Parallel Computing
CM30225

Russell Bradford

2023–2024

CM30225

Parallel computing as a topic has been around for as long as computers have been around

CM30225

Parallel computing as a topic has been around for as long as computers have been around

But recently it has come back into fashion…for reasons to be explored in this Unit

CM30225

Parallel computing as a topic has been around for as long as computers have been around

But recently it has come back into fashion…for reasons to be explored in this Unit

You have PCs, laptops and phones that are multicore: multiple processors are in the mainstream

CM30225

Parallel computing as a topic has been around for as long as computers have been around

But recently it has come back into fashion…for reasons to be explored in this Unit

You have PCs, laptops and phones that are multicore: multiple processors are in the mainstream

This Unit will look at hardware and software in the context of parallel computing

Unit Outline

Structure of this unit: starting with 3 hours lectures per week

- Wednesday 10:15
- Thursday 10:15
- Friday 14:15

Unit Outline

Structure of this unit: starting with 3 hours lectures per week

- Wednesday 10:15
- Thursday 10:15
- Friday 14:15

The aim is to cover the necessary material early in the semester which will leave the last few weeks free for revision and problems classes; and to lay the groundwork for the assignments

Unit Outline

Assessment

Usual combination of assessed coursework and exam: two pieces of coursework plus exam

1. Shared memory programming (15%)
2. Distributed memory programming (10%)
3. End of unit exam (75%)

Unit Outline

Assessment

Coursework timelines (subject to change):

1. set Thu 19 Oct
   due Wed 15 Nov
2. set Thu 16 Nov
   due Mon 8 Jan 2024

Feedback on coursework will be provided via Moodle. There will be general feedback that applies to many people and some individual feedback

Note that marking parallel programs is *very* time intensive (for reasons you will learn in thus unit!), so please don't expect a speedy turnaround

C

The coursework will be writing some parallel programs in C on a supercomputer

C

The coursework will be writing some parallel programs in C on a supercomputer

Though you **must** already be familiar with writing C, you may wish to brush up on your C in preparation

C

The coursework will be writing some parallel programs in C on a supercomputer

Though you **must** already be familiar with writing C, you may wish to brush up on your C in preparation

There is a “Remind Yourself About C” document on the Unit Web page

Unit Outline

Week 6 (starting 6th Nov) will be a “consolidation week”

Unit Outline

Week 6 (starting 6th Nov) will be a “consolidation week”

No lectures for the whole of Computer Science (CM Units)

Unit Outline

Week 6 (starting 6th Nov) will be a “consolidation week”

No lectures for the whole of Computer Science (CM Units)

Presumably other Departments will carry on as usual

Unit Outline

Aims To give students the ability to recognise and understand the problems and opportunities afforded by parallel systems; to recognise the differing types of parallelism available and make advised choices between them; and to take advantage of progress in technology as modern computers become ever more parallel.

Unit Outline

Learning Outcomes Students will be able to:

1. write and debug simple parallel programs;
2. recognise the issues surrounding concurrent access to data;
3. describe the various kinds of parallel hardware, parallel programming methodologies and the relationship between them

Unit Outline

Skills required:

1. Comfortable writing C

Unit Outline

Skills required:

1. Comfortable writing C
2. Ability to think through complicated situations

Unit Outline

1. Basics: supercomputers and the consequences of Moore's Law; bandwidth vs latency; speedup, efficiency, scalability; Amdahl's & Gustafson's Laws; Flynn's taxonomy, SPMD; distributed, shared, NUMA and other memory architectures.

Unit Outline

1. Basics: supercomputers and the consequences of Moore's Law; bandwidth vs latency; speedup, efficiency, scalability; Amdahl's & Gustafson's Laws; Flynn's taxonomy, SPMD; distributed, shared, NUMA and other memory architectures.
2. Shared memory computing: multicore systems (cache coherence and bottlenecks); mutual exclusion and critical regions; low level constructs including POSIX threads and synchronisation methods such as barriers, locks, semaphores, etc.; language-level support including monitors, OpenMP; vector and array (SIMD), HPF, Cn.

Unit Outline

3. Distributed computing: clusters, message passing, MPI. Programming using MPI (and SLURM).

Unit Outline

3. Distributed computing: clusters, message passing, MPI. Programming using MPI (and SLURM).
4. Parallel algorithms and data structures.

Unit Outline

3. Distributed computing: clusters, message passing, MPI. Programming using MPI (and SLURM).
4. Parallel algorithms and data structures.
5. Topics in Parallel Computing: examples might include HPC; MapReduce; distributed file systems; the Grid; GPGPU and OpenCL; instruction level parallelism (SWAR, VLIW).

Here Be Dragons

Note that this is a Final Year Unit, so is a lot more stretching than previous years. It contains a lot of material as parallelism is a big subject

Here Be Dragons

Note that this is a Final Year Unit, so is a lot more stretching than previous years. It contains a lot of material as parallelism is a big subject

Also it is *very important* that you are a confident programmer with good experience in C. Otherwise you will be spending a disproportionate amount of time on the coursework. Do think very carefully about this

Here Be Dragons

Note that this is a Final Year Unit, so is a lot more stretching than previous years. It contains a lot of material as parallelism is a big subject

Also it is *very important* that you are a confident programmer with good experience in C. Otherwise you will be spending a disproportionate amount of time on the coursework. Do think very carefully about this

Many in the past have assumed “it will be ok, I can wing it”, and subsequently had great difficulty in the coursework

Here Be Dragons

Note that this is a Final Year Unit, so is a lot more stretching than previous years. It contains a lot of material as parallelism is a big subject

