Parallel Computing
s

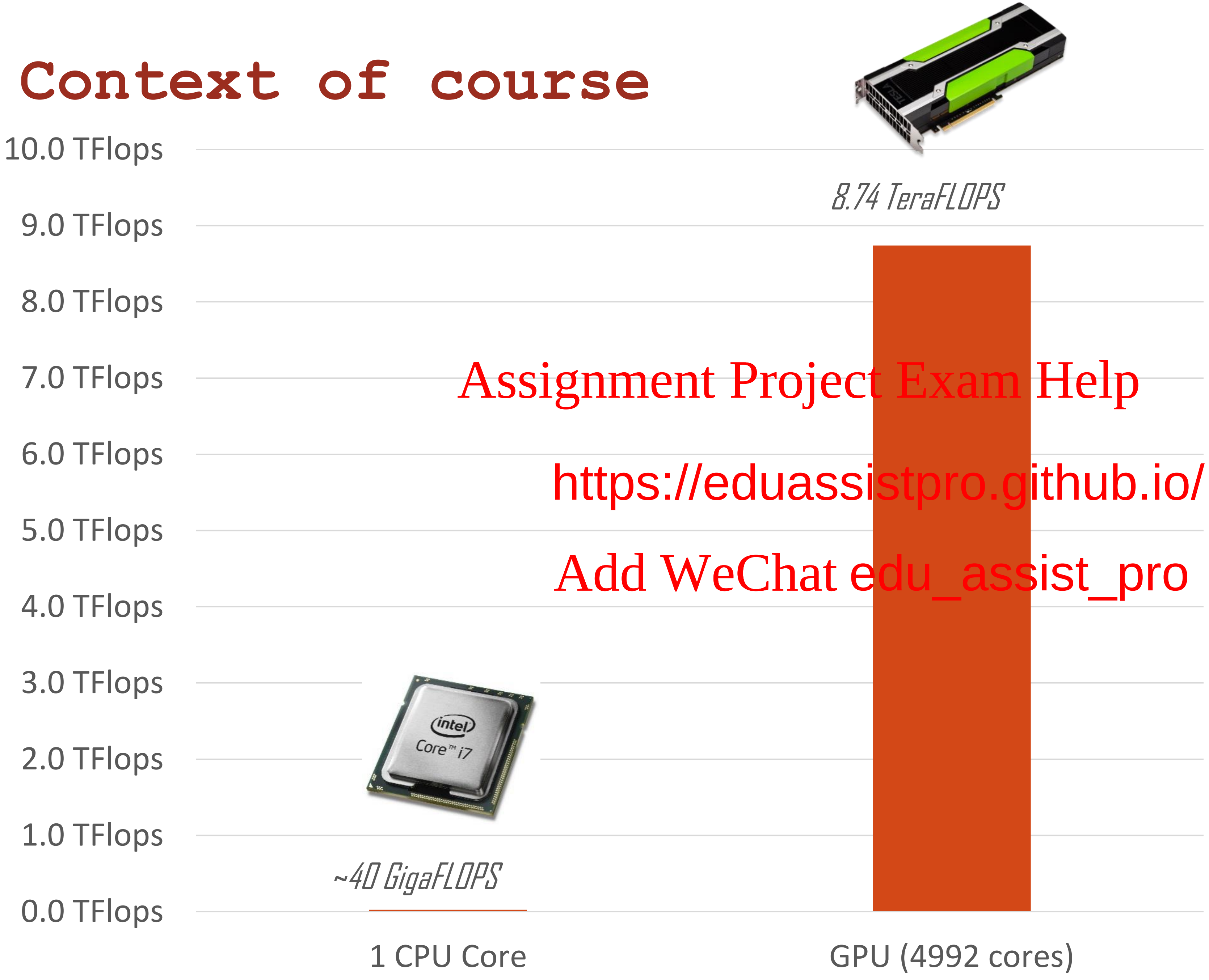
Dr Paul Ric
http://paulrichmond.shef.ac COM4521/

The University Of Sheffield.

