Locality-Aware Loop Scheduling Auto-tuning for GPUs

Rengan Xu, Xiaonan Tian, Sunita Chandrasekaran and Barbara Chapman
Department of Computer Science, University of Houston

Background

In this poster, we present a locality-aware auto-tuning strategy for loop scheduling on GPGPUs. The goal is to find an optimal loop schedule which maps a loop nest to GPU threads hierarchy.

- The compiler generates several kernels with different loop schedules, and the runtime picks the kernel that has the optimal loop schedule.
- The decision about which kernel should be chosen is based on the memory access cost. In the memory access cost model, the number of global memory loads is impacted by the memory coalescing level which is the indicator of spatial locality. The temporal locality is measured in the GPU L1 and L2 cache hit/miss rate which are modeled by reuse distance theory.
- Several micro-benchmarks with different types of data reuse are evaluated.

GPU Threads Hierarchy

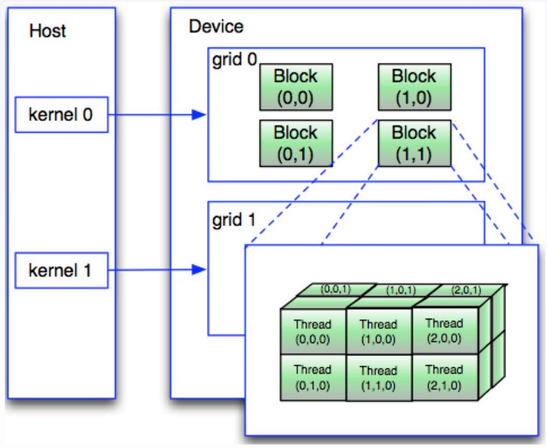


Figure 1: GPU Threads Hierarchy

Auto-tuning of Loop Scheduling

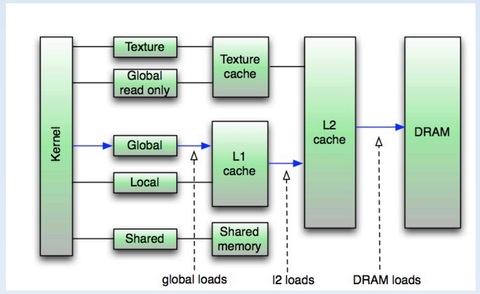


Figure 2: GPU memory hierarchy

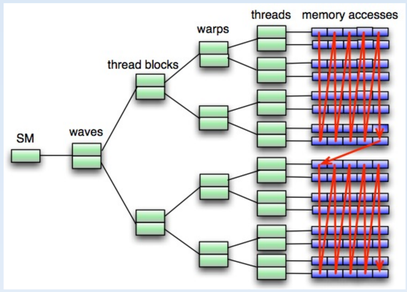


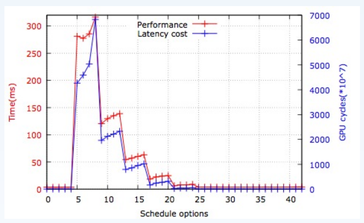
Figure 3: Memory access pattern in the model

$$Cost_{mem} = Mem_{L1} + Mem_{L2} + Mem_{DRAM}$$
$$Mem_{L1} = global\_loads * (1 - L1\_miss\_rate) * L1\_latency$$
$$Mem_{L2} = global\_loads * L1\_miss\_rate * 4 * L2\_latency$$
$$Mem_{DRAM} = global\_loads * L1\_miss\_rate * L2\_miss\_rate * 4 * DRAM\_latency$$

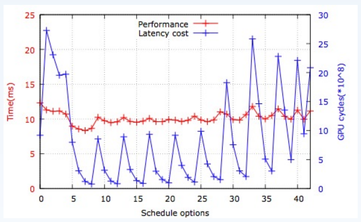
Figure 4: Memory access cost model

Table 1: Reuse distance example. Assume cache line has 16 bytes and cache size is 32 bytes. The reuse distance is based on cache line granularity.

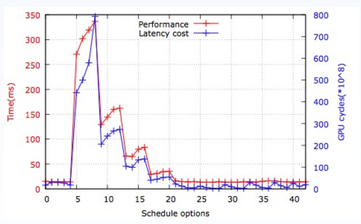
Address	0	8	16	96	8	16	17	104
Cache line	0	0	1	6	0	1	1	6
Reuse distance	INF	0	INF	INF	2	2	0	2
Cache status	Miss	Hit	Miss	Miss	Miss	Miss	Hit	Miss



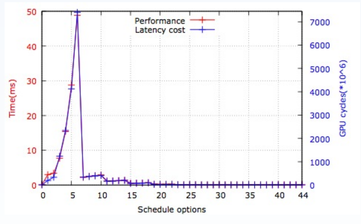
(a) inner data reuse



(b) outer data reuse



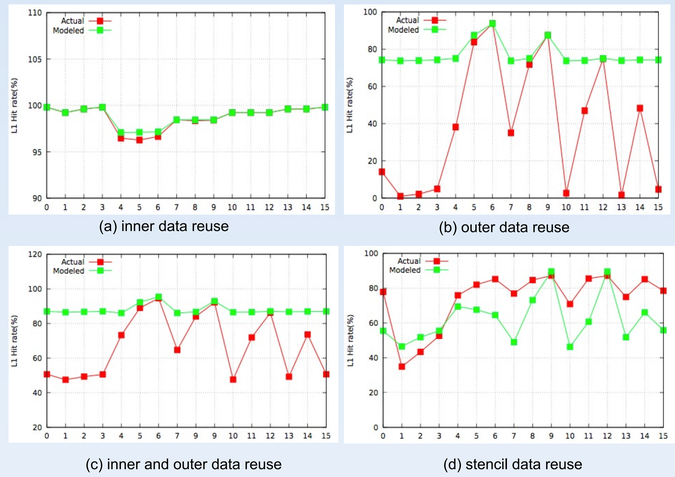
(c) inner and outer data reuse



(d) stencil data reuse

Figure 5: The correlation between performance and memory access cost

Preliminary Results



(a) inner data reuse

(b) outer data reuse

(c) inner and outer data reuse

(d) stencil data reuse

Figure 6: Comparison between modeled and actual L1 cache hit rate

Conclusions

- The kernel performance and the memory access cost are positively correlated, therefore the proposed model is reasonable to choose the optimal loop schedule.
- The modeled and actual L1 cache hit rate are similar but the model accuracy still needs to be improved.
- The L2 cache hit rate and the evaluation of the whole model are in progress.

References

George Almasi, Calin Cascaval and David A. Padua. Calculating Stack Distances Efficiently, ACM SIGPLAN Notices, Vol. 38, No. 2, pp. 37-43, 2002

Cedric Nugteren, Gert-Jan van den Braak, Henk Corporaal and Henri Bal. A Detailed GPU Cache Model Based on Reuse Distance Theory, in HPCA 2014, pages 37-48, 2014