然而如 2.2 节的关键问题 2)所述,SR-IOV 设备只能在单根系统中使用,SR-IOV 设备的各个虚拟功能也只能识别由其物理设备所属单根系统所分配的 ID 号,并响应由该 ID 号所标识的 PCIe 配置空间读写数据包.为此,我们将服务器进行分类,将 SR-IOV 设备真实所属的单根服务器系统称为主控制根节点,将通过共享分配使用设备虚拟功能的单根服务器系统称之为从属根节点.

相应地,PCIe 配置空间共享管理模块的主要功能包括:1)拦截从属根节点访问虚拟功能的配置空间的读写数据包,将其所携带的 ID 号转换为主控制根节点为该虚拟功能分配的 ID 号,再将修改后的数据包发往真实设备;2)拦截设备响应包,将其所携带的 ID 号转换为从属根节点为该虚拟功能分配的 ID 号,再将修改后的响应包返回给从属根节点.

本文使用如图 3(a)所示的硬件基于内容的查找表(content addressable memory, CAM)结构来实现 1)的从属根节点 ID 号到主控制根节点 ID 号的转换,其中,从属根节点 ID 号为 CAM 的输入,译码逻辑和 ID 合成逻辑根据 CAM 的输出合成虚拟功能在主控制根节点域的 ID 号;本文使用如图 3(b)所示的硬件随机存取存储器(random access memory, RAM)结构实现 2)的主控制根节点 ID 到从属根节点 ID 的转换,主控制根节点 ID 号作为 RAM 的读取地址输入,从属根节点 ID 号为 RAM 的读取内容输出.上述两种结构都只需要单个时钟周期就可以完成相应操作.同时,共享管理引擎是分布式实现且与设备绑定的,上述两种查找表的大小都只与设备本身可提供的 SR-IOV 虚功能数量相关,与共享系统内根节点服务器的数目无关,例如若设备可支持 SR-IOV 虚功能的最大个数为 128,则两种查找表分别就有 128 个表项,每个表项的大小为 64 b,即使设备支持的虚功能最大个数为1 024,每个表的大小也只有 8 KB,且表的大小不会因共享该设备的根节点数目的增加而改变,具备良好的可扩展性.



(a)　CAM structure for downstream SR-IOV virtual function ID remapping



(b)　RAM structure for upstream SR-IOV virtual function ID remapping

Fig. 3　ID remapping between master and slave.

图 3　主从根节点 ID 号转换架构示意图

3.2.2　I/O 功能空间共享管理

主控制根节点和从属根节点都通过配置虚拟功能 PCIe 配置空间的基地址寄存器和中断地址寄存器,将该虚拟功能所对应的 I/O 功能空间映射到各自的处理器内存地址空间.主控制和从属根节点与虚拟功能 I/O 功能空间之间的交互操作就通过对相应地址的读写来完成.

与 ID 号相同,真实设备的 SR-IOV 虚拟功能只能识别并响应携带主控制根节点处理器地址的读写数据包,其向上层系统发出的中断数据包携带的也是该地址,因此 I/O 功能空间共享管理模块的主要工作包括:1)记录两种根节点为同一虚拟功能所分配的处理器内存地址空间地址,建立两种地址相互

之间的映射关系;2)拦截从属根节点的 I/O 功能空间读写包和设备虚拟功能的中断包,对其进行地址空间转换,再将修改后的数据包发往设备或从属根节点.因为上述两种操作的响应包使用 ID 号进行路由和匹配,所以 I/O 功能空间共享管理模块需要借用 PCIe 配置空间共享管理模块中的 ID 号转换.

本文使用如图 4 所示的结构实现从属根节点内存地址空间到主控制根节点内存地址空间的转换,图 4 中 6 个 CAM 的输入是从属根节点内存地址空间地址,输出是基地址寄存器的命中信息和 SR-IOV 虚拟功能的命中信息,然后根据这两种信息合成相应的地址,读取存储主控制根节点域地址信息的 RAM,最终根据读取的结果以及原始的从属根节点地址合成 SR-IOV 虚拟功能可识别的地址.上述地址映射查找结构的大小也只与设备本身可提供的 SR-IOV 虚功能数量相关,具备良好的可扩展性.

