Terascale, Petascale, Exascale and Beyond

Charlie Peck, Daniel Ernst, Andrew Fitz Gibbon and Henry Neeman

Earlham College, Shodor Foundation, University of Wisconsin-Eau Claire and Oklahoma University-OSCER

National Computational Science Institute, TeraGrid

Where are we?

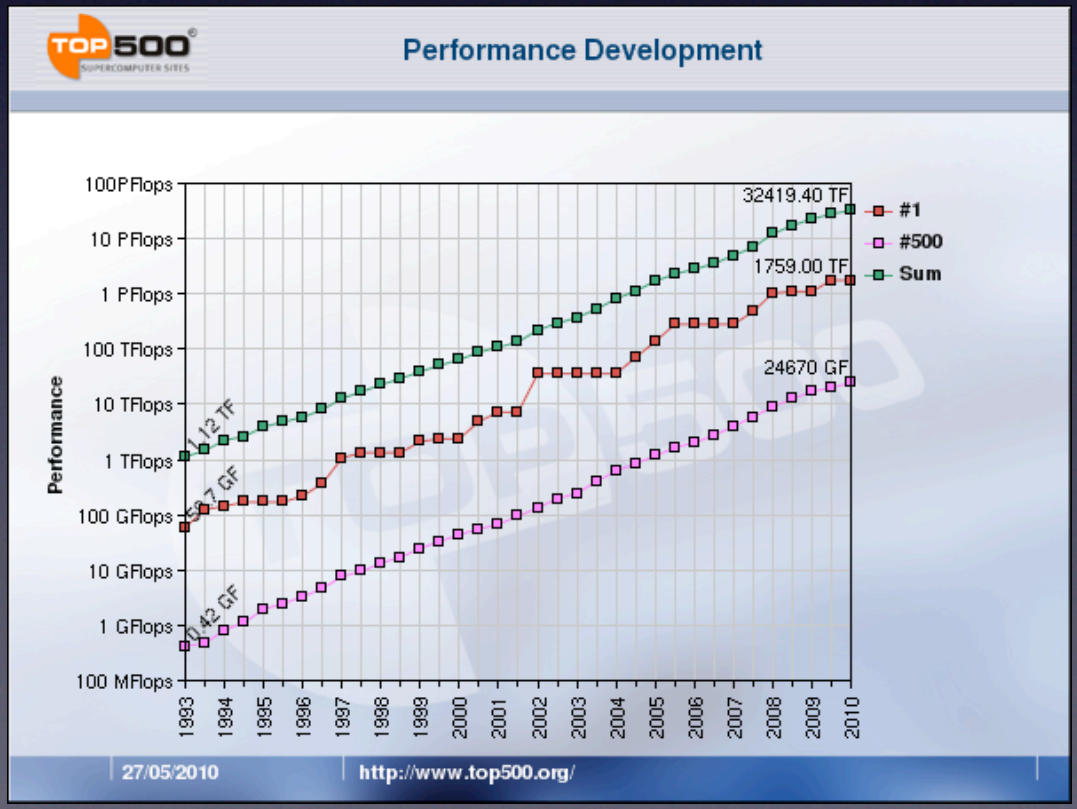
- Giga FLOPS - 10^9 floating point operations per second
- Tera FLOPS - 10^12
- Peta FLOPS - 10^15
- Exa FLOPS - 10^18

- Jaguar - 2.3 Peta FLOPS (peak), 250K CPU cores
- Nebulae - 2.9 Peta FLOPS (peak), 120K CPU cores + GP-GPU cores
- Blue Waters - 10.1 Peta FLOPS (peak, estimated), 2.4M CPU cores

- NVIDIA Tesla C2050 - 1.2 Tera FLOPS
  - ~#1 on the June 1993 Top 500 list
  - ~#500 on the Nov 2003 Top 500 list

Well, how did we get here?

- Clock frequency stalled around 2005
- Advent of GP-GPU hardware and libraries
- Performance increases now provided by core-count growth (CPU and GP-GPU)



What makes this hard?

- Power consumption, heat dissipation
  Currently ~400 MFLOP/Watt on HPL or ~2.5 MWatt per PetaFLOP, goal is factor of 100 improvement
- Reliability
  More parts, some of which are less reliable, makes for a lower MTBF, checkpointing doesn't scale well
- Programming
  Ability to exploit all the available parallelism
  Checkpointing does not scale well
  Mapping algorithms to the underlying architectures efficiently

Into the Blue again...

- Blue Waters - NCSA, IBM, and the Great Lakes Consortium for Petascale Computing
- IBM POWER7 CPUs (8 cores)
- Water cooled chassis
- Multi-level memory hierarchy (we've seen this before)
- Multi-level network hierarchy (this is fairly new to us)

Well, how might we get there?

- Hybrid programming models
  Message passing + shared memory
  Message passing + GP-GPU
- Capability vs capacity computing
  What work could be done effectively with a capacity approach rather than a capability approach?
- Improved software techniques
  Algorithms that scale more efficiently, e.g. strong vs weak scaling
  Tools which facilitate scaling the software engineering processes
  Improved memory utilization, cost of operations vs fetches
  Overlap computation and communication
  Improvements to load balancing algorithms
- Improved software resiliency
  Replacements for the checkpoint approach

Questions?