Enhancing an Out-of-Order Processor for Latency-Critical Cloud Applications

Prof. Mattan Erez (mattan.erez@utexas.edu)

Billions of dollars are invested every year into building warehouse scale computers (WSCs). By amortizing power, cooling, and management overhead, WSCs promise significantly higher cost-efficiencies compared to private datacenters and attract applications with a large range of performance characteristics. One important category of WSC workloads includes a combination of latency-critical and latency-noncritical tasks. Ideally, noncritical tasks are used to "backfill" compute resources to fully utilize the WSC. Unfortunately, this is often difficult to achieve while still maintaining the performance goals of the latency-critical tasks because their performance degrades from resource interference. In fact, hardware is often intentionally over-provisioned to ensure quality-of-service (QoS) goals for latency-critical tasks, and the resulting under-utilization translates into huge wastes of system capacity and capital investment. As a result. performance interference bottlenecks the system utilization and causes significant loss in cost-efficiencies of datacenters.

In this project, you will explore opportunities in improving cost-efficiencies of datacenters from an architectural perspective. Your goal is to extend and optimize existing features in current commercial processors to enable safe colocation of latency-critical and latency-noncritical tasks; i.e., keep the performance of latency-critical tasks close to standalone execution, while latency-noncritical tasks make as much progress as possible.

A holistic architectural design is needed to achieve this goal. Specifically, it is likely that three key components will be required:

- In-pipeline resource arbitration and rollback
- Intelligent cache management
- QoS-aware memory subsystem

The design space is large, but guidance and initial thoughts will be provided. There are also multiple paths to realize and evaluate the designs. The most straightforward is to enhance an open-source academic simulator and a mix of public workloads with various resource requirements and performance characteristics. We can also build promising solutions on top of open-source FPGA designs for a more extensive and challenging evaluation.