# GPU Measurement Breakout

J. Hollingsworth, M. Brim,
J. DelSignore, M. Fagan,
J. Galarowicz, K. Karavanic, X.
Liu,
A. Malony, R. Peri, F. Wolf

# Currently Available Tools

- Nvidia
  - Cupti counters (used by PAPI)
  - Parallel insight (now a free download)
  - CUDA profiler
- Intel
  - Gpa (nsight)
    - Windows and Linux
    - APIs – for device driver
  - North bridge counters
    - Information on data motion
  - Sandybridge
    - PMU for on die GPU
- AMD
  - Code Analyst
- OpenCL
  - Source to source instrumentation

# Who Are The Consumers of Perf Data?

- App developers
  - Rarely, and then often the 1-2 performance experts
  - May be programming high level framework
    - MPI/GPU/etc may be abstracted away
- Library writers/porters/tuners
  - Tend to be more performance experts
- Auto-tuning & other software

# What GPU Performance Info Is Needed?

- CPU-GPU interface
  - When code starts/stops on GPUs
  - Actions associated with data transfer (host<->device)
  - may be enough for understanding overall performance
- GPU internals
  - Necessary for optimizations of kernels
  - Understand interaction/overlap of kernels on device
  - More precise association of counters to kernel operation
- Observations
  - First need:
    - Good CPU tools to find the slow spots in the code
    - Mental model of what GPUs can do (well)
  - What is correct level of abstraction?
    - Thread, device, warp, …

# PTX (Nvidia IR)

- Natural Place for GPU Instrumentation
  - Nvidia specific, but cross many generations of GPUs
  - LLVM currently supports reading/writing ptx files
- Some compiler tools generate PTX
  - Ocelot (Georgia Tech)
  - …

# JITs

- Many GPUs use JITs
  - May not even reveal real ISA
- Need To Identify Tool needs to JITs
  - Notification of when JIT happens
  - Ability to Fore re-JIT for instrumentation

# Summary

- Still need good CPU tools
- PTX natural source of instrumentation
- Need API with JIT hooks
- Lack of convergence of programming models makes life hard for tool builders