Terascale, Petascale, Exascale and Beyond

Charlie Peck and Tom Murphy
Earlham College and Contra Costa College

High Performance Computing Modernization Program
Summer 2011 Workshop on
Intermediate Parallel Programming & Cluster Computing
produced in conjunction with the
National Computational Science Institute and the
SC11 Conference,
co-hosted at PUPR and OU

Where are we?

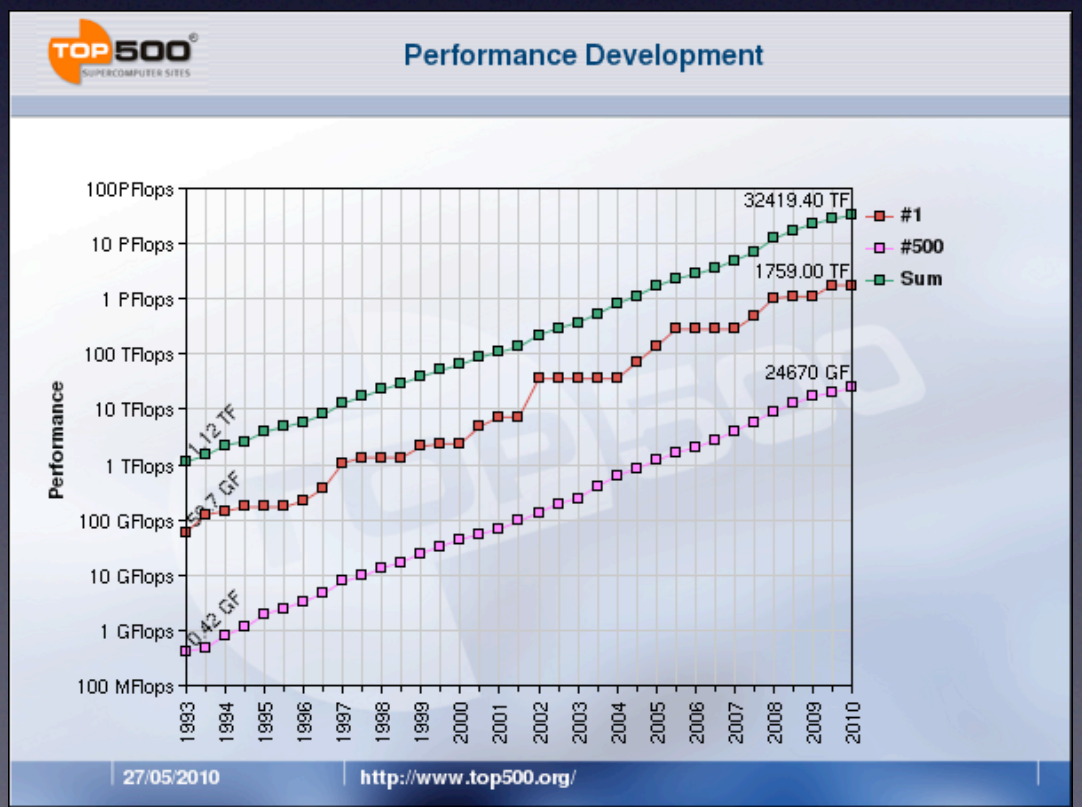
- Giga FLOPS - 10^9 (floating point operations per second)
- Tera FLOPS - 10^12
- Peta FLOPS - 10^15
- Exa FLOPS - 10^18
- Zeta FLOPS - 10^21

- 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), 300K CPU cores

- NVIDIA Tesla C2050 card - 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)



Where do we want to go?

- Grand challenge problems
  Whole body blood circulation
  Systems biology, e.g. mapping the human proteome
  Turing test
- Weather forecasting and climate modeling
  Predicting severe weather events
  Calculating localized sea level rise
- Data mining
  Roughly 1ZB of stored digital data in 2009, estimated 35ZB in 2020 (IDC)
  Medical image scan at ~30GB each (was ~4GB each in 2005)
  Large Hadron Collider produces ~1.5GB/second of operation

Same as it ever was, only more so

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

Into the Blue again...

- Blue Waters - NCSA, IBM, and the Great Lakes Consortium for Petascale Computing
- IBM POWER7 CPUs (8 cores, 12 execution units each)
- 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
  Improvements to and replacements for the checkpointing approach

Questions?

With thanks and apologies to David Byrne and the Talking Heads...