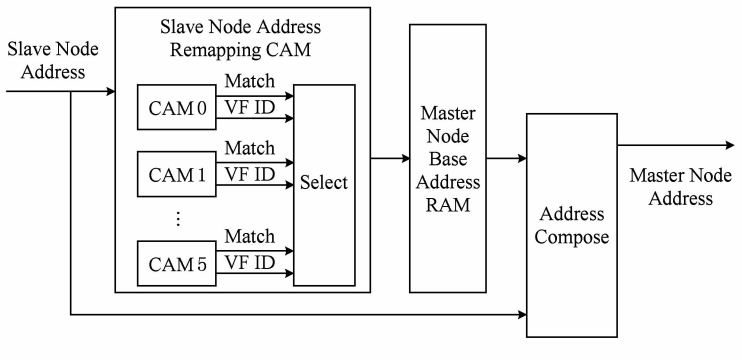


Fig. 4 Structure of address translating between master and slave.

图 4 从属节点地址到主控节点地址转换结构

3.3 多服务器间的 I/O 共享资源分配

目前设计实现了 2.2 节中关键问题 3)所阐述的第 2 种资源分配方式,即使用静态配置,同时配合设备热插拔技术来控制共享服务器与 SR-IOV 虚拟功能的从属关系.

首先,静态配置由系统管理者通过主控制根节点对多根交换机和 I/O 共享管理引擎进行配置来实现,配置的对象包括多根交换机交叉开关的端口交换表(见 3.1 节)以及 I/O 共享管理引擎的 ID 号转换表(见 3.2.1 节),前者用于粗粒度地控制共享服务器对整个 SR-IOV 设备的访问权限;后者用于细粒度地改变共享服务器对设备内虚拟功能的使用关系.

其次,设备热插拔技术实现动态配置.本文为每个下游端口配备一个热插拔控制器来控制 SR-IOV 设备的动态归属.系统管理者在不同的从属根节点服务器内通过驱动程序向热插拔控制器写入针对相应 SR-IOV 设备热插拔指令,引发控制器向该从属根节点发起热插拔中断,根节点操作系统会相应地进入处理设备热插拔的系统内核程序,该程序对连接设备与服务器的 PCIe Downstream 桥和设备本身执行一系列配置操作,并为设备分配或清除与根节点相关的地址资源.热插拔内核程序运行期间,热插拔控制器以中断形式通知上层系统相关操作的完成,而整个程序运行的结束则标志了设备热插拔操作的完成.

为实现按需的 I/O 资源共享,基于该资源分配系统的资源调度策略研究是未来的主要工作.

4 原型实现

本文使用 Xilinx Virtex6 ff365t FPGA 实现了原型系统,如图 5 所示,其向外部呈现 3 个 PCIe 接口,其中 PCIe Gen2×8 金手指接口和 PCIe Gen2×8 线缆接口分别用于连接为主控制根节点和从属根节点;PCIe Gen2×8 的插槽接口用于连接 SR-IOV 设备;主控制根节点和从属根节点通过该原型系统共享该 SR-IOV 设备的多个虚拟功能.FPGA 内部实现了第 3 节所述的多根交换机、SR-IOV 多根共享管理引擎以和热插拔控制器.

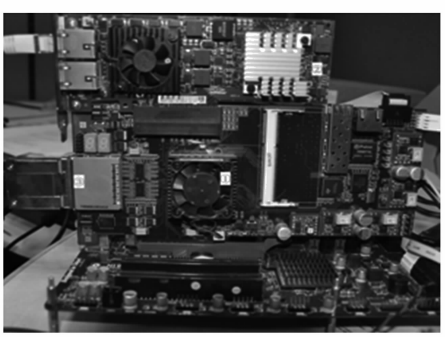


Fig. 5 Prototype of our system.

图 5 本文多根 I/O 资源池化原型系统

5　性能评测

测试平台搭建如下:两台物理服务器(机器配置:CPU 为 Intel i5-3470@3.2 GHz;内存为 8 GB;操作系统:内核版本为 2.6.32 的 CentOS 6.0)通过原型系统共享同一块 Intel 82599 万兆以太网卡,该网卡支持 PCIe SR-IOV 协议.服务器 1 在共享该万兆以太网卡 SR-IOV 虚拟功能的同时,还将充当主控制根节点,服务器 2 则充当从属根节点.下面基于该测试平台,分别从功能完整性和 I/O 共享性能两方面对原型系统进行评测.

5.1　功能完整性测试

根据设计定义,该测试平台应该正确完成如下 3 方面功能以体现原型系统功能的完整性:1)从属根节点正确发现并配置为其分配的 SRIOV 虚拟功能;2)从属根节点使用虚拟功能正确完成相关 I/O 功能操作;3)虚拟功能可以动态地在从属根节点系统中插入和移除.

