Enabling predictable parallelism in single-GPU systems with persistent CUDA threads

PAOLO BURGIO
UNIVERSITY OF MODENA, ITALY

PAOLO.BURGIO@UNIMORE.IT

GP-GPUs / General Purpose GPUs

- Born for graphics, subsequently General Purposes computation
  - Massively parallel architectures
- Baseline for next-generation of power efficient embedded devices
  - Tremendous Performance/Watt
- Growing interest also for automotive and avionics
  - Still, not adoptable within (real-time) industrial settings

Why not real-time GPUs?

- Complex architecture harnesses analyzability
  - Poor predictability

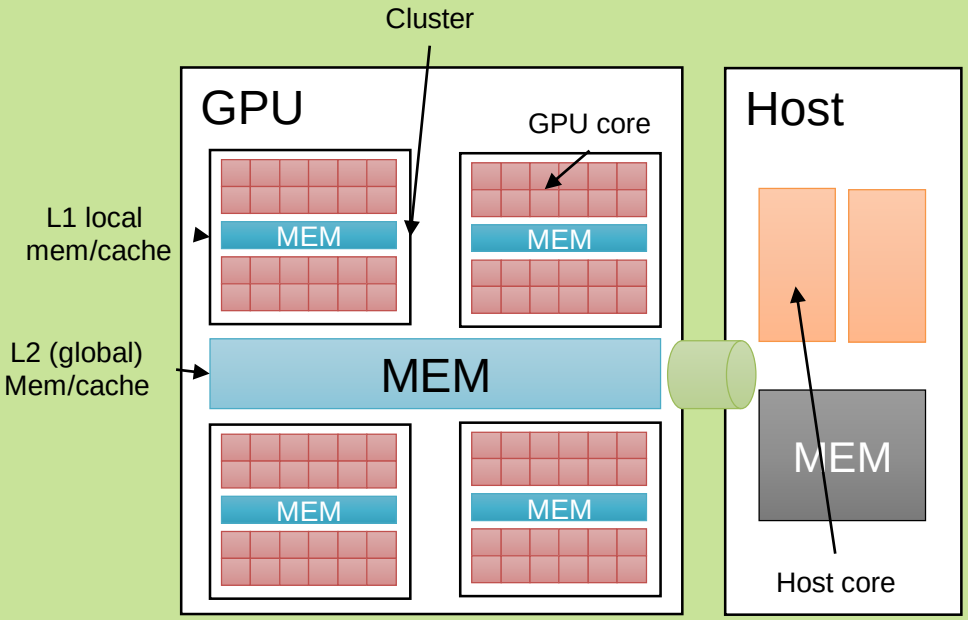
- Non-openness of drivers, firmware..
  - Hard to do research

- Typically, GPU treated a "black box"
  - Atomic shared resource

Hard to extract timing guarantees

LightKer

- Expose GPU architecture at the application level
  - Host-accelerator architecture
  - Clusters of cores
  - Non-Uniform Memory Access (NUMA) system
- Same as modern accelerators
- Pure software approach
  - No additional hardware!



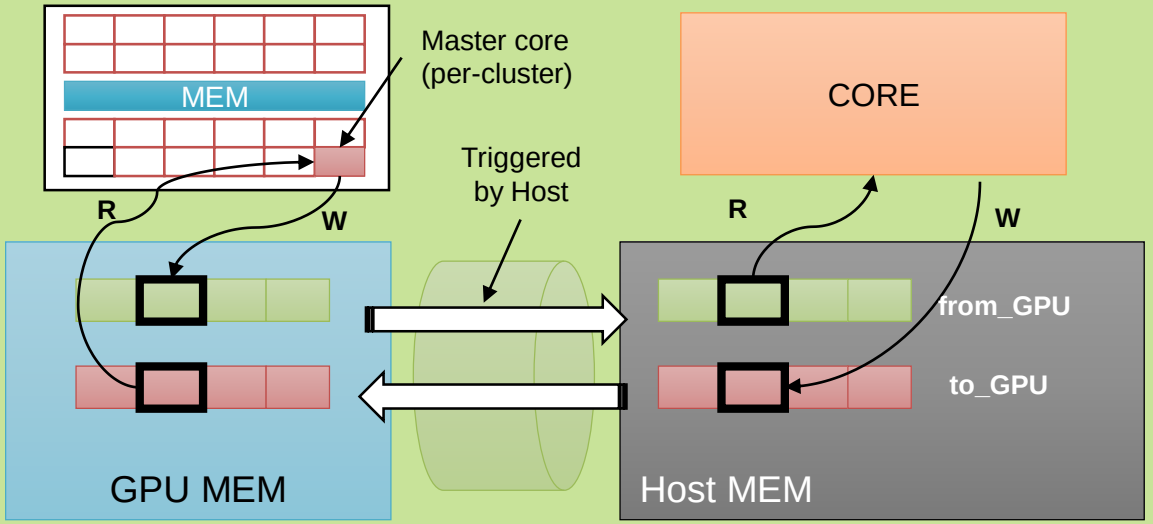
Persistent GPU threads

- Run at user-level
- Pinned to cores
- Continuously spin-wait for work to execute

1 CUDA thread ⇔ 1 GPU core
1 CUDA block ⇔ 1 GPU cluster

Host-to-device communication

- Lock-free mailbox
  - 1 mailbox item for each cluster
- Clusters exposed at the application level
- Master thread for each cluster



LK vs traditional execution model

- LK execution split in
  - Init, { Copyin, Trigger, Wait, Copyout}, Dispose
- "Traditional" GPU kernel
  - { Alloc, Copyin, Launch, Wait, Copyout, Dispose }
- Testbench
  - NVIDIA GTX 980
  - 2048 CUDA cores, 16 clusters

Validation

- Synthetic benchmark
  - Copyin/out not yet considered
  - Trigger phase 1000x faster ☺
  - Synch/Wait is comparable

Single SM			
LK Init	LK Trigger	LK Wait	LK Dispose
509M	**239**	**190k**	30M
CUDA Alloc	CUDA Spawn	CUDA Wait	CUDA Dispose
496M	**3.9k**	**175k**	274k
Full GPU			
LK Init	LK Trigger	LK Wait	LK Dispose
503M	**210**	**190k**	30M
CUDA Alloc	CUDA Spawn	CUDA Wait	CUDA Dispose
497M	**3.8k**	**176k**	247k

Try it!

- LightKernel v0.2
  - Open source
  - http://hipert.mat.unimore.it/LightKer/

- ...and visit our poster ☺

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement: 688860.

HERCULES