Zhuang Wang

Email： tmwangzhuang@outlook.com
Homepage: https://alvenwong.github.io/

Education

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China 09/2014 – 06/2017
Master of Engineering, Computer Science
GPA: **87.90/100.00,** Advised by Prof. Mingyu Chen
Master thesis: An Application-driven Flow Scheduling in Data Centers (in Chinese)

Huazhong University of Science and Technology, Wuhan, China 09/2010 – 06/2014
Bachelor of Engineering, Computer Science
GPA: **87.66/100**; Ranking: **7/320**

Publications

1. **Zhuang Wang**, Ke Liu, Long Li, Weiyi Chen, Mingyu Chen, Lixin zhang, "A Novel Approach for All-to-All Routing in All-optical Hypersquare Torus Network," in *Proc. of ACM International Conference on Computing Frontiers (CF)*, 2016.
2. Ke Liu, **Zhuang Wang**, Jack Y. B. Lee, Mingyu Chen, Lixin Zhang, "Adaptive Rate Control over Mobile Data Networks with Heuristic Rate Compensations," in *Proc. of IEEE/ACM International Symposium on Quality of Service (IWQoS)*, 2016.
3. **Zhuang Wang**, Ke Liu, Yifan Shen, Jack Y. B. Lee, Mingyu Chen, Lixin Zhang, "Intra-host Rate Control with Centralized Approach," in *Proc. of IEEE Cluster* 2016, short paper.
4. **Zhuang Wang**, Weifa Liang, Meitian Huang, Yu Ma, "Delay-Energy Joint Optimization for Task Offloading in Mobile Edge Computing," in CoRRabs/1804.10416 (2018)

Professional Experiences

Brown University 08/2018 – present
Advised by Prof. Theophilus Benson

- Probe TCP flows and packets information from the kernel with eBPF
- Trace packet transmission across different layers in the kernel
- Measure NIC-to-NIC latencies
- Diagnose performance problems in microservices architecture

Australian National University, Canberra, Australia 08/2017 – 05/2018
Research Assistant, Advised by Prof. Weifa Liang

- Designed task offloading algorithms in Mobile Edge Computing

Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China 01/2014 – 05/2014
Advised by Prof. Binzhang Fu

- Worked on congestion prediction in Network on Chips with machine learning methods

Awards and Honors

Award	Organization	Date
National Scholarship	Ministry of Education of P. R. China	2011
National Encouragement Scholarship	Ministry of Education of P. R. China	2013
National Scholarship	Ministry of Education of P. R. China	2016
International Postgraduate Research Scholarship (IPRS)	Australian government	2017

Selected Projects

μBPF, a performance diagnosis framework in microservices architecture [**GitHub**]

μBPF leverages eBPF to capture TCP metrics from the kernel for performance diagnosis in microservices architectures. It probes two levels of metrics: flows and packets. The flow-level metrics include sixteen elements, such as flight size, CWnd and sampled RTT. The packets-level metrics are the breakdown of RTTs, including latencies in TCP layer, IP layer, MAC layer and the network (from NIC to NIC). μBPF solved several challenges, such as network address translation in microservices architecture, clock synchronization and trace sampling.

Haiyun, a labeled network system [GitHub]

Haiyun enables NICs to distinguish packets with different service-level agreements (SLAs). With the support of DPDK and mTCP, Haiyun could significantly reduce the long tail delay in data centers by keeping packets with different SLAs in different queues from NICs to applications.

My work consists of two components. One is a flow generator, which could generate more than 1 million concurrent TCP connections with just one physical server. The other is a third-party tool to measure the performance of Haiyun. To measure the delay resulted from packets processing in servers, we use switches' mirror function to monitor packets and measure their delays on a specific server. To reduce the overheads of operating system, we leverage DPDK to bypass the kernel to optimize the performance of this tool.

Panda, a flow scheduling in data centers [GitHub]

Panda emulates shortest job first (SJF) scheduling to optimize the average flow completion time for throughput-intensive applications on the premise that flow information is not known *a priori*. Also, it bounds low latency for delay-sensitive applications by leveraging the distinct flow size distributions of the two kinds of applications.

At its heart, Panda derives an optimal threshold to divide packets into two categories: large and small, ensuring that small packets dominate traffic from delay-sensitive applications and large ones dominate traffic from throughput-intensive applications. Panda allocates each flow a counter initiated with zero. Large packets increase the counter while small packets decrease it. Then Panda assigns priorities to flows according to their counters.