针对功能 1),首先在整个共享系统初始化阶段,通过写 I/O 虚拟共享引擎的设备号转换表将 Intel 82599 网卡的 4 个虚拟功能分配给服务器 2,在服务器 2 内安装 82599 虚拟网卡驱动,系统正确启动后,使用 lspci 命令查看系统 PCIe 设备配置如图 6 所示.该结果表明从属根节点主机可以在不修改操作系统和设备驱动情况下,正确识别 4 个 Intel 82599 虚拟网卡,并对其配置和加载驱动.

```
[****@sriov-platform2 ~]$ lspci -vt
-[0000:00]-+-00.0  Intel Corporation Device 0150
           +-01.0-[01-03]--+-00.0  Xilinx Corporation Device 8086
           |               \-00.1-[02-03]----00.0-[03]-
           |  +-00.0  Intel Corporation 82599 Ethernet
           |  |       Controller Virtual Function
           |  |
           |  +-00.1  Intel Corporation 82599 Ethernet
           |  |       Controller Virtual Function
           |  |
           |  +-00.2  Intel Corporation 82599 Ethernet
           |  |       Controller Virtual Function
           |  |
           |  \-00.3  Intel Corporation 82599 Ethernet
           |          Controller Virtual Function
```

Fig. 6　PCIe information tree of slave node.

图 6　从属根节点 PCIe 树状架构系统信息

针对功能 2),选择服务器 2 操作系统所识别的任一虚拟网卡接入内网,ping 内网任意节点使用 ssh 进行远程连接,使用 scp 远程拷贝数据如图 7 所示.该结果表明从属根节点可以正确使用分配到的 Intel 82599 虚拟网卡进行网络通信.

针对功能 3),在服务器 2 内使用热插拔控制器驱动,向原型系统中的热插拔控制器发起对服务器 2 使用的虚拟功能的热插入或热拔出操作,通过 dmesg 命令查看系统打印信息如图 8 所示,并且在虚拟功能热插入之后,重新测试其联网通信功能,实验表明该原型系统的设备热插拔功能完全正确.

```
[****@sriov-platform2 ~]$ ping 169.254.9.153
PING 169.254.9.153 (169.254.9.153) 56(84) bytes of data.
64 bytes from 169.254.9.153: icmp_seq=1 ttl=64 time=0.606 ms
64 bytes from 169.254.9.153: icmp_seq=2 ttl=64 time=0.191 ms
64 bytes from 169.254.9.153: icmp_seq=3 ttl=64 time=0.196 ms
64 bytes from 169.254.9.153: icmp_seq=4 ttl=64 time=0.188 ms
64 bytes from 169.254.9.153: icmp_seq=5 ttl=64 time=0.199 ms
64 bytes from 169.254.9.153: icmp_seq=6 ttl=64 time=0.243 ms
--- 169.254.9.153 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5331ms
rtt min/avg/max/mdev = 0.188/0.270/0.606/0.152 ms

[****@sriov-platform2 ~]$ ssh ****@169.254.9.153
****@169.254.9.153's password:
Last login: Thu Aug ** **:**:** 2013 from 169.254.6.193
[****@virtstation ~]$ scp iperf.iso ****@169.254.6.193:/home/
****/
The authenticity of host '169.254.6.193 (169.254.6.193)' can't
be established.
RSA key fingerprint is
43:42:e8:aa:68:cf:bb:20:dd:c4:d6:7e:1b:30:99:23.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '169.254.6.193' (RSA) to the list of
known hosts.
****@169.254.6.193's password:
iperf.iso    100%  430KB 430.0KB/s   00:00
```

Fig. 7　Slave nodes use SR-IOV virtual functions.

图 7　从属节点使用 SR-IOV 虚拟功能通信

```
[root@sriov-platform2 driver]# ./hotplug_out
Hotplug_out
Finished
[root@sriov-platform2 driver]# dmesg -c
pciehp 0000:01:00.0:pcie24: Button pressed on Slot(1)
pciehp 0000:01:00.0:pcie24: PCI slot #1 - powering off due to
button press.

[root@sriov-platform2 driver]# ./hotplug_in
Hotplug_in
Finished
[root@sriov-platform2 driver]# dmesg -c
pciehp 0000:01:00.0:pcie24: Button pressed on Slot(1)
pciehp 0000:01:00.0:pcie24: PCI slot #1 - powering on due to
button press.
```

Fig. 8　Hot-plug of SR-IOV virtual function in server 2.