NVIDIA GPU RESEARCH CENTER

- Context and Hardware Trends
- Supercomputing
- Software and Parallel Computing
- Course Outline

Context of course

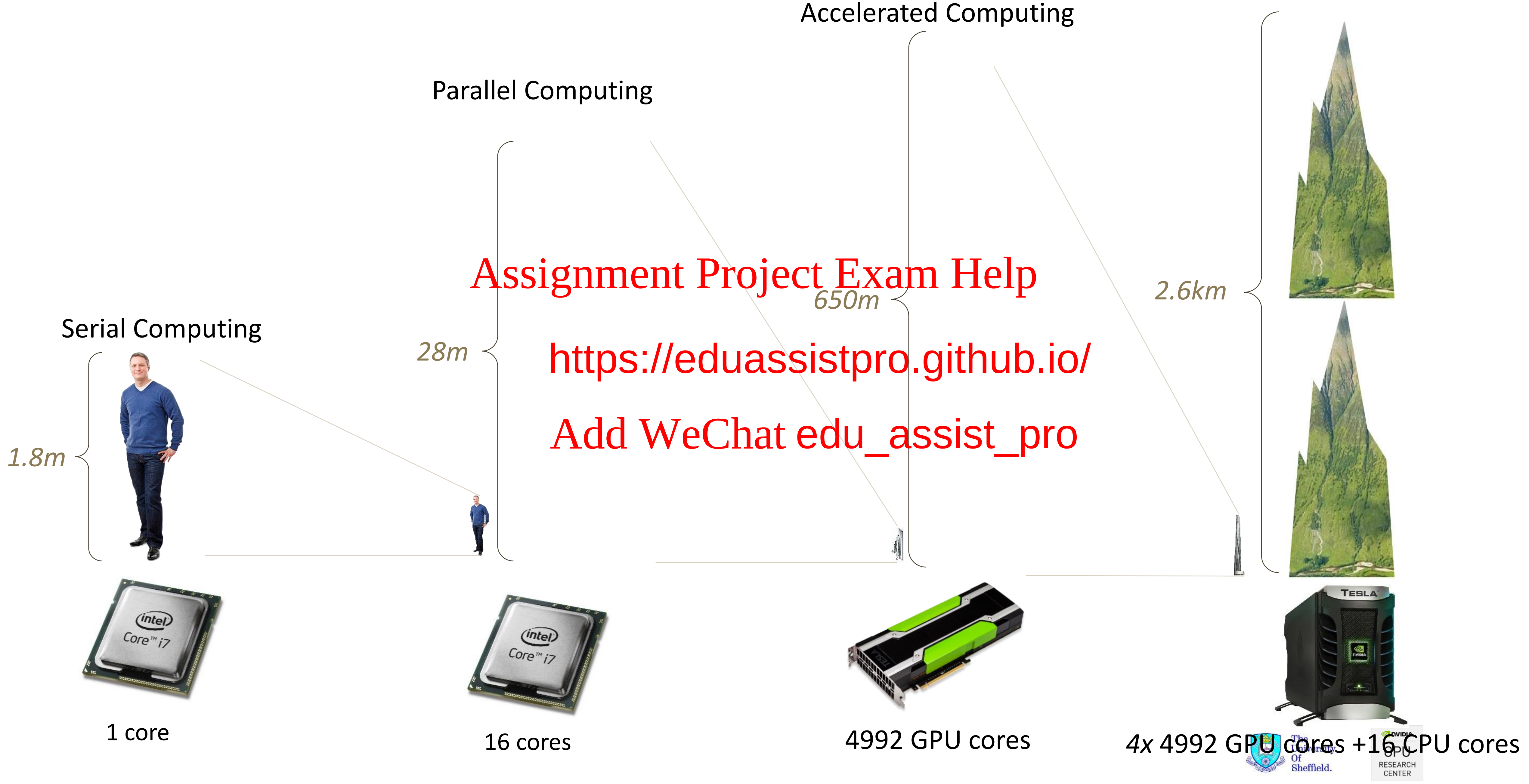


6 hours *CPU* time

vs.

