Accelerated Computing with GPUs, APUs, and FPGAs

Summer 2015
Prof. Karen L. Karavanic
karavan@pdx.edu
http://web.cecs.pdx.edu/~karavan
1: The World's Fastest Computer

- www.top500.org
- Linpack
- Floating Point Operations per Second (FLOP/s)

- June 9, 2008: The Roadrunner breaks the “PetaFLOP barrier” - computer achieves a rate of 1.026 petaFLOP/s
WhattaFLOPS ???

- Floating Point Operations Per Second
- MFLOPS $10^6$
- GFLOPS $10^9$
- TFLOPS $10^{12}$
- PetaFLOPS $- 10^{15}$
  $1,000,000,000,000,000$
- ___FLOPS $- 10^{18}$ coming soon.....
1: The World's Fastest Computer

- www.top500.org
- Linpack
- Floating Point Operatings per Second (FLOP/s)

- June 9, 2008: The Roadrunner computer achieves a rate of 1.026 petaFLOP/s

- Yes, that's $1.026 \times 10^{15}$ floating point operations per second
1: The World's Fastest Computer

• [www.top500.org](http://www.top500.org)
• Linpack
• Floating Point Operatings per Second (FLOP/s)

• June 9, 2008: The Roadrunner computer achieves a rate of 1.026 petaFLOP/s

• Yes, that's 1.026 *quadrillion* floating point operations per second
• How did they do it?
  • 6,948 dual-core AMD Opteron chips
  • 12,960 8-core Cell BE chips
Roadrunner

• 80 terabytes of memory
• housed in 288 refrigerator-sized racks occupying 6,000 square feet
• 57 miles of fiber optic cable
• A complex physics calculation that will take Roadrunner one week to complete, would have taken the fastest computer of 1998 20 years to finish -- it would be half done today!
• If it were possible for cars to improve their gas mileage over the past decade at the same rate that supercomputers have improved their cost and efficiency, we'd be getting 200,000 miles to the gallon today
1. The World’s Fastest Computer

• #2 on Today’s List (Nov 2014): Titan
  • Cray XK7 with Opteron 6274s and NVIDIA K20x
  • Runs Cray Linux
  • 560,640 processor cores
    • including 261,632 NVIDIA K20x graphics processors
  • Theoretical Peak: 27 PetaFLOPs
  • Achieved on LINPACK: 17.59 PetaFLOPs
  • (power 8209 kW)
NVIDIA Kepler (K20)

• Peak double precision: 1.17 TeraFLOPs
• Peak single precision: 3.52 TeraFLOPs
NVIDIA Kepler (K20)

• Peak double precision: 1.17 TeraFLOPs
• Peak single precision: 3.52 TeraFLOPs

• Yes, that’s just for one Graphics card !!
1. The World’s Fastest Computer

- #1 on Today’s List (Nov. 2014): Tianhe-2
  - 3,120,000 cores, 33 PetaFLOPs LINPACK, 17,000 kW
  - Intel Xeon processors with Intel Xeon Phi
  - “MIC” – many integrated core architecture
Intel Phi Coprocessor

Knights Corner Coprocessor

Intel® Xeon® Processor

TCP/IP

PCIe x16

System Memory

KNC Card

GDDR5 Channel

... GDDR5 Channel

GDDR5 Channel

GDDR5 Channel

> 50 Cores

Knights Corner

Linux OS

>= 8GB GDDR5 memory

Intel®
Intel Phi Coprocessor

Knights Corner Micro-architecture
Accelerated Computing

• Heterogeneous Computing
• Use different hardware for different types of computation
• Ex: GPU really bad at sequential computing
• Idea: Combine the general purpose x86 with one or more accelerators
• Advantages: speedup, power savings
• Today’s state of the art to accomplish large-scale computing
• Different types of accelerators
• Our hands-on focus will be CUDA/GPUs
Accelerated Computing Course Structure (Summer 2015)

• Expected background
  • Programming Skills, Knowledge of: Operating Systems, Architecture, Compilers (eg CS333, CS201 or CS341, CS321)
  • CS "Senior" Undergraduates or Post-Bac or M.S. students

• Workload
  • Homeworks (40% of grade)
  • Class participation (20% of grade)
  • Programming Project (40% of grade)

• See course web page for tentative calendar, slides:
  • http://web.cecs.pdx.edu/~karavan/gpu

• We will have 1 or 2 short breaks each class
• We will spend some time in the Linux Lab
Introductions

Please write on yellow sheet:

• 1. Your name
• 2. Your goals for this course (including what grade you want)
• 3. Your experience with C and C++
• 4. Your experience with parallel programming
• 5. Your experience with CUDA
HW #1
(due Tuesday June 30 before class start)

• Compile and run the vector addition example found at:
  • http://legacy.lclark.edu/~jmache/parallel/CUDA/examples/vector_addition.cu
  • Try some different sizes for the arrays

• Determine information about the graphics card available on the linux lab machines
  • Capability?
  • Number of cores?
  • Etc
  • What version of CUDA are you running?

• USEFUL:
  • NVIDIA CUDA C Programming Guide (be sure it’s the right version)
  • addpkg
HW #1

• How to submit: Describe your results from running the vector addition code
• Email your work to: karavan@pdx.edu
• With Subject line: ACCEL HW1
Thanks…

• This talk includes slides, ideas, and examples from: Kathy Yelick (UC Berkeley), Wen-mei Hwu (UIUC), NVIDIA, and Michael Smith (M.S. PSU; now at Cray)