TLP Warmup

April 29, 2009

Background

- Suppose your company wants to purchase some new computers.
- There are two types of computers that you can choose:
  - Multiple single core systems (that will be clustered)
  - Fewer expensive quad-core systems
- Your budged is limited.
- Considering the budge and performance tradeoff, your technical manager needs to make a purchase decision with your suggestions about the best type of computers to purchase.

Introduction

- Two similar techniques.
  - Pthreads
    - A program will be split up into threads, which share memory on a single system.
  - MPI protocol
    - A program will be split up across a cluster of systems.
- Performance trade off
  - A fast quad-core system vs. a cluster of cheaper single core systems.

Useful tools

- Tool Chains
  - pthreads howto
  - MPI_Documentation
  - General LAM MPI documentation
- Sample codes
  - dotprod.c.pthreads
  - dotprod.c.mpi

pthread

- Make sure and get code working on ECE machines
- Test code with various thread configurations
  - ECE single core machines
  - Tetra quad-core machine

Thread (1)

- Threads in the same process share:
  - Process instructions
  - Most data
  - open files (descriptors)
  - signals and signal handlers
  - current working directory
  - User and group id

Thread (2)

- Each thread has a unique:
  - Thread ID
  - set of registers, stack pointer
  - stack for local variables, return addresses
  - signal mask
  - priority
  - Return value

pthread-step by step (1)

- First logon to one of the ECE unix systems.
- Record the processor speed information
  - *$ cat /proc/cpuinfo*
- Modify the pthreads version of *dotprod* to use 8 threads, and compile
  - *$ cc -lpthread -o dotprod dotprod.c*
- Run the program, and save the output of the run.
- Modify the pthreads version of *dotprod* by increasing the vector length to 10,000,000. Compile and run it, save the output.

pthread-step by step (2)

- logon to tetra.cs.ucdavis.edu and perform steps 1-4. Tetra is a quad core system.

MPI

- Run MPI version of code on various numbers of cluster members.

MPI-step by step (1)

- logon to one of the ECE unix systems.
- Create a *lamboot* config file listing the hostname of the current system, as well as 3 other ece systems.
  - **indigo.ece.ucdavis.edu**
  - mamba.ece.ucdavis.edu
  - redbelly.ece.ucdavis.edu
  - viper.ece.ucdavis.edu
- Run the *lamboot* command to startup up MPI.
  - *$ lamboot -v lam_boot_schema*
- Run the *lamnodes* command and save the output in the log file.

MPI-step by step (2)

- Compile MPI version of the dotprod.c file, using the following command.
  - *$ mpicc -g -o dotprod dotprod.c*
- Run dotprod across 4 nodes, logging the output to hand in.
  - *$ mpirun -c 4 dotprod --*
- Modify the MPI version of dotprod.c to use a vector length of 10,000,000, compile and rerun. Log the output to hand in.

Turn In

- 1 file will be submitted via SmartSite.
  - A zip file
    - The log files from the pthreads and MPI runs
    - The file containing the notes you recorded in response to the question, i.e., pthreads step 1.
    - A text file named “README” that describes each of the log and note file.

- **DUE DATE: Wednesday 5/6 at 5PM.**