

Cluster Parallelism and the Message Passing Interface

Or ...

How I Spent My
Summer Vacation

by

Tim Reilly

Overview

Parallelism

- An introduction

Clustering

- Some background

MPI

- Bringing them together

Part 1

Parallelism: An Introduction

Why we do it

Exploiting hardware resources

Mixing computation/communication

Decreasing 'wasted' resources

Implementation

- Instruction
- Application
- Operating System
- Hardware-Assisted

Data Access Models

Shared Everything

- Threads, Distributed Shared Memory

Shared Nothing

- Processes, Message Passing

Shared Everything

- Resources
 - Implicitly shared
- Synchronization
 - Mutual Exclusion

Shared Nothing

Resources

- Local to execution unit

Synchronization

- Message passing

Part 2

Clusters

What They Are

Commodity hardware

- Distinct machines

High speed interconnect

Why it's done

Cost

- 2000 CPU machine
- 2000 machines

Scalability

Exploiting clusters

- Manually
 - Run tasks on individual machines
- Automatically
 - Requires abstraction
 - Cluster level: MOSIX
 - Software level: MPI

Data Access Models, revisited

- Shared Everything
 - Distributed shared memory
- Shared Nothing
 - message passing (not MPI specifically)

Part 3

Message Passing Interface

What it is

- A Specification
 - Many implementations
 - MPICH, LAM-MPI, OpenMPI, others
- A Library
 - Existing code unaffected
- Platform neutral
 - Standard mandates interoperability

Strengths

Simplicity

- I/O model is similar to other models

Portability

- Differing MPI implementations
- Differing compilers
- Differing hardware

Strengths (cont.)

Scalable

- Suitable for:
 - Multi-core machines
 - Multi-socket machines
 - Clusters

Strengths (cont. again)

Useful abstractions

- Reduce
 - Take a variable that exists across nodes, perform a function on them, aggregate them to another
- Communication groups
 - Create virtual groups of nodes

Weakness

Not suited for all parallel tasks

- GUI interaction
- I/O centric tasks
- Can be done, but not designed for it

Example app

Monte Carlo Pi estimation

- On Generals (our old cluster)
 - 1 CPU: ~130 sec
 - 2 CPU: ~ 65 sec
 - 4 CPU: ~ 33 sec
 - And the pattern follows as more CPUs/machines added

The code, in brief

(External to slide)

Further Information

Books

Using MPI; Gropp, Lusk, Skjellum

Web

- <http://www-unix.mcs.anl.gov/mpi/>

Permission to reproduce these slides for personal use only is granted to the persons in attendance of this presentation. All other uses require explicit permission.

Contact Information

Timothy Reilly

treilly1@cs.binghamton.edu