Using Remote HPC Resources to Teach Local Courses

Oklahoma Supercomputing Symposium
10/06/10

Larry F. Sells, Oklahoma City University
Clay B. Carley III, East Central University
Chao (Charlie) Zhao, Cameron University
The Impact of OSCER on Software Engineering at Oklahoma City University

Larry F. Sells
Department of Computer Science
Oklahoma City University
Software Engineering at OCU

- Same instructor for many years
- SE concepts and team/project course
- The last 4 semesters project focus has been on MPI and OpenMP
Instructor’s Training

- OU Supercomputing Center for Education and Research (OSCER) resources
- Importance of NCSI summer 2010 Intermediate Parallel Computing workshop in pulling many things together
Course Objectives

- Engage students in a first course in software engineering (Roger Pressman text)
- Help students work in a UNIX, C, MPI environment
- Help student teams create MPI project code along with SE documentation (requirements, design, test plan, user manual, final source code, executables, and report)
Software Engineering Fall 2010 - Prerequisites

- 2 years experience in C, C++, or Java
- Knowledge of data structures
- Basic background in Linux (UNIX) helpful
- No previous study of parallel programming, HPC, or MPI
- No previous knowledge of cryptology
Software Engineering Fall Project

- Inspired by Simon Singh’s “Cipher Challenge”
- Ciphers include: homophonic, Vigenere, Playfair, ADFGVX, DES, and RSA
- Goal is to decipher Singh’s ciphers using MPI and C or C++ and to develop appropriate SE documentation
Dr. Henry Neeman, OSCER, and Sooner

- OSCER operations team created Sooner accounts for SE students
- Henry Neeman and Josh Alexander came to OCU to do an introduction to Sooner lab
- Importance of Neeman’s 11 SiPE (Supercomputing in Plain English) presentations – especially #5 and #6
- We are working to set up an OU Sooner tour – gives gut understanding of a cluster.
Relevant OSCER 2010 Workshop Ideas

- Client/server, data parallelism, task parallelism, and pipeline parallel strategies
- Comparing MPI output on Sooner and Earlham cluster
- MPI debugging
- Introduction to CUDA
- Introduction to hybrid HPC – CUDA and MPI
Sooner is Better

Clay B. Carley III
Department of Computer Science
East Central University
Parallel Programming
The Future

- High Performance Computing
- The Cloud
- Multicore Architectures

Equals => more pressure on future graduates to understand parallel programming
Parallel Programming Spring 2010 - Prerequisites

- Experience in C
- Linux (UNIX) experience
- No previous study of parallel programming, HPC (High Performance Computing) or MPI (Message Passing Interface) before
- No experience with batch processing
Course Objectives

- Engage students in a first course in parallel programming ("Parallel Programming with MPI," Pacheco)
- Help students work in a C, MPI, batch environment
- Help students understand the different parallel computing architectures
OSCER Resources

- *Priming the Pump* with Dr. Henry Neeman's “Supercomputing in Plain English” slides
- Hardware and Memory Issues
- Workshop links
- MPI – examples are available in C and FORTRAN
**Priming the Pump**

- Starting from scratch
- Getting an Instructor's account on Sooner
- Online Resources
  - Workshops
  - PowerPoint Slides
  - Exercises and Code Examples
Course Kickoff

Visit by Dr. Neeman and Josh Alexander
- Connecting to Sooner for the first time
- Guidance for first exercises
- Q&A about supercomputing and supercomputers
The Sooner Linux Cluster

- 1,072 Intel Xeon CPU chips/\textbf{4288 cores}
- 8,768 GB RAM
- \(~\text{105 TB globally accessible disk}\)
- QLogic Infiniband
- Force10 Networks Gigabit Ethernet
- Red Hat Enterprise Linux 5
- Peak speed: 34.45 TFLOP$^*$

*TFLOP$^*$: trillion calculations per second

sooner.oscer.ou.edu
Sooner Benefits for Course

- “real supercomputer environment”
- Ability to explore different options to see what impact they have on performance
  - Increasing/decreasing number of cores
  - Increasing/decreasing number of processes
  - Increasing/decreasing granularity of the problem
Methods for Teaching Some Basic Concepts of Parallel Computing to Undergraduate CS Students at Cameron University

Chao Zhao
Associate Professor
Computing and Technology Department
Cameron University
Parallel Computing at Cameron

- Cameron is a five year regional public university.
- BS in Computer Science is offered in Computing and Technology Department.
- CS 3813 Parallel Computing is a required course in CS curriculum (ACM 2000).
- MPI is used as message passing library.
- OSCER has been used as significant teaching resources.
Instructor’s Training and Cooperation with OSCER

- OU HPC Summer Workshops (06, 07, 08, 09)
- Inviting supercomputing expert to deliver speech to students (Dr. Neeman: Basic Parallel concepts and Logics)
- Visiting OSCER Supercomputing Center
- Using OSCER’s supercomputer to run students’ parallel programs:
  - Dr. Neeman and Josh Alexander campus visits
  - Sooner account for each student
Why Parallel Computing?

- Take advantage of multiple core machines
- Parallel approach may improve computing efficiency:
  - \( S_{p(n)} = \frac{T_s}{T_p} \)
  - \( E_p = \frac{T_s}{T_p \cdot n} \) or
  - \( E_p = \frac{S_p}{n} \)
- Solve some problems that CANNOT be solved by sequential approach
- No speed limit in theory
Parallel Program Logical Structure

Main function
{
    common part (variables declaration and initialization);
    if ( myrank equal master)
    {
        code that will be executed by the master process;
    }
    else {
        code that will be executed by slave processes;
    }
    program termination part;
}
Teaching Methods

- Job Balance (Matrix Multiplication)
  \[ A (m, n) \times B (m', n') = C (n, m') \]

**Master process does the following in order:**
- Broadcasting matrix B to all slave processes;
- Sending a row of matrix A to each process.
- Receiving a row of matrix C from a slave process.
- Copying the received row into matrix C
- If the number of sent rows is less than the number of rows in matrix A, send a row to an idle process that completed its task. Repeat C, D, and E until the job is done.

**A Slave process does:**
- Receiving matrix B;
- Receiving a row r of matrix A;
- Multiplying row r to matrix B to produce a row of matrix C
- Sending the resulted row back to the master process
- Repeating B, C, and D until the completion notice is received.
Teaching Methods (continued)

- Communicator Creation
- Monte Carlo method to compute $\pi$
- Master process generates a set of random number repeatedly until it is noticed to terminated.
- Slave processes use the random numbers to generate points.
- Master process and slave process belong to different communicators.
Conclusions

- Instructor training is essential to offer a sound teaching to our students in parallel computing and Software Engineering.
- OSCER is a very useful resource that can be used to improve teaching and learning quality.
- Proper teaching methods provide instructors with an efficient way to deliver their teaching materials.
- HPC has much to offer to the CS curriculum.
- Thanks to OSCER and its excellent staff!