PreClass: Static Pre-Admission Classification and Lightweight Runtime Guards for Safe Multi-Tenancy on Shared GPUs

Ali Yazdanpanah
Supervisors: Farshad Khunjush, Mohsen Raji
Department of Electrical and Computer Engineering, Shiraz University

Abstract

GPU systems frequently suffer from performance interference and out-of-memory (OOM) failures when schedulers co-locate arbitrary workloads on the same device. We present **PreClass**, a pre-execution system that *steers safe GPU co-location with minimal online profiling.* PreClass predicts whether an incoming job will be compute- or memory-bound using only submission-time signals, *kernel names, launch configurations, input sizes, batch size, framework*, and a *lightweight static arithmetic-intensity estimate (with tensor-op flags)*, extracted from job specifications or shallow binary analysis. Decisions are made before launch, avoiding profiler overhead while capturing the dominant interference modes that drive co-run safety.

To improve memory safety under multi-tenancy, PreClass deploys a transparent user-space CUDA wrapper (`LD_PRELOAD`) that interposes major allocation paths (`cudaMalloc`, `cudaMallocAsync`, `cuMemAlloc*`, memory pools, and CUDA Graph APIs). The wrapper enforces per-job VRAM quotas, supports an optional small guard-band to reduce peer-induced failures, and logs launch metadata with $< 2\%$ overhead, providing best-effort limits in user space (device-side allocations and unified memory remain risks). For capacity and health guardrails, PreClass performs *low-rate, device-level NVML sanity checks* at admission (and optionally during early execution) to verify free VRAM, utilization, and thermal/power/ECC status; these checks are best-effort and do not collect per-application telemetry. PreClass integrates with the NVIDIA Container Toolkit and `cgroups-v2` for process and device isolation and exposes a Kubernetes admission webhook with conservative fallbacks for missing metadata. Our testbed does not use NVIDIA MIG (a common setting where MIG is unavailable or too coarse); PreClass is *orthogonal* and remains effective for steering within or across MIG slices when enabled.

Evaluation. On Rodinia, the classifier achieves 91% accuracy with profiling features and 82% using only pre-runtime features (5-fold CV). In a two-week campaign on a 4-GPU testbed running $\sim$1,200 mixed DL/HPC jobs (training and inference, plus HPC kernels) with extensive co-runs, PreClass reduces mean job completion time by 13% (95% CI: $\pm 1.8\%$, $p < 0.001$), increases GPU utilization by 14%, raises the safe co-location rate from 22% to 68% (slowdown budget $\tau = 15\%$, no OOM/crash), and eliminates *self-induced* OOMs on intercepted application allocation paths (vs. 11% baseline). These gains match dynamic feedback systems such as AntMan (OSDI'20) while requiring no online DCGM/Nsight, no migration, and no privileged access.

Limitations include unified-memory oversubscription, noisy static intensity for irregular kernels, lack of visibility into fused/multi-kernel graphs, and metadata drift; full robustness benefits from lightweight framework-level introspection.

Keywords: GPU scheduling; static workload classification; interference mitigation; memory isolation; `LD_PRELOAD`; CUDA memory pools; container orchestration; VRAM quotas.