ECE 574 – Cluster Computing
Lecture 18

Vince Weaver

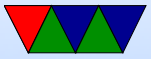
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

2 April 2019

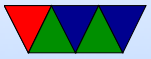
Announcements

- HW#8 was posted



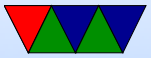
Project Topic Notes

- I responded to everyone's e-mail. If your group didn't get one let me know
- Have a wide variety of machines to run on.
- If interested in power measurement let me know.



CUDA Notes

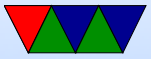
- Nicely, we can use only block/thread for our results, even on biggest files
- In the past had to split 3 ways, grid/block/thread
- In past there was a limit of 64k blocks with "compute version 2" but on "compute version 3" we can have up to 2 billion



Other Accelerator Options

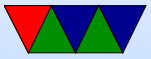
- XeonPhi – came out of the larabee design (effort to do a GPU powered by x86 chips). Large array of x86 chips(p5 class on older models, atom on newer) on PCIe card. Sort of like a plug-in mini cluster. Runs Linux, can ssh into the boards over PCIe. Benefit: can use existing x86 programming tools and knowledge.

- FPGA – can have FPGA accelerator. Only worthwhile if you don't plan to reprogram it much as time delay in reprogramming. Also requires special compiler support



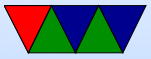
(OpenMP?)

- ASIC – can have hard-coded custom hardware for acceleration. Expensive. Found in BitCoin mining?
- DSPs – can be used as accelerators



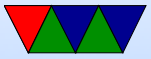
Google TPU

- Tensor Processing Unit
- ISCA paper – In Datacenter Performance Analysis of a Tensor Processing Unit



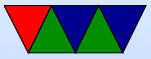
OpenACC

- Sort of like OpenMP but can offload to GPU as well as CPUs
- Cray, CAPS, Nvidia and PGI
- Lots of pragmas

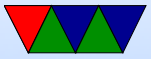


Other Libraries

- Metal – from Apple, their replacement for OpenCL. C++ like, sort of a mix of OpenCL and OpenGL
- Defunct low-level GPU libraries: Glide (3dfx), Mantle (AMD)
- Other low-level GPU libraries: GNM (playstation 4), NVN (Nvidia/Switch)
- WebGPU – GL/GPGPU Javascript (currently under development)
- WebCL – OpenCL Javascript bindings

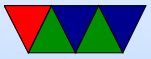


- OpenVG – 2d vector graphics accel



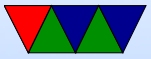
Vulkan

- More modern OpenGL
- Supposedly OpenCL merging into Vulcan?
- based on AMD Mantle

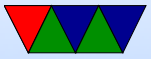


OpenCL

- CUDA is only for NVIDIA
- What if you have Intel or AMD (ATI) chip? Or ARM MALI (sadly not Raspberry Pi)
- OpenCL is sort of like CUDA, but cross-platform
- Not only for GPUs, but can target regular CPU, DSP, FPGAs, etc
- Vendor provides a driver

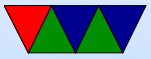


- Khronos (the OpenGL + Vulkan people?) also run OpenCL
- Windows, OSX, Linux



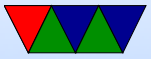
Cluster Computing Power

Can spend a whole class (i.e. ECE571) discussing where power goes in a modern computing system.



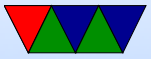
Cluster Computing Power

Why is low-power super-computing important?

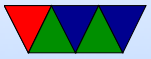


Green500

- Green 500 list
- Push for more accurate power reporting in the Top500 list
- Top 5, Nov 2016
  1. NVIDIA DGX-1 Xeon/Tesla 350kW, 9.462 GFLOPs/W
  2. (#8) Swiss Piz Daint Cray XC50 Xeon/Tesla 1.3MW, 7.453 GFLOPs/W

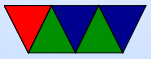


3. Riken ZettaScaler Xeon/PEZY-SCnp
   2 ARM Cores/1024 RISC Cores, 1.5TFLOPs
   150kW, 6.7GFLOPs/W
4. (#1) Sunway TaihuLight, Sunway, 15MW, 6.0GFLOPs/W
5. Fujitsu PRIMERGY , Xeon Phi, 77kW, 5.8GFLOPs/W



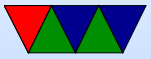
SuperComputer Power

- Cooling
- DVFS
- Power-capping
- Up to 12% spent by the interconnect
  Pi cluster, 90W, 20W or so is the ethernet switch

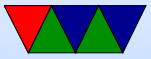


Measuring Power and Energy

- Sense resistor or Hall Effect sensor gives you the current
- Sense resistor is small resistor. Measure voltage drop. Current V=IR Ohm's Law, so V/R=I
- Voltage drops are often small (why?) so you made need to amplify with instrumentation amplifier
- Then you need to measure with A/D converter
- $P = IV$ and you know the voltage
- How to get Energy from Power?

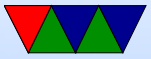


Measuring Power and Energy



Why?

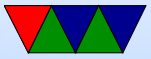
- New, massive, HPC machines use impressive amounts of power
- When you have 100k+ cores, saving a few Joules per core quickly adds up
- To improve power/energy draw, you need some way of measuring it



Energy/Power Measurement is Already Possible

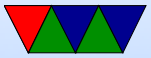
Three common ways of doing this:

- Hand-instrumenting a system by tapping all power inputs to CPU, memory, disk, etc., and using a data logger
- Using a pass-through power meter that you plug your server into. Often these will log over USB
- Estimating power/energy with a software model based on system behavior



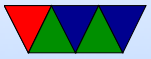
Where does the power go in a system?

- CPU
- DRAM
- GPU
- Disk
- Network
- Cooling/Fan
- Power Supply



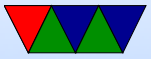
Measuring system wide

- Can use kill-o-watt, WattsUpPro, or similar



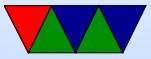
Measuring hardware

- Really hard
- DRAM, PCI, USB, fans: can put sense resistor in line with power supply, measure current and voltage to calculate power
- CPU harder, how do you intercept? There is the P4 line (a 12V special power cable from PSU on recent systems) but it might also power other parts of the system



Estimating Power

- Can construct a model to estimate power
- Inputs like performance counters
  - CPU might be related to instructions/cycle, pipeline stalls, FPU instructions retired, etc
  - DRAM might be related to cache misses, bytes/second
- Hopefully you validate the model



RAPL Power Estimates

- Running-Average Power Limit
- Recent Intel CPUs
- Need to estimate power usage for power-capping and turbo-boost
- Nicely provide the values to userspace
- On most systems (excluding some Haswell models) is an estimate from an on-chip power model based on various inputs (temperature, perf counters, etc)
- Very easy to read, using the perf tool