COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

DISSERTATION DEFENSE

AMIRHOSSEIN MIRHOSSEINI

Datacenter Architectures for the Microservices Era

Thursday, March 4, 2021
12:00 PM
Virtual (Passcode: 1337)

ABSTRACT: Modern internet services are shifting away from single-binary, monolithic services into numerous loosely-coupled microservices that interact via Remote Procedure Calls (RPCs), to improve programmability, reliability, manageability, and scalability of cloud services. Computer system designers are faced with many new challenges with microservice-based architectures, as individual RPCs/tasks are only a few microseconds in most microservices. In this dissertation, I seek to address the most notable challenges that arise due to the dissimilarities of the modern microservice-based and classic monolithic cloud services, and design novel server architectures and runtime systems that enable efficient execution of μs-scale microservices on modern hardware.

In the first part of my dissertation, I seek to address the problem of Killer Microseconds, which refers to μs-scale "holes" in CPU schedules caused by stalls to access fast I/O devices or brief idle times between requests in high throughput microservices. Whereas modern computing platforms can efficiently hide ns-scale and ms-scale stalls through micro-architectural techniques and OS context switching, they lack efficient support to hide the latency of μs-scale stalls. I propose Duplexity as a heterogeneous server architecture that employs aggressive multithreading to hide the latency of killer microseconds, without sacrificing the tail latency of cloud microservices.

Next, I comprehensively investigate the problem of tail latency in the context of microservices and address multiple aspects of it. First, I introduce Q-Zilla, a scheduling framework to tackle the tail latency of μs-scale microservices from a queuing perspective. Queuing is a major contributor to end-to-end tail latency, wherein nominal tasks are enqueued behind rare, long ones, due to Head-of-Line (HoL) blocking. Q-Zilla is composed of the Server-Queue Decoupled Size-Interval Task Assignment (SQD-SITA) scheduling algorithm and the Express-lane Simultaneous Multithreading (ESMT) microarchitecture, which together seek to address HoL blocking by providing an "express-lane" for short tasks, protecting them from queuing behind rare, long ones. By combining the ESMT microarchitecture and the SQD-SITA scheduling algorithm, Q-Zilla significantly reduces the tail latency of μs-scale microservices, compared to the state-of-the-art server architectures.

Finally, I investigate the tail latency problem of microservices from a cluster, rather than server-level, perspective. Whereas Service Level Objectives (SLOs) define end-to-end latency targets for the entire service to ensure user satisfaction, with microservice-based applications, it is unclear how to scale individual microservices when end-to-end SLOs are violated. I introduce Parslo as an analytical framework for partial SLO allocation in virtualized cloud microservices. Parslo takes a microservice graph as an input and employs a Gradient Descent-based approach to allocate "partial SLOs" to different microservice nodes, enabling independent auto-scaling of individual microservices. Parslo achieves the optimal solution, minimizing the total cost for the entire service deployment, and is applicable to general microservice graphs.

CHAIR: Prof. Thomas F. Wenisch