Wenyuan Shao

2782 Patrick Henry St, Auburn Hills, Michigan, 48326
shaowy@gwu.edu, (202) 766-5336

Education

George Washington University, Washington, DC — Jan 2019 – May 2025
PhD, Computer Science
Advisor: Prof. Gabriel Parmer

- *Research Interests:* Real-time Operating Systems, Edge Computing, Cloud Computing, High Performance Networking, Scheduling, Isolation, Reliable Computing; Embedded Systems, OS Security, Network Security.

George Washington University, Washington, DC — Jan 2017 – Dec 2018
MS, Computer Science

Tongji University, Shanghai, China — Sep 2012 – Jun 2016
BS, Software Engineering

Publications & Research

My research focuses on providing high throughput and latency-sensitive OS supports on multi-tenant edge clouds while maintaining both strong spatial and temporal isolation with limited resources. The majority of research lies in isolation, control plane abstractions, and scheduling on the edge cloud. I am the primary researcher on the edge cloud variant of the Composite μ-kernel.

- **Wenyuan Shao**, Bite Ye, Huachuan Wang, Gabriel Parmer, Yuxin Ren, "Edge-RT: OS Support for Controlled Latency in the Multi-Tenant, Real-Time Edge", at the 44th IEEE Real-Time Systems Symposium (RTSS) 2022, **Best Student Paper**.
- **Wenyuan Shao**, Xinyu Han, Evan Stella, Linnea Dierksheide, Phani Kishore Gadepalli, Gabriel Parmer, "Janus: OS Support for a Secure, Fast Control-Plane", at the 31st IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2025.
- Yuxin Ren, Guyue Liu, Vlad Nitu, **Wenyuan Shao**, Riley Kennedy, Gabriel Parmer, Timothy Wood, "Fine-Grained Isolation for Scalable, Dynamic, Multi-tenant Edge Clouds", at the 2020 USENIX Annual Technical Conference.
- Gabriel Parmer, Runyu Pan, Yuxin Ren, Phani Kishore Gadepalli, **Wenyuan Shao**, "Component-based OS Design for Dependable Cyber-Physical Systems", in Proceedings of the 1st International Workshop on Next-Generation Operating Systems for Cyber-Physical Systems (NGOSCPS), 2019.

Computer Skills

Languages:
- *Proficient in:* C, C++.
- *Familiar with:* Shell-scripting, Python, Java, JavaScript, XML, HTML, Matlab.

Software:
- *Platforms:* Linux (various flavors), Windows, MacOS
- *Embedded Platforms:* Composite, Linux RT, μC/OS, seL4.
- *Tools:* GCC/Clang, GDB, Git, Various network workbenches.

Research Projects

Fine-Grained Isolation for Scalable, Dynamic, Multi-tenant Edge Clouds
Achievements: We reduced startup latency by **170X** compared to Linux processes, and improve latency by **three orders of magnitude** when running 300 to 1000 edge-cloud Memcached instances on one server.
Motivation: 5G edge clouds promise a pervasive computational infrastructure a short network hop away, enabling a new breed of smart devices that respond in real-time to their physical surroundings. Unfortunately, today's operating system designs fail to meet the goals of scalable isolation, dense multi-tenancy, and high performance needed for such applications.
Research: In this paper, we introduce EdgeOS that emphasizes system-wide isolation as fine-grained as per-client. We propose a novel memory movement accelerator architecture that employs data copying to enforce strong isolation without performance

penalties. To support scalable isolation, we introduce a new protection domain implementation that offers lightweight isolation, fast startup, and low latency even under high churn. We implement EdgeOS in a micro-kernel-based OS and demonstrate running high-scale network middleboxes using the Click software router and endpoint applications such as Memcached, a TLS proxy, and neural network inference.

Contributions:

- A carefully optimized "Memory Movement Accelerator" (MMA) communication and buffer management architecture that enforces isolation with data copying, while retaining high throughput, and low latency.
- A "Feather Weight Process" (FWP) that redefines the process abstraction to a minimal memory footprint and set of capabilities needed to support dense deployments of edge computation, and provides strong isolation between each client in multi-tenant environments.
- A control plane with flexible routing and FWP chain caching to support microsecond speed initialization of complex services in high churn environments.

Technical Skills: In this project, I am mainly responsible for WebServer Applications which demonstrates the performance of EdgeOS, including:

- Adaptation of a lightweight TLS lib axTLS to EdgeOS which includes a lightweight HTTPs-based web proxy optimized for embedded systems, and the lwip TCP/IP networking stack.
- Adjustment of the DPDK implementation in EdgeOS to enable seamless integration and functionality with both lwIP and axTLS.
- Adaptation of the ARM CMSISNN neural network inference lib to EdgeOS, and use the example CIFAR10 configuration which takes as input a 32x32 pixel color image which classifies to number 09.

Edge-RT: OS Support for Controlled Latency in the Multi-Tenant, Real-Time Edge

Achievements: Compared to Linux and EdgeOS, Edge-RT can both maintain higher throughput and meet significantly more deadlines both for systems with bimodal workloads with utilization **above 60%**, in the presence of malicious tasks, and as the system scales up in clients.

