ECE 574 – Cluster Computing
Lecture 12

Vince Weaver

http://www.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

8 October 2015

Announcements

- Homework #4
- Midterm on the 20th

Shared Memory vs Distributed Systems

Shared Memory

- How big can a shared-memory machine be?
- SGI UV systems at least 4096 cores and 16TB running one Linux image

  http://www.techeye.net/hardware-2/sgi-builds-pittsburgh-4096-processor-core-16tb-shared-memory-supercomputer

  Scaling Linux to the Extreme paper
- Digression about SGI
- What are the challenges? Locking contention?

- Benefits?
  (Relatively) easy to code?
  Easier to port code
  Many libaries do it for you. For example, OpenBLAS.

Distributed System

- Communicate over a network
- Many systems each with own memory, communicate via Message passing
- Each node has own copy of operating system

Network Topology

- Packet-switching vs bus
- Ring, mesh, star, line, tree, fully connected
- Cube, hypercube
- Mesh networks and routing

Network Types

- Latency vs Bandwidth
- Top500 in Jun 2015:

interconnect	#
infiniband FDR	160
10GB ethernet	83
infiniband QDR	73
gigabit ethernet	63
Cray Gemini	15

- Ethernet – 10/100/1Gb/10GB/40Gb/s

- InfiniBand – low latency, most common in supercomputers
  copper or fiber
  QDR with 12 channel, 96Gb/s, FDR with 12 channel, 163Gb/s, EDR with 12 channel, 290Gb/s
- Cray Gemini – Mesh/torus – 64Gb/s
- Fibrechannel
- Older: custom, Myrinet

Cluster Building

- Get at least two computers
- Install Linux on both
- Setup a network so they can talk to each other
- Setup common authentication method
- Setup some shared disk space
- Install some sort of message-passing library (MPI is common)
- Optional: network boot: why?
  (updates, keeping things in sync)

- Optional: cluster management software, like ganglia, etc.
- Optional: job submission software

Message Passing Interface (MPI)

Abstraction for sending chunks of data around network. You can put together an array of 100 floats, and say ”send this to process Y” and like magic it appears there.

Can you implement by hand?

- Sort of how you can use pthread directly?
- Yes, use ssh (like rsh) to run copy of your program on all machines
- Then write custom network code to open sockets and communicate among them all
- Network code is a pain
- Just crying out for abstraction

MPI

- Message Passing Interface
- Distributed Systems
- MPI 1.0 – 1994. MPI 3.0 – 2012
- MPI 1.2 widely used. MPI2.0 is complicated and adoption not as high as it could be.
- MPICH – CH stands for Chameleon – Argonne and Missippi State

- MVAPICH – from Ohio State, based on MPICH

- OpenMPI – merger of 3 MPI implementations: FT-MPI from the University of Tennessee, LA-MPI from Los Alamos National Laboratory, and LAM/MPI from Indiana University

- Any other options? PVM was a predecessor

- Python Bindings, Java bindings, Matlab

Writing MPI code

- `#include "mpi.h"`
- Over 430 routines
- use `mpicc` to compile
- mpirun -n 4 ./test_mpi
- MPI_Init() called before anything else
- MPI_Finalize() at the end

Communicators

- You can specify communicator groups, and only send messages to specific groups.

- `MPI_COMM_WORLD` is the default, means all processes.

Rank

- Rank is the process number.
- `MPI_Comm_rank(MPI_Comm comm, int size)`
- You can find the number of processes:
  `MPI_Comm_size(MPI_Comm comm, int size)`