图 8　在服务器 2 内对 SR-IOV 虚拟功能热拔和热插

5.2　I/O 共享性能测试

本节使用网络性能测试工具 iperf[20] 来测试 2 台服务器共享使用 82599 网卡虚拟功能时的 TCP 带宽性能.iperf 测试是双向的,即要求测试环境中有一端作为 iperf server 监听到达的测试请求,另一端作为 iperf client 发起测试会话,因此,我们在测试平台中引入了服务器 3(机器配置:CPU 为 Intel i5-3470@3.2 GHz;内存为 8 GB;操作系统:内核版本为 2.6.32 的 CentOS 6.0),其配备的 Intel 82599 万兆以太网卡与被共享的网卡通过网线直接连接,网卡最大传输单元 MTU 为 1500,如图 9 所示.下面在该直连网络的基础上构建 3 种 I/O 资源共享分配场景,在不同场景下测试相应物理服务器的 TCP 带宽性能.

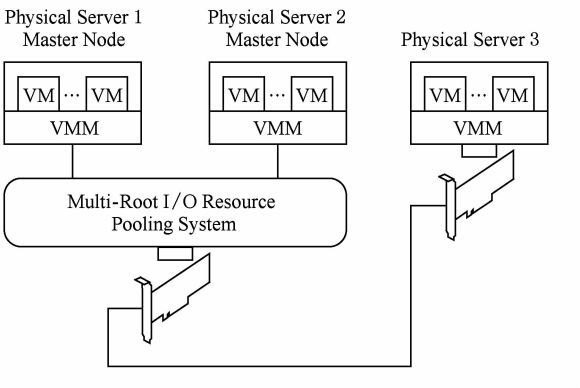


Fig. 9 Schematic plot of iperf testing bed.

图 9 iperf 测试平台示意图

1) 单个从属服务器使用共享网卡的场景:服务器 3 作为 iperf 服务器/客户端,为服务器 2 分配单个 SRIOV 虚拟功能作为 iperf client/server,测试数据如图 9 所示,与其进行对比的是独立使用网卡物理功能的服务器 1 作为 iperf client/server 测试得到的数据.

图 10 中直接使用 82599 网卡的物理服务器作为 iperf client 时的 TCP 带宽为 9.4 Gbps,作为 iperf server 时的 TCP 带宽为 9.26 Gbps;而通过原型系统共享使用网卡 SR-IOV 虚拟功能的物理服务器作为 iperf client 时的 TCP 带宽为 9.39 Gbps,作为 iperf server 时的 TCP 带宽为 9.15 Gbps. 几乎相同的带宽性能表明,本文设计虽然在主机 I/O 路径上增加了额外的处理操作,但是所带来的 I/O 性能损失极少,能够为服务器的提供近乎独占设备的 I/O 性能.

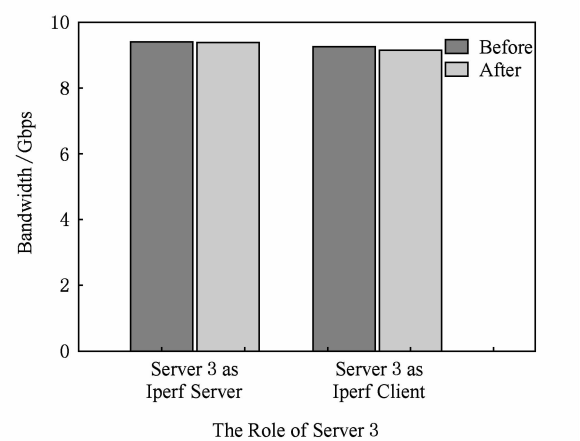


Fig. 10 TCP bandwidth comparison.

