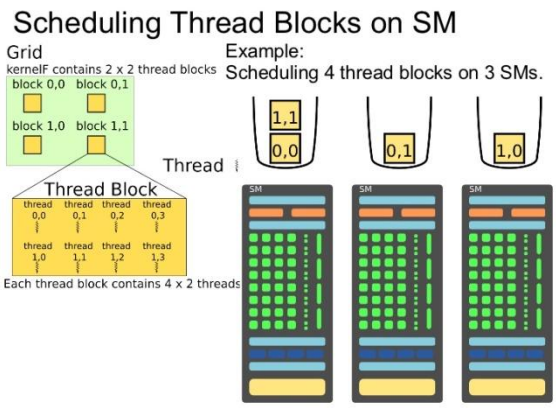
AES Embedded Systems Architecture

Master Thesis Project

Technische Universität Berlin

Locality-Aware Thread Scheduling for GPUs

Background and Motivation: GPUs are high throughput devices and use large number of threads to hide the long latency of memory operations. Threads are grouped into thread blocks, also known as cooperative thread arrays (CTA). All the threads in a given CTA are executed on the same streaming multiprocessor (SM) usually in groups of 32 threads. These groups of threads are called warps. A GPU warp scheduler chooses a new warp from a pool of ready warps when the current warp is waiting for data from memory. Switching warps help to keep the cores busy by hiding long latency of memory operations. In addition to this, the inclusion of caches on GPUs can help to reduce the latency of memory operations and act as a bandwidth filter, provided there is locality in the access stream. We need to understand the locality in GPUs in order to exploit it. Locality occurring from data that is referenced and re-referenced by the same warp is classified as intra-warp locality, while the locality resulting from data that is initially referenced by one warp and re-referenced by another is classified as inter-warp locality. Similarly, we can define the locality as intra-CTA and inter-CTA.



Src::://www.slideshare.net/ugurcandan/gpu-and-the-brick-wall

When a kernel starts execution on a GPU, a CTA scheduler is responsible for assigning CTAs to the SM. In the current GPU architectures, the CTAs assignment is done in round robin (RR) manner. The RR policy is simple to implement, however, it has the disadvantage that it may not exploit inter-CTA locality at L1 cache. This is indicated by the experiments which show very low inter-CTA locality at L1 cache, while the inter-CTA locality is much higher at L2 cache.

Project Goal: CTA and warp scheduling plays an important role for exploiting locality and achieving higher performance. The goal of the project is to study inter-CTA locality in GPU applications and based on the gathered knowledge implement locality-aware CTA scheduling for GPUs. While implementing a locality-aware CTA scheduler, we may also need to tune the warp scheduler to optimize the performance further.

Desired Skills:

- Excellent C/C++
- Strong knowledge of computer architecture
- Preferably knowledge of GPU architecture and CUDA

Contact Person:

Sohan Lal (sohan.lal@tu-berlin.de)

References:

- Adwait Jog et al., “OWL: Cooperative Thread Array Aware Scheduling Techniques for Improving GPGPU Performance”, ASPLOS , 2013