An Introduction to Parallel Systems

Lecture 1 - Who, What, Why, Where, When?

Martin Brain

University of Bath

November 15, 2007

Introduction (Who & Where)

- Martin Brain, 3^{rd} year PhD Student, University of Bath
  `ma9mjb@bath.ac.uk`
- Course website
  `http://www.cs.bath.ac.uk/~mjb/parallel/`
- An *introduction* to ...
- “multi-disciplinary”

When

Week 1	Introduction – Who, What, Why, Where, When?
Week 2	Data Parallelism and Vector Processors
Week 3	Message Passing Systems
Week 4	Shared Resource Parallelism

Why?

Why bother learning about parallel systems?

Why?

Why bother learning about parallel systems?

- Faster
- Use of parallel hardware
- More efficient use of hardware
- Reliability
- Natural programming model

What?

What are parallel systems?

- Hardware (within core, multi-core, multi-processor)
- Within the operating system kernel
- Programming languages / userspace
- Between computers
- Between groups of computers
- Between people

Classification of Parallel Systems

- Synchronous vs. asynchronous
- Homogeneous vs. heterogeneous
- Static vs. dynamic
- Reliable vs. unreliable
- Scalability
- Granularity
- Concurrency

How?

1. Take an *algorithm* (not a task) and find the bits that can be done simultaneously - i.e find the bits that are independent.
2. Consider what order sub tasks have to be done and the dependencies between them.
3. (Access to) resources are the limit.

Resource Sharing

no sharing ↔ explicit sharing ↔ implicit sharing

data parallel ↔ message passing ↔ shared resource

Examples

- Matrix calculations
- Search engines
- Game playing / search algorithms
- Virtual world
- Databases
- Climate simulation

The Good

- Amdahl's Law:
  $0 \leq P \leq 1$, proportion of task that can be done in parallel.
  N, the number of nodes.

$$speedup = \frac{1}{(1 - P) + \frac{P}{N}}$$

The Good

- Amdahl's Law:
  $0 \leq P \leq 1$, proportion of task that can be done in parallel.
  N, the number of nodes.

$$speedup = \frac{1}{(1 - P) + \frac{P}{N}}$$

- Superlinear speed up is possible

The Good

- Amdahl's Law:
  $0 \leq P \leq 1$, proportion of task that can be done in parallel.
  N, the number of nodes.

$$speedup = \frac{1}{(1-P)+\frac{P}{N}}$$

- Superlinear speed up is possible

Before you start, work out how much improvement you can expect – is it worth it?

The Bad

Running the system may be non deterministic / timing dependant.

The Bad

Running the system may be non deterministic / timing dependant.

- Can't find bugs by testing
- Hard to debug
- Hard to profile

The Bad

Running the system may be non deterministic / timing dependant.

- Can't find bugs by testing
- Hard to debug
- Hard to profile

Design and formal modelling are *important*.

Problems Unique to Parallel Systems

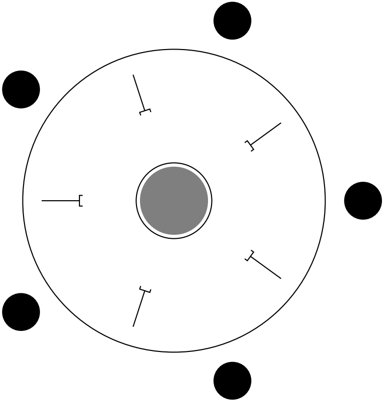
Include …

- Race conditions

  ```
  array[numberOfItems++] = input;
  ```
- Synchronisation of processes
- Synchronisation of data (replication)

The Dining Philosopher's Problem

Something to think about ...



Conclusion

- Parallel systems exist at many different levels in computing and have a variety of properties.
- Potentially linear speed up (or more) but introduce a number of theoretical and practical problems.
- Resources and the sharing of resources are key.

Questions?

Questions?

Made using only Free Software