图 10 使用虚拟网卡与物理网卡的主机间 TCP 带宽对比

2) 主从服务器同时共享使用网卡场景:服务器 3 作为 iperf server/client,为服务器 1 和服务器 2 分别分配一个 SRIOV 虚拟功能作为 iperf client/server,得到服务器 1 与服务器 2 之间的带宽性能以及带宽分配情况,如图 11 和图 12 所示:

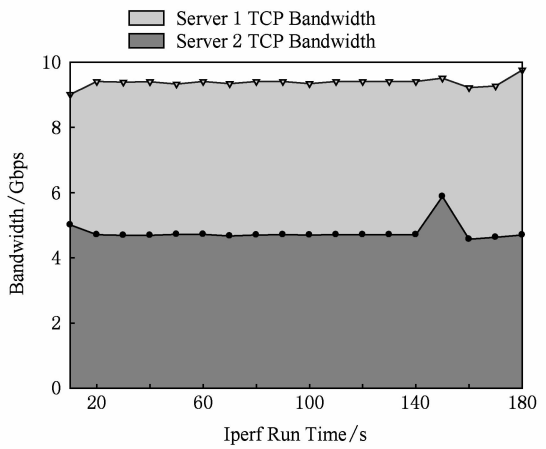


Fig. 11 TCP bandwidth allocation.

图 11 Server 共享虚功能作为 iperf client 时的 TCP 带宽

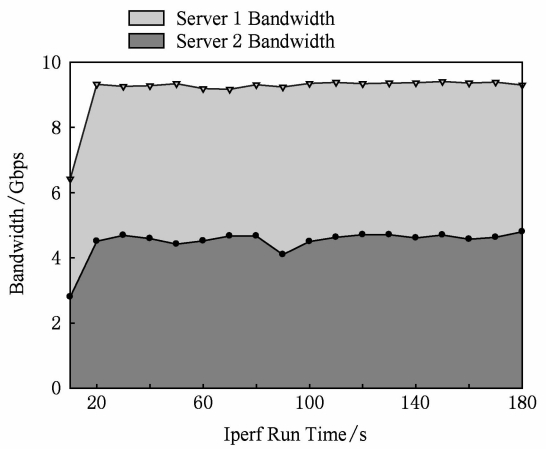


Fig. 12 TCP bandwidth allocation.

图 12 Server 共享虚功能作为 iperf server 时的 TCP 带宽

服务器 1 与服务器 2 作为 iperf client 时的平均 TCP 带宽分别为 4.78 Gbp 和 4.61 Gbps,作为 iperf server 时的平均 TCP 带宽分别为 4.66 Gbps 和 4.49 Gbps,它们的带宽总和基本与独立物理网卡的带宽峰值相当. 该结果表明本文的设计与实现在保证共享设备 I/O 峰值性能的同时,还能够为获得相同共享资源量的物理服务器提供基本公平的 I/O 性能分配.

3) 主从服务器内搭载虚拟机的场景:服务器 3 作为 iperf server/client,服务器 2 内的虚拟机以直接 I/O 分配的方式使用宿主机分配得到的 SR-IOV 虚拟功能,作为 iperf client/server. 在保证每个虚拟机与 SR-IOV 虚拟功能一一对应的情况下,增加虚

拟机数量,测试得到各个虚拟机间的带宽分配情况如图 13 和图 14 所示:

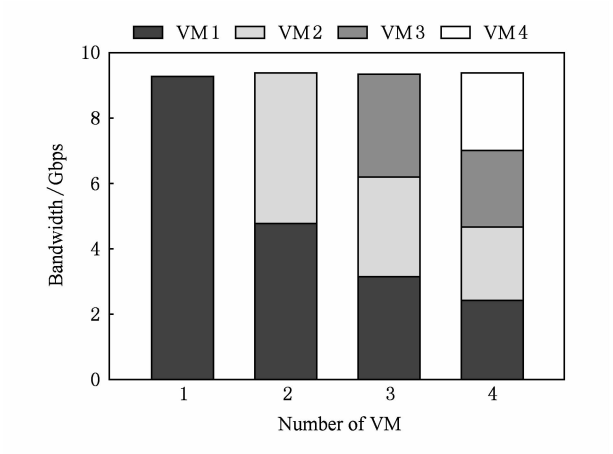


Fig. 13　Bandwidth allocation for virtual machines.

图 13　从内虚拟机作为 iperf client 时的带宽分配

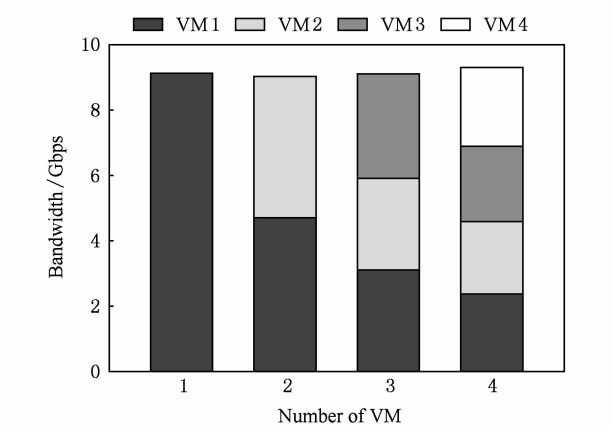


Fig. 14　Bandwidth allocation for virtual machines.

