Teaching Parallel Computing to Science Faculty

Best practices and common pitfalls

David A. Joiner, Kean University
Paul Gray, University of Northern Iowa
Thomas Murphy, Contra Costa Community College
Charles Peck, Earlham College

Motivation from HPC

- “Computational science is now indispensable to the solution of complex problems in every sector, from traditional science and engineering domains to such key areas as national security, public health, and economic innovation…Yet, despite the great opportunities and needs, universities and the Federal government have not effectively recognized the strategic significance of computational science in either their organizational structures or their research and educational planning.”
  - June 2005 PITAC report
- There is currently a shortfall in both the curriculum and resources available for teaching supercomuting.

Motivation from Multi-Core

- Dual core now
- 64 cores in 5 years?

  – As desktop and laptop platforms move to parallel architectures, future CS professionals will need to train/retrain to design to the new platforms.

Overview of Workshops

- Findings from action research studies over 16 workshops since 2006
  - National Computational Science Institute (NCSI) The Minority Serving Institutions—High Performance Computing (MSI-HPC) program of the National Computational Science Alliance
  - The Consortium for Computing Sciences in Colleges (CCSC)
  - The Center for Excellence in High Performance Computing
  - The Oklahoma University Supercomputing Symposium series
  - The Super Computing (SC) conference series education program

What We Did

- Delivered Curriculum
  - Week long workshop in parallel computing
  - Project based & hands on
- Gathered Information
  - Short daily feedback surveys
  - Pre- and Post- surveys still under analysis
  - Modification of materials from one workshop to the next
  - Cataloguing of results of both beneficial and harmful mistakes

The Workshop Curriculum

- 1/3 Pedagogy
  - Key concepts
  - Application to sciences
- 1/3 Protocols
  - MPI Programming
- 1/3 Architecture
  - OSCAR
  - BCCD

Pedagogy

- Teaching computers without computers
- “New Classic” examples
  - N-Body
  - Traveling Salesman
  - Forest Fire
  - Life
  - Parameter Space studies
- Speedup

Showing Speedup

- People get the concept of why you want to use a parallel computer better if they see examples that actually speed up first.
  - Real simulations
  - Real machines
- Examples that don't speed up (at least for introducting concepts) don't have the same impact
  - "Hello World"
  - Simple calculations which run in little time
  - "Simulated" clusters—multiple processes on a single CPU.

Real Time, Visual Results

- Visual Learners
  - “See”ing speedup means that the results need to “look” faster
  - stdout is better than “time”
    - Provided that you don’t have too much output
  - X is better than stdout
    - Provided you don’t saturate your network or create a synchronization point in your code

Engaging Lessons

- Interactive
  - Multiple runs with changeable input
  - Simulations that respond to input
    - NAMD and IMD
- Active
  - Creation and modification of code
- Relevant
  - Simulation and Modeling
  - Realistic algorithms

Performance Analysis

- Authenticity
  - Scaling doesn’t always happen as one would expect
- Reproducibilty
  - Teaching scaling should happen as one would expect
- Accesibility
  - Teaching scaling requires a machine that scales

