**GPUScheduler**

*User-Level Preemptive Scheduling for NVIDIA GPUs*

Shaleen Garg, Kishore Kothapalli, Suresh Purini

CSTAR, International Institute of Information Technology, Hyderabad

---

**Matrix Transpose**

**Matrix Mul**

**Program P1**

**Matrix Multiplication (P0)**

**Our Approach**

We fulfill the above traits of a good scheduler by using the following technique:

- We break the kernel into smaller micro-kernels to facilitate preemption.
- Our State save policy involves saving one dim3 variable, hence very low overheads.
- The scheduling framework can employ different scheduling policies in a plug and play fashion.

**Saving the State**

Consider a GPUScheduler compliant program running. Its state needs to be saved in order to resume computations when it is context switched back at a later stage.

**Traits of a Good Scheduler**

- **Preemptive** To reduce wait time of a program waiting in the queue.
- **Low Overheads** To reduce scheduling overheads so as to reduce response time.
- **Flexibility** Ability to support different scheduling policies to cater to different scheduling needs and Service Level Agreement (SLA).

**Current GPU Model**

When Multiple Programs come into picture, the model looks like so.

**Limitations of the Current Model**

Assume arrival of two programs in the following order:

- P0 (large kernel): Matrix Multiplication program on $2^{13} \times 2^{13}$ sized matrices (~ 3 seconds).
- P1 (small kernel): Matrix Transpose program on $2^{13} \times 2^{13}$ sized matrices (~ 3 milliseconds).

The following is what happens when program P0 arrives before program P1.

**Example**

Here is an example to show conversion of a native GPU program to a GPUScheduler compliant GPU program.

**Experimental Results**

Overheads ratio when Matrix Multiplication program is run with and without using GPUScheduler.

**Figure 1:** GPU Working Model

**Figure 2:** Program P0 arrives before Program P1

**Figure 3:** User Program State Diagram

**Figure 4:** Example Round Robin

**Figure 5:** Example Round Robin

**Figure 6:** Overheads for Matrix Multiplication

**Figure 7:** State Diagram for Matrix Multiplication and Matrix Transpose

---

For further queries, contact shaleen.garg@research.iiit.ac.in