1 minute *GPU* time

Scale of Performance



Scale of Performance: Titan Supercomputer

Transistors != performance

- Moores Law: A doubling of transistors every couple of years
  - Not a law actually an observation
  - Doesn't actually say anything about performance

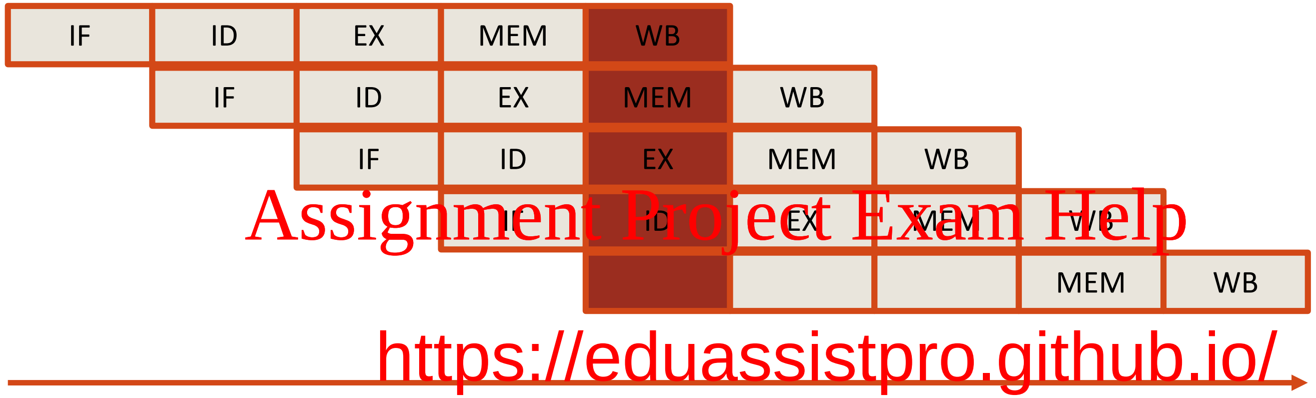
Dennard Scaling

"As transistors get smaller their power density stays constant"

$$\text{Power} = \text{Frequency} \times \text{Voltage}^2$$

- Performance improve[illegible]ionally realised by increasing frequency
- Decrease voltage to maintain a steady power
  - Only works so far
- Increase Power
  - Disastrous implications for cooling

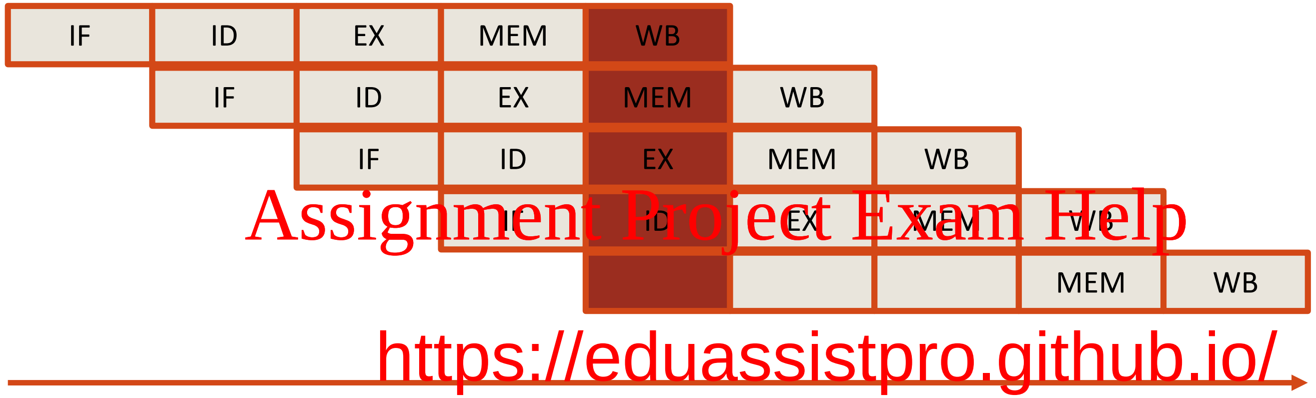
Instruction Level Parallelism



- Transistors used to build more co[illegible]itectures
- Use pipelining to overlap instruction execution

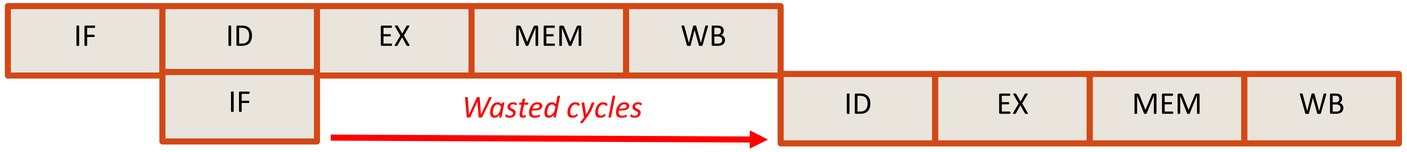
The University Of Sheffield.
NVIDIA GPU RESEARCH CENTER

Instruction Level Parallelism



- Transistors used to build more complex architectures
- Use pipelining to overlap instruction execution

```
add 1 to R1
copy R1 to R2
```



Golden Era of Performance

- 90s saw great improvements to single CPU performance
  - 1980s to 2002: 100% performance increase every 2 years
  - 2002 to now: ~40% every 2 years

Adapting to Thrive in a New Economy of Memory Abundance, K Bresniker et al.