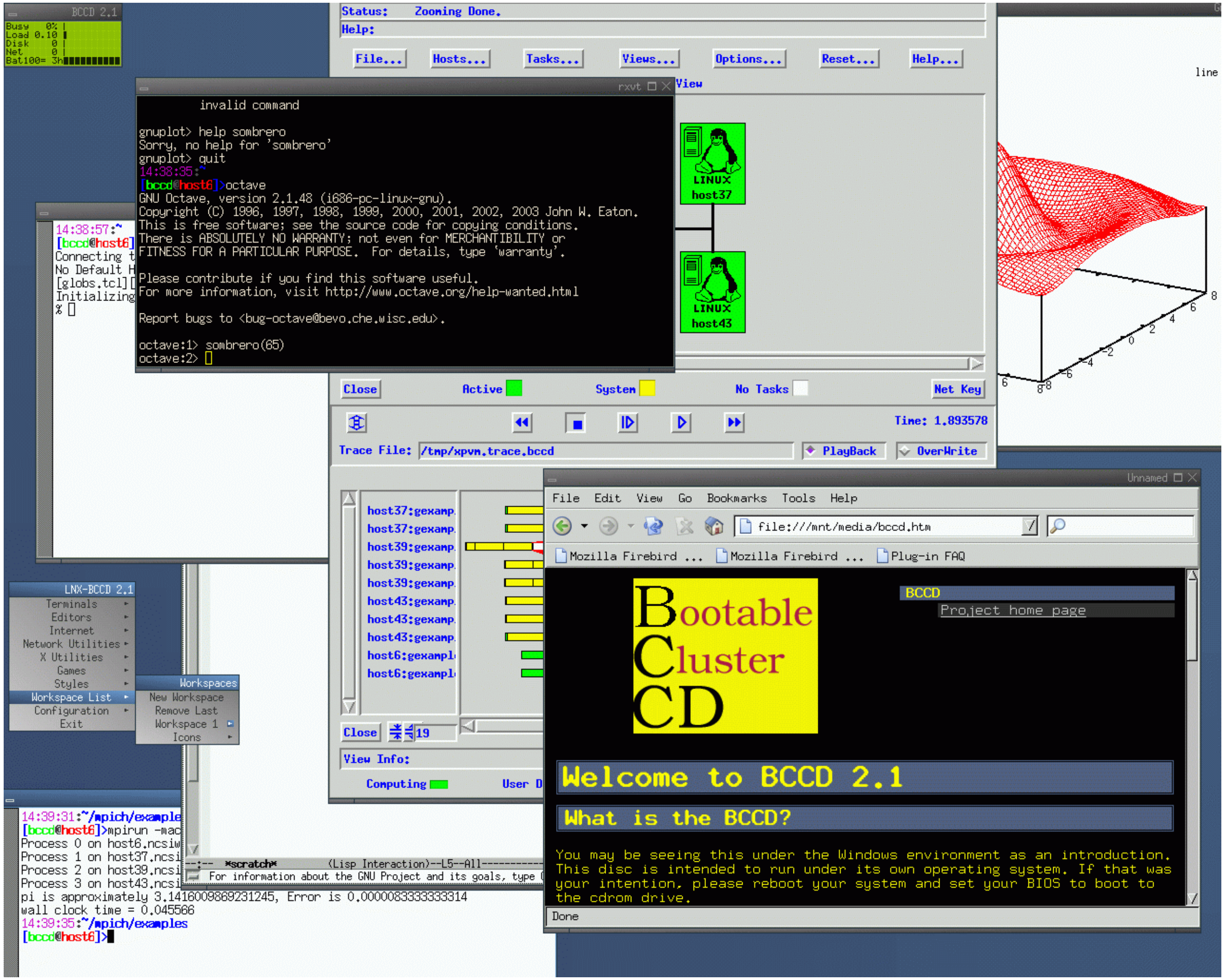
Pedagogical Conclusions

- Teaching HPC can be motivated by
  - Modeling and simulation, tied with visualization, of real science applications.
  - Relation of cluster computing to real demands of operating system design, computer design, and system administration.
- Curriculum and Infrastructure solutions have been tested
  - Example codes available at CSERD
  - Additional lessons using community codes being designed
  - BCCD

Infrastructure Issues

- How do you give faculty and students a consistent, uniform, scalable, accesible platform for learning parallel computing?
  - Turn campus PC labs into diskless clusters
    - The BCCD
  - Provide low cost, low power clustering options as student projects
    - Little Fe

BCCD - http://bccd.cs.uni.edu/



BCCD - http://bccd.cs.uni.edu/

- Diskless cluster solution
  - Uses DNS, DHCP to setup nodes
  - Automated scripts allow users to “opt-in” to clustering activities
  - Many major open source clustering tools ready to go
    - MPICH, LAM-MPI, PVM
    - XMPI, UPSHOT
    - OpenMosix
  - Classroom activities pre-installed, more available in future as downloadable packages

Little-Fe – http://cs.earlham.edu/little-fe/

- Portable low cost cluster design for classroom instruction
  - 4-8 “brick” nodes, network booted from a single disk image on a single power supply
  - Mounted into a lightweight open air rack
  - Under 50 lbs and fits into standard Pelican case for airline travel
  - Early versions used Debian and OSCAR tools, new versions designed to use BCCD
  - Total cost ~$2K

Acknowledgments

This work funded by National Science Foundation awards DUE-0435187 (CSERD – NSDL Project) and DUE-0127488 (NCSI – CCLI-ND Project).