Qiang Su

+852 66787446 | jacksonsq97@gmail.com | https://qiangsu97.github.io/ | Hong Kong, China

Education

City University of Hong Kong — Hong Kong, China
Ph.D. in Computer Science — *Sep. 2016 – Present*

- Advisors: Prof. Chun Jason Xue and Prof. Hong Xu
- GPA: 4.15/4.30

Northeastern University — Shenyang, China
B.E. in Computer Science and Engineering — *Sep. 2014 – Jun. 2018*

- GPA: 4.1146/5.0
- Rank: 6/256

Publication

Qiang Su, Chuanwen Wang, Zhixiong Niu, Ran Shu, Peng Cheng, Yongqiang Xiong, Dongsu Han, Chun Jason Xue, Hong Xu, PipeDevice: A Hardware-Software Co-Design Approach to Intra-Host Container Communication, ACM **SIGCOMM** Demos and Posters, 2022.

Zhixiong Niu, **Qiang Su**, Peng Cheng, Yongqiang Xiong, Dongsu Han, Keith Winstein, Chun Jason Xue, Hong Xu, **NetKernel: Making Network Stack Part of the Virtualized Infrastructure**. IEEE/ACM Transactions on Networking (**IEEE/ACM ToN**), 2021. (**CCF A**, Top journal)

Zhixiong Niu, Hong Xu, Peng Cheng, **Qiang Su**, Yongqiang Xiong, Tao Wang, Dongsu Han, Keith Winstein, **NetKernel: Making Network Stack Part of the Virtualized Infrastructure**. Proceedings of the USENIX Annual Technical Conference (**USENIX ATC**), 2020. (**CCF A**, AR: 65/348 = 18.6%)

Work Experience

Microsoft Research Asia — Research Intern
Research on SmartNIC as a service — *Jan. 2022 - Jun. 2022*

City University of Hong Kong — Reaserch Assistant
Research on big data analytics and datacenter networking — *Oct. 2018 - Aug 2019*

Project

SmartNIC Virtualization — *ONGOING*

- *Exploring lightweight virtualization for SoC-based SmartNICs (e.g., Mellanox BlueField2 NIC).*
  This project aims to explore lightweight virtualization for SoC-based SmartNICs (e.g., Mellanox BlueField2 NIC). SmartNIC is increasingly popular in today's data centers as it can provides significant performance benefits for a variety of applications by offloading parts of software computation onto hardware, especially latency-critical applications, like microservices and network functions. However, it is still challenging to enable the usage of SmartNIC in public clouds and there is no proper virtualization mechanism. Problems including resource and performance isolation and offloading paradigm are still not well-solved.

PipeDevice — *COMPLETED (Cooperated with Microsoft Research)*

- *A hardware-software co-design approach to intra-host container communication.*
  Containers have become prevalent in public clouds due to the performance, portability, and deployment benefits compared to virtual machines. Containerized applications often entail extensive bulky data transfers to exchange intermediate results of data processing among peers. Examples include the shuffle stage in MapReduce jobs and the model update process with parameter server and allreduce in distributed machine learning.
  Moreover, it is increasingly common for these bulky transfers to occur in the intra-host scenario. With the server machines becoming more resourceful in terms of CPU cores and memory, hundreds of containers can reside on the same server. Cloud operators also tend to consolidate a tenant's containers to as few servers as possible in order to improve performance and management efficiency. For example, Spark distributes mappers and reducers in multiple containers that may co-locate at the same server; Kubernetes launchs multiple containers inside a Pod where each Pod resides in a physical or virtual machine in the cluster, and multiple Pods may also co-locate on the same machine. As a result, PipeDevice focuses on bulky transfers in intra-host container communication which is relatively less explored in the community.
  Containers rely on the host TCP/IP stack which is well-known to incur high CPU overheads in data centers. To reduce such overheads, PipeDevice exploits FPGA accelerators that have already been connected to each server in the cloud to forward data across co-located containers, effectively creating a device that facilitates a communication pipe for them, without introducing new hardware deployment cost. PipeDevice relies on hardware offloading: each socket is allocated dedicated memory out of a hugepage region in the hypervisor,

destination memory address. This eliminates the overheads of 1) copy between user and kernel spaces and 2) TCP stack. Further, PipeDevice keeps the connection states in host DRAM and manages them entirely in software, so that data copy is performed by hardware in a stateless manner. This eliminates contention of the hardware resources and resolves the connection scalability issue.
Therefore, PipeDevice represents a general hardware-software co-design approach that is deployable in today's public clouds. It is entirely possible that PipeDevice's hardware functionality be implemented on other hardware, such as programmable NICs (SmartNICs) and RDMA NICs.