Why More Cores?

- Use extra transistors for multi/many core parallelism
  - More operations per clock cycle
  - Power can be kept low
  - Processor designs can be simple – shorter pipelines (RISC)

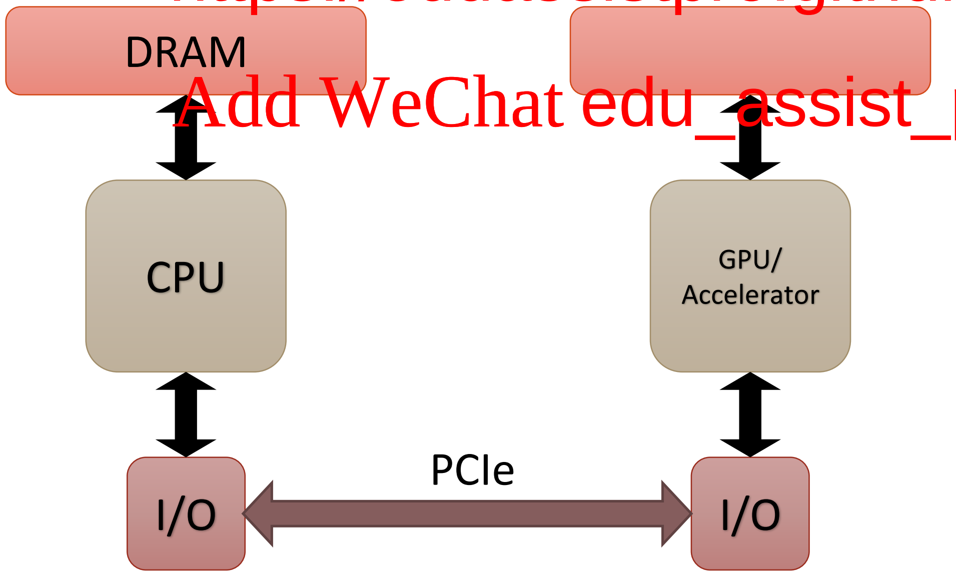
GPUs and Many Core Designs

- Take the idea of multiple cores to the extreme (many cores)
- Dedicate more die space to compute
  - At the expense of branch prediction, out of order execution, etc.
- Simple, Lower Power and
  - Very effective for HPC appl

om GTC 2017 Keynote Talk, NVIDIA CEO Jensen Huang

Accelerators

- ❑Problem: Still require OS, IO and scheduling
- ❑Solution: “Hybrid System”,
  - ❑CPU provides management and
  - ❑“Accelerators” (or co-processors) such as GPUs provide compute power