Motivation: Embedded and real-time devices in many domains are increasingly dependent on network connectivity. The ability to offload computations encourages Cost, Size, Weight, and Power (C-SWaP) optimizations, while coordination over the network effectively enables systems to sense the environment beyond their own local sensors, and to collaborate globally. The promise is significant: Autonomous Vehicles (AVs) coordinating with each other through infrastructure, factories aggregating data for global optimization, and power-constrained devices leveraging offloaded inference tasks. Low-latency wireless (e.g., 5G) technologies paired with the edge cloud, are further enabling these trends. Unfortunately, computation at the edge poses significant challenges due to the challenging combination of limited resources, required high performance, security due to multi-tenancy, and real-time latency.

Research: This paper introduces Edge-RT, a set of OS extensions for the edge designed to meet the end-to-end (packet reception to transmission) deadlines across chains of computations. It supports strong security by executing a chain per-client device, thus isolating tenant and device computations. Despite a practical focus on deadlines and strong isolation, it maintains high system efficiency. To do so, Edge-RT focuses on per-packet deadlines inherited by the computations that operate on it. It introduces mechanisms to avoid per-packet system overheads, while trading only bounded impacts on predictable scheduling.

Contributions:

- A system design that (1) focuses on per-packet end-to-end deadline scheduling with dynamic, dense workloads, and (2) minimizes per-message system overheads.
- The mechanisms and abstractions for predictable buffering, inter-core coordination, and scheduling that enable efficient packet processing.

- The implementation of Edge-RT and the parameter studies to understand system overhead trade-offs to guide the system's configuration.
- The evaluation of Edge-RT for various workloads compared with Linux and EdgeOS.

Technical Skills:

- Implementation of a deadline-aware software DMA which runs on a dedicated core whose job includes copying packets between FWPs (Feather-Weight Process), send notifications to the corresponding core and keep track of the deadline for each packets.
- Adaptation of the kernel-bypass DPDK library for fast packet transactions while enabling end-to-end scheduling of packets.
- An implementation of the scheduling algorithm that executes in constant time, irrespective of the number of tasks.
- Multi-tenant Linux compare cases through virtual NICs provided by SR-IOV and Open vSwitch paired with DPDK.

Janus: OS Support for a Secure, Fast Control-Plane

Achievements: Compared to a Linux approach, a specialized latency-sensitive control plane using Janus provides over a **6x** improvement in throughput, while providing 99th percentile tail latencies almost **3x** lower. In a multi-tenant system, tail latency improves by **orders of magnitude**.

Motivation: The emergence of the edge cloud, empowered by advanced wireless technologies such as 5G, is aimed at delivering predictable latency-sensitive services with the potential for interaction within low milliseconds. Even in the traditional cloud, latency-sensitive services are pervasive. To enable low-latency software, much focus has been on data-plane optimizations for fast processing of requests. Unfortunately, these efforts alone are insufficient: effective low-latency service also require advances in the control-plane. However, control-plane operations require strong spatial and temporal isolation between tenant computations, thus can result in significant overhead.

Research: This paper introduces Janus an OS abstraction for a flexible control-plane which provides strong isolation as well as managed latency through the use of pervasive kernel-bypass. Janus leverages hardware Memory Protection Key (MPK) to enable low-cost control operations for protected procedure call (PPC) and thread dispatch – two essential building blocks of spatial and temporal isolation. By transparently improving these fundamental control-plane operations, Janus enables efficient and predictable user-defined and customizable system control policies and mechanisms, while maintaining strong isolation. We evaluate Janus's ability to define new control operations, increase the efficiency of an existing RTOS, and support low-latency services in a Memcached server.

Contributions:

- Janus introduces a latency sensitive and customizable control-plane abstraction for managing execution of multiple, latency-sensitive and untrusted tenants.
- Janus transparently integrates the existing isolation mechanisms in a security-centric OS with those provided by hardware extensions for memory isolation (MPK), enabling efficient and predictable extension of control-plane policies.
- We evaluate Janus and compare it with existing µ-kernels, a µ-kernel-based RTOS, and Linux.

Technical Skills:

- The implementation of user-level IPC callgate and thread dispatch gate based on hardware architecture specific features.
- Adapt the well-known in-memory key-value store Memcached to Janus and setting up multi-tenant Memcached by separating tenant computation into components in Composite and Janus.
- Setting up and adapting other real-time operating systems such as seL4 for compar-

ison cases, and implement L4-style IPC in Janus.

- Leverage Patina which provides predictable interfaces including communication channels, event management, and synchronization to demonstrate performance of Janus on basic interfaces for embedded systems.

Experience

SRC Research Scholars Program — Research Scholar
Washington, DC — Sep 2019 – Sep 2021

My research focuses on: Fine-Grained Isolation for Efficient, Low-latency Processing from the Micro-controller to the Edge. The key ideas of my research are creating an Edge system infrastructure based on lightweigted process and allowing deadline-aware scheduling for separate flows.

QAD China — Software Engineer Internship
Shanghai, China — Jun 2015 – May 2016

Contribute to the enhancement of the legacy serialization system by identifying and implementing potential improvements. Developed a prototype website utilizing AJAX and JavaScript, incorporating drag-and-drop functionality for managing products and packages.