图 14　从内虚拟机作为 iperf server 时的带宽分配

服务器 3 作为 iperf server/client,服务器 1 和服务器 2 内的虚拟机都以直接 I/O 分配的方式使用宿主机分配得到的 SR-IOV 虚拟功能,作为 iperf client/server. 在保证每个虚拟机与 SR-IOV 虚拟功能一一对应的情况下,增加虚拟机数量;主根节点内虚拟机个数分别为 1,2,3,对应从属根节点内虚拟机个数为 3,2,1;测试得到各个虚拟机间的带宽分配情况如图 15 和图 16 所示.

图 13～16 中数据表明,本文 I/O 共享系统的设计实现可以良好地兼容现有的服务器虚拟化技术,可以在虚拟机以直接 I/O 分配模式使用 SR-IOV 虚拟功能时,为其提供总 TCP 带宽近乎物理网卡独立使用时的性能,同时在每个虚拟机使用相同数量的 SR-IOV 虚拟功能情况下,本文所实现的原型系统可以提供公平的 I/O 性能分配.

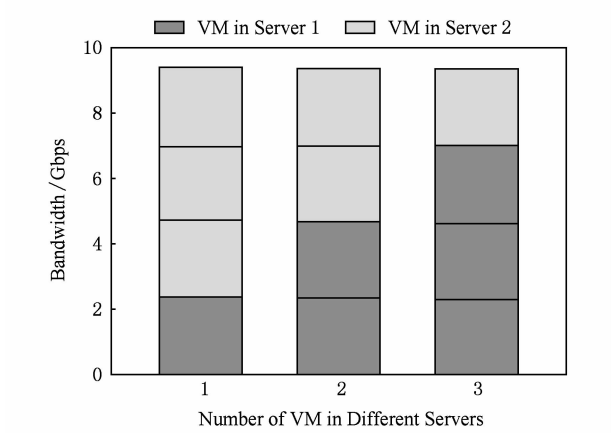


Fig. 15　Bandwidth allocation for virtual machines.

图 15　主从内虚拟机作为 iperf client 时的带宽分配

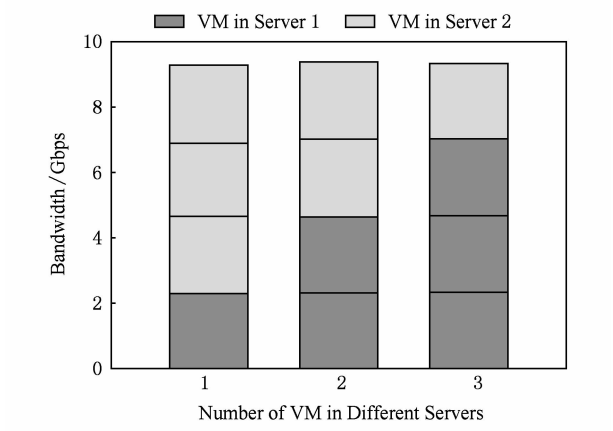


Fig. 16　Bandwidth allocation for virtual machines.

图 16　主从内虚拟机作为 iperf server 时的带宽情况

以上通过 3 方面的 iperf TCP 带宽性能测试,验证了本文方法能够在高效提供多根 SR-IOV 虚拟功能共享的同时,保证各个物理服务器之间公平的 I/O 性能分配.

6　总　　结

本文工作在 PCIe SR-IOV 技术基础之上,实现了 SR-IOV 设备虚拟功能在多个服务器间的多根共享和灵活分配,可以有效地解决单机虚拟化技术给数据中心服务器带来的 I/O 资源冗余、I/O 设备和 I/O 连线数量增加等问题,从而提高数据中心 I/O 资源利用率,降低单体服务器购置成本,提高数据中心的服务器密度. 本文进一步的工作是在现有硬件平台的基础之上,开发更灵活的 SR-IOV 设备虚拟功能分配技术,并为上层软件提供可定制的 I/O 服务质量保证.