Also it is *very important* that you are a confident programmer with good experience in C. Otherwise you will be spending a disproportionate amount of time on the coursework. Do think very carefully about this

Many in the past have assumed “it will be ok, I can wing it”, and subsequently had great difficulty in the coursework

The coursework is trivial as a sequential program, but very testing as a parallel program

Unit Outline

Resources

The subject of Parallel Computing is nearly as old as that of computers and so there are *lots* of books

Unit Outline

Resources

The subject of Parallel Computing is nearly as old as that of computers and so there are *lots* of books

None of them really suitable for this course, as we will try to take a broad overview of the subject

Unit Outline

Resources

The subject of Parallel Computing is nearly as old as that of computers and so there are *lots* of books

None of them really suitable for this course, as we will try to take a broad overview of the subject

Part of the problem of parallel computing is that there is no simple unified model (like von Neumann for sequential computing), and everybody has their own idea on how things should be done

Unit Outline

Resources

The subject of Parallel Computing is nearly as old as that of computers and so there are *lots* of books

None of them really suitable for this course, as we will try to take a broad overview of the subject

Part of the problem of parallel computing is that there is no simple unified model (like von Neumann for sequential computing), and everybody has their own idea on how things should be done

Leading to loads of books saying “this is the one true path to parallel computing”

Unit Outline

Resources

The subject of Parallel Computing is nearly as old as that of computers and so there are *lots* of books

None of them really suitable for this course, as we will try to take a broad overview of the subject

Part of the problem of parallel computing is that there is no simple unified model (like von Neumann for sequential computing), and everybody has their own idea on how things should be done

Leading to loads of books saying “this is the one true path to parallel computing”

Take them with a pinch of salt!

Unit Outline

Resources

Some books I found on my shelf:

Hardware

- “Highly Parallel Programming”, Almasi & Gottlieb, Benjamin Cummings
- “Computer Architecture and Parallel Processing”, Hwang & Briggs, McGraw-Hill

Unit Outline

Resources

Software

- “Concurrent Programming Principles and Practice”, Andrews, Benjamin Cummings
- “Introduction to Parallel Computing”, Kumar, Grama, Gupta, Karypis, Benjamin Cummings
- “Concurrent Programming”, Burns & Davies, Addison-Wesley
- “Designing and Building Parallel Programs”, Foster, Addison Wesley
- “Distributed Algorithms”, Lynch, Morgan Kaufmann

Unit Outline

Resources

Theory

- “Principles of Concurrent and Distributed Programming”, Ben-Ari, Prentice Hall
- “Communicating Sequential Processes”, Hoare, Prentice Hall

Unit Outline

Resources

N.B. Some of these were given to me by the publishers so I’m not saying they are the best books out there

Unit Outline

Resources

N.B. Some of these were given to me by the publishers so I’m not saying they are the best books out there

The thing to do it look at several and find one that suits you: they contain roughly the same material

Unit Outline

Resources

You don’t need me to tell you that there is a large amount of material out there on the Web?

Unit Outline

Resources

You don’t need me to tell you that there is a large amount of material out there on the Web?

Wikipedia is fairly accurate in this area: but, as usual with Wikipedia, you should check with other sources

Unit Outline

Resources

You don’t need me to tell you that there is a large amount of material out there on the Web?

Wikipedia is fairly accurate in this area: but, as usual with Wikipedia, you should check with other sources

There is a Unit Moodle page, but as Moodle is so horrible I tend to use my own Web page:

`http://people.bath.ac.uk/masrjb/CourseNotes/cm30225.html`

Standard Introductory Slides

Remember:

Standard Introductory Slides

Remember:

You are expected to do some work outside of lectures

Standard Introductory Slides

Remember:

You are expected to do some work outside of lectures

Lectures are the *start* of the learning process, not the end!

Standard Introductory Slides

Remember:

You are expected to do some work outside of lectures

Lectures are the *start* of the learning process, not the end!

These slides are reminders to me on what to say in lectures

Standard Introductory Slides

Remember:

You are expected to do some work outside of lectures

Lectures are the *start* of the learning process, not the end!

These slides are reminders to me on what to say in lectures

They are often abbreviated in style, and so are not the whole story and would not be suitable to be quoted verbatim in an exam

Standard Introductory Slides

Don’t try to copy everything down from the slides in lectures—the slides will be available after each lecture

Instead, make a note of what is important and use that later—in conjunction with the slides—to guide your further reading and study

Standard Introductory Slides

Do not rely purely on my notes for your revision

Standard Introductory Slides

Do not rely purely on my notes for your revision

People who do this live to regret it

Standard Introductory Slides

Do not rely purely on my notes for your revision

People who do this live to regret it

Like every Unit, you are expected to read around the subject for yourself

Standard Introductory Slides

Do not rely purely on my notes for your revision

People who do this live to regret it

Like every Unit, you are expected to read around the subject for yourself

You need to take your own notes, read, and *participate*

Standard Introductory Slides

Do not rely purely on my notes for your revision

People who do this live to regret it

Like every Unit, you are expected to read around the subject for yourself

You need to take your own notes, read, and *participate*

You don't expect to get fit simply by paying to joining a gym…

> *"If you have college courses in CS, buy the books and spend day and night the few days before class going through the books and taking notes and answering questions and programming examples before the first class even starts. If you really want to do this in your life, that's what you should do, not just wait for the education to be handed you. Those who finish at the top will always be in high demand. You can learn outside of school too but you have to put a lot of time into it. It doesn't come easily. Small steps, each improving on the other, is what to expect, not instant understanding and expertise."*

Steve Wozniak, co-founder of Apple

Standard Introductory Slides

Computer Science is not a spectator sport

Anon