Peter Jimack

University of Leeds

Lecture 20: Summary

Previous lectures

This module has been structured to focus on one parallel archi

1.
2.
3. General purpose GPU (with OpenCL); Lectures 14-19.

The practical elements of the module *worksheets and courseworks)* also followed this structure.

- Some *mentions* were out-of-order, *e.g.* OpenMP barriers mentioned in Lectures 11 and 17.

Today's lecture

Want to understand **general parallel programming concepts** that transcend particular architectures.

- e.
-

In this final lecture we will summarise all of the mater **parallel concept** rather than by architectu

- Easier to see the commonalities.

At the end I'll also spend a few minutes talking about the **Final assessment** for the module!

Why parallel?

Lectures 1 and 4

Parallel hardware allows **simultaneous** computations.

-
- could be *e.g.* time-sharing on a single c

Want to attain good **scaling** - decrease i

time t_{p} for increasing number of **processing units** *(threads, processes* etc.*)* p.

Measuring parallel performance

Lecture 4

Two useful metrics for parallel performance are the **speed-up** S and **efficiency** E

Achieving $S = p$ (*i.e.* $E = 1$) usually regar difficult to achieve due to various **over**

- Synchronisation, load balancing, com calculations, . . .
- **Super-linear scaling** $S > p$ possible (but rare) due to memory cache.

Laws for maximum parallel performance

Lectures 4, 19

Strong scaling is when the problem size *n* is fixed.

- $\frac{1}{\;}$

-

Weak scaling allows *n* to increase with

- Related to the **Gustafson-Barsis l**

The **work-span model** provides anothe maximum *S* from **task-graphs** *[Lecture 19]*.

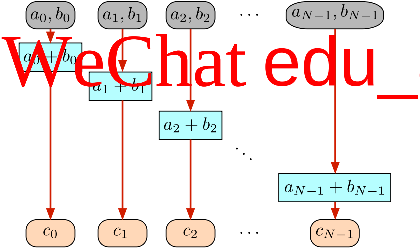
- $S \leq (\text{work})/(\text{span})$, with **work** and **span** determined from the task graph.

Loop parallelism and data dependencies

Lectures 3, 5, 9, 15

Initially looked at parallelising loops.

- If there are no **data dependencies**, is **data parallel** or a **map**.
-
-



Synchronisation

Lectures 7, 9, 11, 17

All but the simplest parallel problems require **synchronisation** betw

- **ation**.
-

Can lead to reduced **performance**.

- *e.g.* the extra operations required to achi

Can also lead to **deadlock** *[Lectures 7, 9]*.

- When one or more processing units wait for a synchronisation even that never occurs.

Load balancing and task parallelism

Lectures 13 and 19

Idle time is an example of poor **load balancing** *[Lecture 13]*.

-
-

Can improve load balancing by using a
independent **tasks** are sent to processing uni
become idle.

- This is an example of **dynamic** scheduling; can also be **static**.

Parallelising by **tasks** rather than loops is known as **task parallelism** *[Lecture 19]*.

-

In gen

- Can represent as a **task graph**, wit ks and **directed edges** denoted the dep
- For tasks that take the same time, can define t he total number of tasks, and the **span** as the length of the critical path.

Data reorganisation

Lecture 10

Parallel data reorganisation can be indexed by **read** locations ('gat

In sha n

example of a **data race** *(see later)*.

In distrubuted memory systems, can exploit **communication methods** that are usual

- One-to-many, many-to-one (also many-to-many).
- *e.g.* broadcasting, scattering and gathering.

Parallel hardware

Lectures 2, 8, 14, 16

Modern HPC clusters are increasingly using all three architectures:

- re

-

Most multi-core CPUs usually have **m**
cache coherency and **false sharing** [

Network connectivity affects communi
hypercube often used *[Lecture 8]*.

GPU's most suited for **data parallel problems** and have multiple types of **memory** *[Lectures 14, 16]*.

Data races / race conditions

Lectures 5, 6, 18

A **data race** potentially arises when two or more processing units read t .

-
-

Can control using **critical regions**.

- Exclusive access by a single processing unit
- Simple critical regions can be implemented more efficiently (*i.e.* by compiler and hardware) as **atomics**.

Lower level control

Lectures 7, 18

At a lo

mut

-
- Improper use of multiple locks can result in **deadlock**.

At an even lower level, locks can be implemented usi **exchange** and **atomic compare-and-exchange** *[Lecture 18]*.

- **Lock-free data structures** are desirable whenever possible.

Explicit communication

Lectures 9, 10, 12, 15, 19

If memory is distributed (in some sense), may need to use **explicit com**

- .
- Between CPU and GPU, *i.e.* **host**

Communication can be:

- **Blocking**: Returns once all resources safe to re-use.
- **Synchronous**: Does not complete until sender and receiver start their communication operations.

Latency hiding

Lectures 12, 19

Can improve performance by **overlapping** communication with com

Often used with **domain partitioning** *[Lecture 12]*.

Can also overlap host-device communication w a GPU *[Lecture 19]*.

- Can also perform calculations on host and device simultaneously.

The end

I will now say a few words about the final assessme this module...

The final assessment

This will be an online exam to be taken at 1500 (China time) on Tuesday 17th May 2022...

-

    -
    -
      on 17th May (unless you have been grante
- It is an "open book" exam but you will receive no c copying material directly from your lectur

- Your overall module grade will be the sum of your scores on the 3 courseworks and your score (out of 50) on this final assessment
    - Hence the final assessment is worth 50% of the total marks

The final assessment (continued)

- The paper will consist of two questions worth 25 marks each
    -
-

    - The 2020 and 2021 papers were both "open book" but the students were given more time to comple questions are longer than they will be this y
    - The 2018 and 2019 papers were both 2 hour they were "closed book" – so some section questions

- See the announcement on Minerva for some further advice

Your revision questions

- The final Zoom session will be held at 1730 on Tuesday 10th May
  -
  - questions (and my responses)
  - There will also be a Zoom session at 1730 tod you already have any questions for me!
- I will also answer any questions that you post o XJCO3221 Discussion Boards on Minerva prior to 1600 on 17th May
  - I will NOT answer any questions after the start of the final assessment!