A New Perspective on Supporting QoS in Architecture: Computer as a Network

Tianni Xu, Jiuyue Ma, Zhicheng Yao, Xiufeng Sui, Yungang Bao
Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS)

I. INTRODUCTION

Contemporary data centers confront with challenges in managing the trade-offs between resource utilization and applications' quality of services (QoS). To resolve this issue, as suggested in the community white paper "21st Century Computer Architecture" [1], computer architecture needs to provide new, higher-level interfaces beyond a conventional instruction set architecture (ISA) to convey an application's QoS requirements to the hardware.

To address the challenge, we proposes a new computer architecture *PARD (programmable architecture for resourcing-on-demand)* that provides such a new programming interface. *PARD* is inspired by the perspective that a computer is inherently a network in which hardware components communicate via packets (e.g., over the NoC or PCIe), as shown in Figure 1. So we can apply networking technologies, e.g., principles of software-defined networking (SDN) [2], to this intra-computer network.

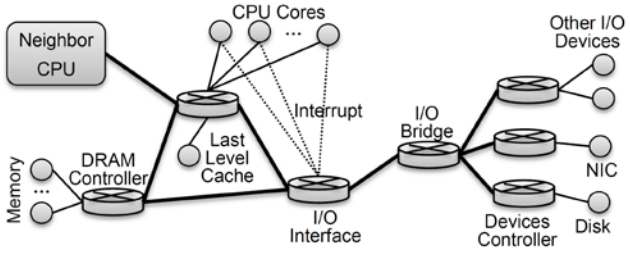

Figure 1. A traditional computer can be viewed as a network because hardware components communicate via packets, e.g., over the networks-on-chip (NoCs), PCIe or QPI etc.

Nevertheless, compared with deploying SDN in a network, there are at least three challenges in applying SDN to computer architecture.

- First, how to tag packets generated by various hardware components? In a computer, a number of different hardware components can generate different types of packets.
- Second, how to figure out a uniform control plane structure for different hardware components such as LLC, memory and I/O devices? Unlike network routers that perform almost the same store-and-forward behavior, hardware components in a computer behave differently.
- Finally, how to devise a uniform programmable interface for accessing control planes? For networks, a network router already leverages a firmware to facilitate users to access and configure its control plane. However, conventional computers lack similar firmware.

In our final poster, we will present solutions to these challenges. Specifically, *PARD* includes three mechanisms: (1) attaching a high-level semantic tag (e.g., a virtual machine or thread ID) to each memory-access, I/O, or interrupt packet, (2) devising a table-based programmable control planes that can be integrated into various shared resources (e.g., cache, DRAM, and I/O devices), (3) abstracting all control planes as a device file tree to provide a uniform programming interface via which users create and apply tag-based rules. With these mechanisms, *PARD* enables new functionalities like fully hardware-supported virtualization and differentiated services in computers. (More details are in [3]). Table 1 highlights several new features of PARD compared with traditional servers.

Table 1. A comparison of PARD servers and conventional servers.

	Conventional Server	PARD Server
Virtualization	SW Supported	HW Supported
Perf. Isolation	Unsupported	HW Surpported
Monitoring	High overhead	Realtime
Perf. Adaption	Coarse-grained	Fine-grained

Finally, the poster will present our ongoing work, how PARD's new features influence software stack, including hypervisors, OS kernels and cluster management systems.

REFERENCES

[1] Computing Community Consortium (CCC). 21st century computer architecture. A community white paper, 2012. URL: http://cra.org/ccc/docs/init/21stcenturyarchitecturewhitepaper.pdf

[2] Software-Defined Networking. https://www.opennetworking.org/sdn-resources/sdn-definition/

[3] Jiuyue Ma, Xiufeng Sui, Ninghui Sun, Yupeng Li, Zhihao Yu, Bowen Huang, Tiani Xu, Zhicheng Yao, Yun Chen, Haibin Wang, Lixing Zhang, Yungang Bao, Supporting Differentiated Services in Computers via Programmable Architecture for Resourcing-on-Demand (PARD) , *in the 20th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2015.