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

To cite this version:
Maxime Gonthier, Samuel Thibault, Loris Marchal. Memory-Aware Scheduling Of Tasks Sharing Data On Multiple GPUs. HiPEAC ACACES 2022 - 18th International Summer School on Advanced Computer Architecture and Compilation for High-performance Embedded Systems, Jul 2022, Fiuggi, Italy. hal-04090607

HAL Id: hal-04090607
https://inria.hal.science/hal-04090607
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 area is limited
- The PCI bus becomes a bottleneck when large amounts of data need to be loaded

Framework and objectives

Objective 1: Minimize the number of tasks on each GPU
Objective 2: Minimize the amount of data loads on each GPU

Algorithms from the state of the art

- EAGER: Process tasks in the order in which it receives them → our baseline
- Deque Mode Data Aware Ready (DM DAR): 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)

- 2 queues of tasks
  - Tasks in plannedTasks can be removed.
  - Tasks in taskBuffer are being prefetched and can't be removed.
  - Tasks in GPU are being computed.
- Flow of tasks
  - (1). DARTS fills plannedTasks.
  - (2). Some tasks are popped by taskBuffer and start to be prefetched.
  - (3). If the GPU is idling, a task is popped from taskBuffer.

Summary and conclusion

- High-performance computing applications, like weather and climate forecasting, have an increasing demand in computer power.
- GPUs have a large computation power but have a limited memory. This, coupled with the share of a single PCI express bus by multiple GPUs induces a large amount of data transfers which reduces performance → Limiting data movement is crucial to extract the most out of GPUs.
- We provide a new algorithm (DARTS) and a new eviction policy (LUF) focused on data locality.

Experimental settings

- Tesla V100 GPUs
- 12000 MB/s PCI bandwidth
- GPU memory limited to 500 MB to better distinguish the performance of different strategies even on small datasets

Experimental evaluations

- Large scheduling cost for all algorithms
- DARTS+OPTI is a variant that reduces the complexity by stopping the search for the optimal data earlier
- DARTS+LUF stay close to the asymptotic goal!
- DMDAR suffers from a conflict between prefetches and evictions
- DARTS+LUF stay close to the asymptotic goal!