- *Role*: The architect of the whole system; the 1st committer.
- *Technical contributions*:
  Propose new solutions to low-overhead intra-host container communication in public clouds;
  Design the system architecture and key abstractions to applications;
  Implement the host-side modules, e.g., the communication library and the kernel driver that manages the host resources and communicates with hardware;
  Implement a network function chain based on PipeDevice, which demonstrates its benefits.

NetKernel *COMPLETED*

- *A system that decouples the network stack from the guest VM and offers it as an independent module.*
  In today's public clouds, the virtual NIC (vNIC) is the key abstraction boundary between the tenants and the provider. Then the tenant runs a full network stack of its own within the virtual machine (VM), and the provider is responsible for allocating resources across VMs, and managing the network fabric. Yet the vNIC-centric paradigm suffers from two fundamental limitations and is facing increasingly difficult challenges: 1) The task of maintaining the network stack falls largely under the responsibility of tenants. This is tedious and difficult given the intricacy of network stack implementation (e.g., mTCP and kernel TCP/IP) and the velocity of new feature development (e.g., new congestion control algorithm). We believe that none of the above should fall upon the tenant in most usage scenarios. 2) The provider has almost zero visibility and control of the network stack. This poses significant challenges to many management tasks and degrades the efficiency of running the cloud network fabric. Even with sufficient resources and expertise the provider cannot deploy optimized network stack extensions for tenants, although many such implementations are available (e.g., scalability and zero-copy datapath). Therefore, from a philosophical standpoint it is worth asking: Is there a better way to provision the network stack in public clouds, to make life easier for both tenant and provider?
  NetKernel provides a new paradigm: network stack as a service in the cloud. It uses network APIs such as standard BSD sockets as a better abstraction boundary between the tenant and provider, and allows us to decouple the network stack from the guest OS, offering it as an independent service by the provider. Packets are handled outside the tenant VM in a network stack module (NSM) given by the provider, whose design and implementation are transparent to tenants. Tenants just use the network stack by invoking the APIs, while the provider does all the heavy lifting and may charge a fee to recover the cost of additional resources and service. NetKernel improves network management efficiency for operator, and provides deployment and performance gains for users.
- *Role*: One of the architects of the whole system; the 2nd committer.
- *Technical contributions*:
  Develop a testbed that uses kernel TCP/IP stack and mTCP as two network stack modules (NSMs);
  Implement the key forwarding module named CoreEngine, which quickly switches the traffic between guest VM and NSMs.

Honer and Award

Scholarship

* China National Encouragement Scholarship (top 1%), 2017
* Second-Class Scholarship of NEU (top 5%), 2017
* Chinese National Scholarship (top 1%), 2016
* Second-Class Scholarship of NEU (top 5%), 2016
* "Zhong Tian Gang Tie" Educational Scholarship (top 1%), 2015
* First-Class Scholarship of NEU (top 1%), 2015

Competition

* Honorable Mention of Interdisciplinary Contest in Modeling(ICM) for international college students, 2016

Leadership

* Excellent Student Leader in NEU, 2017
* Excellent Student in NEU, 2016
* Pacesetter of Excellent Student in NEU, 2015

Teaching

- CS5222 Computer Networks & Internets, Fall 2022
- CS5222 Computer Networks & Internets, Fall 2021
- CS3402 Database Systems, Spring 2021
- CS1302 Introduction to Computer Programming, Fall 2020
- CS1102 Introduction to Computer Studies, Spring 2020
- CS2311 Computer Programming, Fall 2019

SKILL

- Programming: C | Python | Linux Kernel Programming | Docker | PyTorch
- Language: Bilingual in English and Chinese