Memory-Aware Scheduling Of Tasks Sharing Data On Multiple GPUs
Maxime Gonthier, Loris Marchal, Samuel Thibault

To cite this version:
Maxime Gonthier, Loris Marchal, Samuel Thibault. Memory-Aware Scheduling Of Tasks Sharing Data On Multiple GPUs. ISC 2023 - ISC High Performance 2023, May 2023, Hamburg, Germany. Lecture Notes in Computer Science. hal-04090595

HAL Id: hal-04090595
https://inria.hal.science/hal-04090595
Submitted on 5 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
**Motivation:** Extract peak performance from GPUs

- Scientific application uses a huge quantity of data
- GPUs can calculate quickly but their memory are limited
- The PCI bus becomes a bottleneck when large amounts of data need to be loaded

**Framework and objectives**

- Independent tasks sharing data
- Same size data
- Multiple GPUs with limited memory

**Algorithms from the state of the art**

- Using (hyper-)graph partitioning: hMETIS
  - Hypergraph ➔ Represents a data being shared by multiple tasks
  - hMETIS produces subsets of task that maximize the amount of edges (data shares) inside each subset
  - We add dynamic reordering and task stealing to improve performance
- EAGER: Process tasks in the order in which it receives them ➔ our baseline
- Deque Model Data Aware Ready (DMDAR): Schedules tasks where their completion times are expected to be minimal. Uses a ready strategy to favor tasks whose data has already been loaded into memory.

**A novel strategy: Data-Aware Reactive Task Scheduling (DARTS)**

**Task flow**

1. DARTS fills Planned Tasks
2. Some tasks are popped by Task Buffer and start to be prefetched
3. If the GPU is idling, a task is popped from Task Buffer
4. After an eviction, tasks in Planned Tasks can be removed

**Experimental evaluations on independent tasksset**

- Tesla V100 GPUs with a 12000 MB/s PCI bandwidth
- GPU memory limited to 500 MB to better distinguish the performance even on small datasets
- 10 runs per experiment (deviation < 2%)

**Summary and conclusion**

- DARTS+LUF stays close to the asymptotic goal!
- Even without memory limitation, DARTS outperforms DMDAR
- DARTS+LUF outperforms DMDAR
- DARTS+LUF stays close to the asymptotic goal!

- DARTS+LUF stays close to the asymptotic goal!
- Even without memory limitation, DARTS outperforms DMDAR
- DARTS+LUF outperforms DMDAR
- DARTS+LUF stays close to the asymptotic goal!