Qiang Su 苏强 ● +852-66787446 ● jacksonsq97@gmail.com ● https://qiangsu97.github.io/

● G7352 #1, Yeung Kin Man Academic Building, City University of Hong Kong, Kowloon

EDUCATION

City University of Hong Kong, Department of Computer Science Hong Kong, China
Ph.D. in Computer Science 09/2019 ~ Present
Adviors: Prof. Hong Xu and Prof. Chun Xue
GPA: 4.15/4.30

Northeastern University (NEU), School of Computer Science and Engineering Shenyang, China
B.E. in Computer Science and Engineering 09/2014~06/2018
GPA: 4.1146/5.0 | Rank: 6/256

PUBLICATION

[1] 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)

[2] 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%)

WORKING EXPERIENCE

Research Assistant City University of Hong Kong | 10/2018~08/2019

- Research on Big Data Analytics and Datacenter Networking
  Advised by Prof. Hong Xu

PROJECT

SmartNIC Virtualization [ongoing]

- 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 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 public usage of SmartNIC and there is no proper virtualization mechanism. Problems including resource/performance isolation and offloading paradigm are still not well-solved.

PipeDevice [completed]

- This project aims to *achieve deployable low-overhead intra-host container communication in public clouds using hardware-offloading techniques*. 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, and application data in the socket buffer is directly accessed by the FPGA’s DMA engine and copied to the 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.

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

NetKernel [completed]

- This project aims to *decouple the network stack from the guest VM and offer it as an independent module*. It uses use 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 tenants.
- My contributions
  Role: One of the architects of the whole system; the 2nd committer.
  Technical contributions:
  1) Develop a testbed that uses kernel TCP/IP stack and mTCP as two network stack modules (NSMs);
  2) 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 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 | TensorFlow | PyTorch
- Language: Bilingual in English and Chinese