Parallel Computing
CM30225

Russell Bradford

2020–2021
Parallel computing as a topic has been around for as long as computers have been around
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
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.

Many of you have laptops or phones that are multicore: multiple processors are in the mainstream.
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

Many of you have laptops or phones that are multicore: multiple processors are in the mainstream

This Unit will look at hardware and software in the context of parallel computing
Structure of this unit:

- Pre-recorded “lectures”, released week-by-week on Re:View/Panopto. See Moodle for links. The main delivery of the content of this unit. These will be in one or more chunks each week: they will vary in number and length as is appropriate for the topic being discussed.
Unit Outline

- Live Interactive Online sessions:
  - Tuesdays 13:15
  - Wednesdays 12:15

On Zoom. See Moodle for links. These sessions will be reactive (“make it up as we go along”) for things like questions, supplementary discussions, coursework, or anything (relevant) you want.
These sessions will be recorded (to Re:View/Panopto) for the benefit of those people who can’t make the timeslot.
Unit Outline

These sessions will be recorded (to Re:View/Panopto) for the benefit of those people who can’t make the timeslot.

If you have a personal issue with being recorded, make sure your camera and microphone are off. You can still interact via chat.
These sessions will be recorded (to Re:View/Panopto) for the benefit of those people who can’t make the timeslot.

If you have a personal issue with being recorded, make sure your camera and microphone are off. You can still interact via chat.

It is possible that we will only need one of these hours each week: this we can decide as necessary.
Usual combination of assessed coursework and exam: two pieces of coursework plus exam
Unit Outline
Assessment

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

1. Shared memory programming (15%)
Usual combination of assessed coursework and exam: two pieces of coursework plus exam

1. Shared memory programming (15%)
2. Distributed memory programming (10%)
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%)
Coursework timelines (subject to change):

1. set Tue 20 Oct
due Mon 16 Nov

2. set Tue 17 Nov
due Mon 4 Jan 2021

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, so please don’t expect a speedy turnaround!
The coursework will be writing some parallel programs in C on the University’s cluster Balena
The coursework will be writing some parallel programs in C on the University’s cluster Balena.

Though you should already be familiar with C, you may wish to brush up on your C yourself in preparation.
The coursework will be writing some parallel programs in C on the University’s cluster Balena.

Though you should already be familiar with C, you may wish to brush up on your C yourself in preparation.

And we can have a C revision class at a suitable point, if requested.
Week 6 will be a “consolidation week”
Unit Outline

Week 6 will be a “consolidation week”

No lectures for the whole of Computer Science (CM Units)
Week 6 will be a “consolidation week”

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

Presumably other Departments will carry on as usual
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:
Skills required:

1. Comfortable writing C (or, at a pinch, Fortran)
Skills required:

1. Comfortable writing C (or, at a pinch, Fortran)
2. Ability to think through complicated situations
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.
3. Distributed computing: clusters, message passing, MPI. Programming using MPI (and SLURM).
3. Distributed computing: clusters, message passing, MPI. Programming using MPI (and SLURM).
4. Parallel algorithms and data structures.
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).
The subject of Parallel Computing is nearly as old as that of computers and so there are *lots* of books.
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
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.
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”
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
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
N.B. Some of these were given to me by the publishers so I’m not saying they are the best books out there
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 is look at several and find one that suits you: they contain roughly the same material
You don’t need me to tell you that there is a large amount of material out there on the Web?
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.
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:
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
Do not rely purely on my notes for your revision
Do not rely purely on my notes for your revision

People who do this live to regret it
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
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
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
Computer Science is not a spectator sport

Anon