参 考 文 献

[1] Chen G, Bozman J. Optimizing I/O virtualization: Preparing the datacenter for next-generation applications [OL]. 2009 [2014-03-24]. http://www.intel.com/content/www/cn/zh/virtualization/data-center-virtualization/i-o-virtualization-data center-paper.html

[2] Intel Corporation, Intel Open Source Technology Center 2008; Parallel Processing Institute of Fudan University. System Virtualization: Principle and Practice [M]. Beijing: Tsinghua University Press, 2009 (in Chinese)
(Intel 公司 2008; Intel 开源软件技术中心; 复旦大学并行处理研究所. 系统虚拟化: 原理与实现[M]. 北京: 清华大学出版社, 2009)

[3] Waldspurger C, Rosenblum M. I/O virtualization [J]. Communications of the ACM, 2012, 55(1): 66-73

[4] Shafer J. I/O virtualization bottlenecks in cloud computing today [C] //Proc of the 2nd USENIX Workshop on I/O Virtualization (WIOV'10). New York: ACM, 2010: 5-5

[5] Wang Guohui, Tse N. The impact of virtualization on network performance of Amazon EC2 data center [C] //Proc of INFOCOM 2010. Piscataway, NJ: IEEE, 2010: 1-9

[6] NextIO Corporation. Understanding network and storage fabric challenges in today's virtualized datacenter environment [OL]. 2011 [2014-03-24]. http://www.nextio.com/resources/white-papers

[7] Liu Jiuxing, Huang Wei, Abali B, et al. High performance VMM-bypass I/O in virtual machines [C] //Proc of the 2006 USENIX Annual Technical. Berkeley, CA: USENIX Association, 2006: 3-3

[8] Intel Corporation. Intel® virtualization technology for directed I/O architecture specification [OL]. 2013 [2014-03-24]. http://www.intel.com/content/www/us/en/intelligent-systems/intel-technology/vt-directed-io-spec.html

[9] Cohen A. I/O virtualization for next-generation datacenters [OL]. 2007 [2014-03-24]. download.microsoft.com/…/d/f/6/…/winhec2007_io-virt.doc

[10] PCI-SIG. Single root I/O virtualization and sharing 1.1 specification [OL]. 2010 [2014-03-24]. http://www.pcisig.com/specifications/iov/single_root/

[11] Wikipedia. Fibre channel over Ethernet [OL]. 2013 [2014-03-24]. http://en.wikipedia.org/wiki/Fibre_Channel_over_Ethernet

[12] Suzuki J, et al. ExpressEther—Ethernet-based virtualization technology for reconfigurable hardware platform [C] //Proc of the 14th IEEE Symp on High-Performance Interconnects (HOTI'06). Piscataway, NJ: IEEE, 2006: 45-51

[13] NextIO Inc. Today's virtualized datacenter environments: Understanding the network and storage fabric challenges [OL]. 2013 [2014-03-24]. http://www.nextio.com/products/vnet

[14] Suzuki J, Hidaka Y, Higuchi J, et al. Multi-root share of single-root I/O vritualization (SR-IOV) compliant PCI Express device [C] //Proc of the 18th IEEE Symp on High Performance Interconnects. Piscataway, NJ: IEEE, 2010: 25-31

[15] Satran J, Shalev L, Yehuda M, et al. Scalable I/O—A well-architected way to do scalable, secure and virtualized I/O [C] //Proc of USENIX 1st Workshop on I/O Virtualization (WIOV'08). Berkeley, CA: USENIX Association, 2008: 3-3

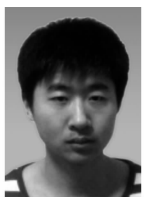
[16] PCI-SIG. Multi-root I/O virtualization and sharing 1.0 specification [OL]. 2008 [2014-03-24]. http://www.pcisig.com/specifications/iov/multi-root/

[17] Oracle. Oracle fabric interconnect [OL]. 2013 [2014-03-24]. http://www.oracle.com/us/products/networking/virtual-networking/fabric-interconnect/oracle-fabric-interconnect-ds-1873212.pdf

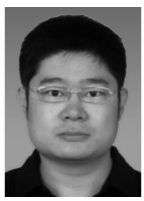
[18] Micron Technology, Inc. Micron's I/O virtualization technology: Helping to create a more manageable and reliable datacenter [OL]. 2012 [2014-03-24]. http://www.micron.com/about/blogs/2012/august/microns-io-virtualization-technology-helping-to-create-a-more-manageable-and-reliable-data-center

[19] PCI-SIG. PCI Express base specification revision 3.0 [OL]. 2010 [2014-03-24]. http://www.pcisig.com/specifications/pciexpress/base3/

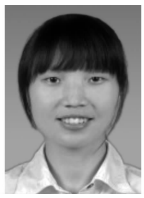
[20] The Board of Trustees of the University of Illinois. iperf [CP/OL]. [2014-03-24]. http://sourceforge.net/projects/iperf/



Wang Zhan, born in 1986. PhD candidate in the University of Chinese Academy of Sciences. Student member of China Computer Federation. His main research interests include virtualization technology and high performance interconnection.

