Parul Sohal

Ph.D. Computer Systems — Performance Isolation — Data-Driven Resource Management
psohal@bu.edu — linkedin.com/in/parulsohal — scholar.google.com/psohal — (+1) 650-878-7384
U.S. Permanent Resident (No sponsorship required)

Systems researcher focused on profile-driven resource management for shared-memory systems. Developed runtime mechanisms that analyze hardware performance counter data, detect workload phases, and translate these measurements into resource allocation policies that reduce contention for shared resources across real-time and cloud platforms. My work emphasizes data-driven design and rigorous evaluation, delivering quantifiable improvements in performance, predictability, and resource efficiency. Interested in extending these systems with machine learning models and AI-assisted tools to analyze large profiling datasets and automatically derive resource management policies for modern large-scale workloads.

Education

Boston University, Ph.D. Boston, MA
Computer Science, Systems *Sept 2017– May 2026*
Advisors: Orran Krieger, Renato Mancuso

Lewis & Clark College, B.A. Portland, OR
Computer Science and Mathematics *Sept 2013– May 2017*

Technical Skills

Systems: Hypervisors, Resource Isolation, Performance Profiling, Linux Systems Programming
Hardware: Intel RDT (CAT, MBA), ARM QoS, Cache Partitioning, DRAM Bandwidth Regulation, PMUs
Languages: C, Python, Bash
ML/AI: Claude Code, Codex, PyTorch
Tools: `Perf`, `SystemTap`, Benchmarking, Git
Platforms: Multi-core x86 (Cascade Lake, Ice Lake), ARM-based SoCs (NXP S32V234)
Awards Best Student Paper, IEEE RTSS 2020; UWC Davis Scholarship (2013–2017)

Experience

Doctoral Researcher, Selected Projects Sept 2017 – Present
Boston University Boston, MA

- **ProPer**: Architected and led the development of *Profile-driven Performance Management Framework*, a system that derives cache and DRAM bandwidth allocation policies for cloud workloads. Built a phase-aware profiling and evaluation pipeline using hardware performance counters and function boundaries collected across runs to translate fine-grained measurements into automated resource control decisions and analyze memory behavior at scale across synthetic benchmarks and real workloads. Integrated enforcement using Intel CAT and a custom userspace DRAM bandwidth regulator, reducing worst-case slowdown under 20-way contention from 6× to 2.5×. Currently extending the system with machine learning models and AI-assisted tools to automate resource management from profiling data.
- **E-WarP** [Code]: Deployed an end-to-end, profile-driven resource management framework for heterogeneous embedded systems to enforce predictable performance under shared main memory contention. Implemented a microsecond-resolution hardware counter profiler into a partitioning hypervisor with ¡1% overhead. Designed an *Envelope-aWare Predictive model* to provide less pessimistic estimates of worst-case execution time of an application under a given bandwidth quota. Coordinated software-based regulation with hardware ARM QoS mechanisms to manage DRAM bandwidth across CPUs and accelerators.
- **Unikernel Linux (UKL)** [Code]: Collaborated with a Ph.D. researcher to develop the initial Linux kernel target for a unikernel-style execution model, enabling a single optimized application to link directly with the kernel and execute with supervisor privileges in a unified address space. Contributed to kernel build system modifications and runtime integration to support direct application–kernel linkage while preserving compatibility with the broader Linux ecosystem. Delivered a lightweight execution environment suitable for both virtualized and bare-metal deployments.

Software Engineer Intern Sept 2019 – Dec 2023
Red Hat Boston, MA

- Collaborated with Red Hat engineers to evaluate and refine cache and DRAM bandwidth management mechanisms in Linux and hypervisor environments across embedded and cloud platforms.