Title: Scaling Up GPU Memory Management

Speaker: Dr. Pratheek B. (Nvidia)

Date and Time: 11th August, 2026 at 4 PM

Venue: RM 101 (room 101 of Rajeev Motwani building)

Abstract:

The volume of data generated worldwide is growing at an unprecedented rate, and GPUs have emerged as the primary compute engine for processing this data. While GPUs offer massive compute power — reaching thousands of TFLOPS — they are constrained by the relatively low memory bandwidth (around a few TB/s) and limited memory capacity (in tens of GBs). As a result, memory is often the primary performance bottleneck in many GPU applications.

In this talk, we explore two key aspects of GPU memory management: address translation and memory oversubscription. Efficient address translation is important in GPUs, as it lies in the critical path of memory accesses, and thus impacts overall GPU performance. Memory oversubscription enables GPU programs to work on datasets larger than the on-board GPU memory, but can lead to severe slowdowns.

In the first part of this talk, we focus on the impact of non-uniformity of Multi-Chip-Module (MCM) design on address translation in GPUs. MCM design allows GPUs to scale beyond limits of single monolithic chips by integrating multiple ‘chiplets' with high bandwidth interconnects. Unfortunately, MCM design leads to non-uniform resource access latencies. We analyzed the impact of MCM design on address translation and found large latency increases due to the disaggregated nature of TLBs and page walkers in MCM GPUs which leads to significant performance degradation. We proposed MCM-aware GPU Virtual Memory (MGVM), which leverages the access patterns of GPU applications to limit remote TLB and page table accesses, ultimately improving performance by 52% across diverse applications.

The second part of this talk looks at challenges posed by GPU memory oversubscription. GPU memory oversubscription enables GPU applications to work with datasets larger than the GPU memory capacity, using the CPU memory as swap space. Unfortunately, applications under GPU memory oversubscription often experience significant slowdowns. Our work, ObservUVM, improves GPU's eviction and prefetching policies by enabling observability into GPU’s memory accesses to pages resident on GPU memory — something current GPUs lack. We show that current eviction and prefetching policies, handled by the driver running on the CPU, are limited in their ability to make informed decisions due to the lack of observability. ObservUVM

enables observability into GPU's memory accesses by repurposing existing hardware access counters, enabling better-informed eviction and prefetching policies. ObservUVM improves UVM performance by around 33% across 14 applications.

Speaker Bio:

Pratheek is a Senior Software Engineer at NVIDIA, working on improving GPU memory management. He obtained his Ph.D. from the Indian Institute of Science, Bengaluru in 2026. His research revolves around improving GPU memory management, focusing on address translation and data placement for large-memory workloads. His work spans the domains of GPU micro-architecture, compiler techniques, and operating systems. Previously, he had worked on reverse-engineering Nvidia GPUs.