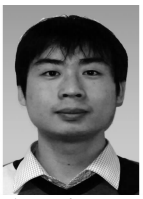


Cao Zheng, born in 1982. Received his PhD degree in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences in 2009. Associate professor. Member of China Computer Federation. His research interests include high performance computer architecture, high performance interconnection, and optical interconnection.

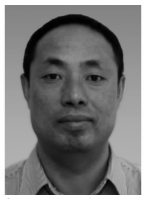


Liu Xiaoili, Born in 1986. Received her Master degree in electromagnetic fields and microwave technology from Beijing University of Posts and Telecommunications in 2011. Engineer. Member of China Computer Federation. Her major interests include IO Virtualization and high performance interconnection networks (liuxiaoli@ncic.ac.cn).

Su Yong, Born in 1976. PhD candidate in the University of Chinese Academy of Sciences. Engineer. Student member of China Computer Federation. His research interests include computer architecture and high performance interconnection networks (suyong@ncic.ac.cn).



Li Qiang, born in 1983. Received his PhD degree in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences in 2012. Assistant professor. His research interests focus on high performance communication (liqiang@ncic.ac.cn).



An Xuejun, born in 1966. PhD, senior engineer. Senior member of China Computer Federation. His research interests include computer architecture, high performance interconnection (axj@ncic.ac.cn).

Sun Ninghui, born in 1968. Professor and PhD supervisor. His main research interests include computer architecture, high performance computing and distributed OS (snh@ncic.ac.cn).

2015 年中国计算机学会人工智能会议(CCFAI 2015)

征文通知

中国计算机学会人工智能会议由中国计算机学会主办,中国计算机学会人工智能与模式识别专业委员会协办,每两年召开一次.本届会议将于 2015 年 8 月 21—23 日在山西省太原市举行,会议由山西大学计算机与信息技术学院、山西大学计算智能与中文信息处理教育部重点实验室联合承办.本次会议旨在为广大的人工智能研究人员提供了一个交流、合作的平台,汇聚从事人工智能理论与应用研究的人员,广泛开展学术交流,研讨发展战略,共同促进人工智能相关理论、技术及应用的发展.

本次会议录用的论文将推荐到《中国科学》、《计算机研究与发展》、《模式识别与人工智能》、《计算机科学与探索》、《计算机科学》、《中文信息学报》、《小型微型计算机系统》、《南京大学学报》(自然科学版)、《山东大学学报》(工学版)、《山西大学学报》(自然科学版)等期刊的正刊发表.

征文范围(包括但不限于)

人工智能理论基础	人工智能应用	智能机器人
Agent 理论及应用	智能控制与智能管理	机器感知与虚拟现实
生物信息学与人工生命	机器学习	数据挖掘
时空知识表示、推理与挖掘	社会网络分析及应用	神经网络与计算智能
人工免疫	粗糙集与软计算	图像和语音处理
模式识别	知识科学与知识工程	自然语言处理和机器翻译

征文要求

1) 论文必须未公开发表过,请勿一稿多投! 会议仅接受中文论文.

2) 论文应包括题目、作者信息、摘要、关键词、正文和参考文献.另附作者通讯地址、邮编、电话及 E-mail 地址.论文格式参考《计算机研究与发展》格式排版,一般不超过 6000 字.

3) 学生(不包括博士后和在职博士生)第一作者的论文稿件请在首页脚注中注明,否则将不具有参选"优秀学生论文"的资格.

4) 本会议采用在线投稿方式, 投稿地址: https://easychair.org/conferences/?conf=ccfai2015

重要日期

征文截止日期:2015 年 3 月 1 日

录用通知日期:2015 年 5 月 10 日

联系方式

联系人:郭虎升 (手机:13546351240)　　王文剑 (手机:13099070737)

通讯地址:山西省太原市坞城路 92 号　计算机与信息技术学院　030006

邮件联系:ccfai2015@126.com　　会议网址:http://ccfai2015.sxu.edu.cn