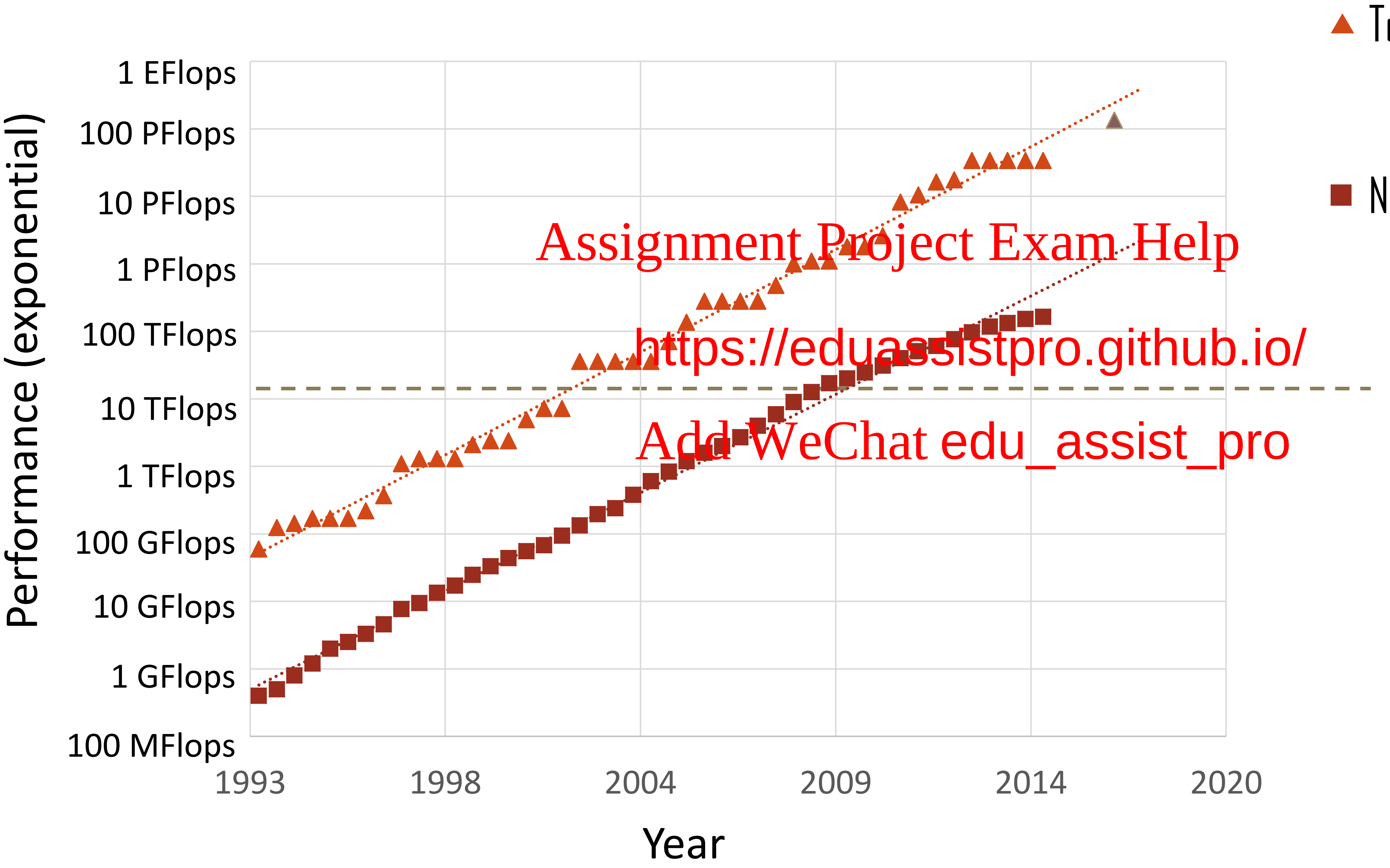
Types of Accelerator

- GPUs
  - Emerged from 3D graphics but now specialised for HPC
  - Readily available in workstations

- Xeon Phis
  - Many Integrated Core
  - Based on Pentium 4 design (x86) with units
  - Closer to traditional multicore
  - Simpler programming and compilation

- Context and Hardware Trends
- Supercomputing
- Software and Parallel Computing
- Course Outline

Top Supercomputers



Volta V100 (15TFLOPS SP)

Supercomputing Observations

- Exascale computing
  - 1 Exaflop = 1M Gigaflops
  - Estimated for 2020
- Pace of change
  - Desktop GPU top sup
  - A desktop with a GPU 08
  - A Teraflop of performance took 1MW
- Extrapolating the trend
  - Current gen top500 on every desktop in < 10 years

Trends of HPC

- Improvements at individual computer node level are greatest
  - Better parallelism
  - Hybrid processing
  - 3D fabrication
- Communication costs
  - Memory per core is re

Supercomputing Observations

https://www.nextplatform.com/2016/11/14/closer-look-2016-top-500-supercomputer-rankings/

Green 500

- Top energy efficient supercomputers

HPC Observations

- Improvements at individual computer node level are greatest
  - Better parallelism
  - Hybrid processing
  - 3D fabrication
- Communication costs are increasing
  - Memory per core is reducing
- Throughput > Latency

http://sc16.supercomputing.org/2016/10/07/sc16-invited-talk-spotlight-dr-john-d-mccalpin-presents-memory-bandwidth-system-balance-hpc-systems/

- Context and Hardware Trends
- Supercomputing
- Software and Parallel Computing
- Course Outline

Software Challenge

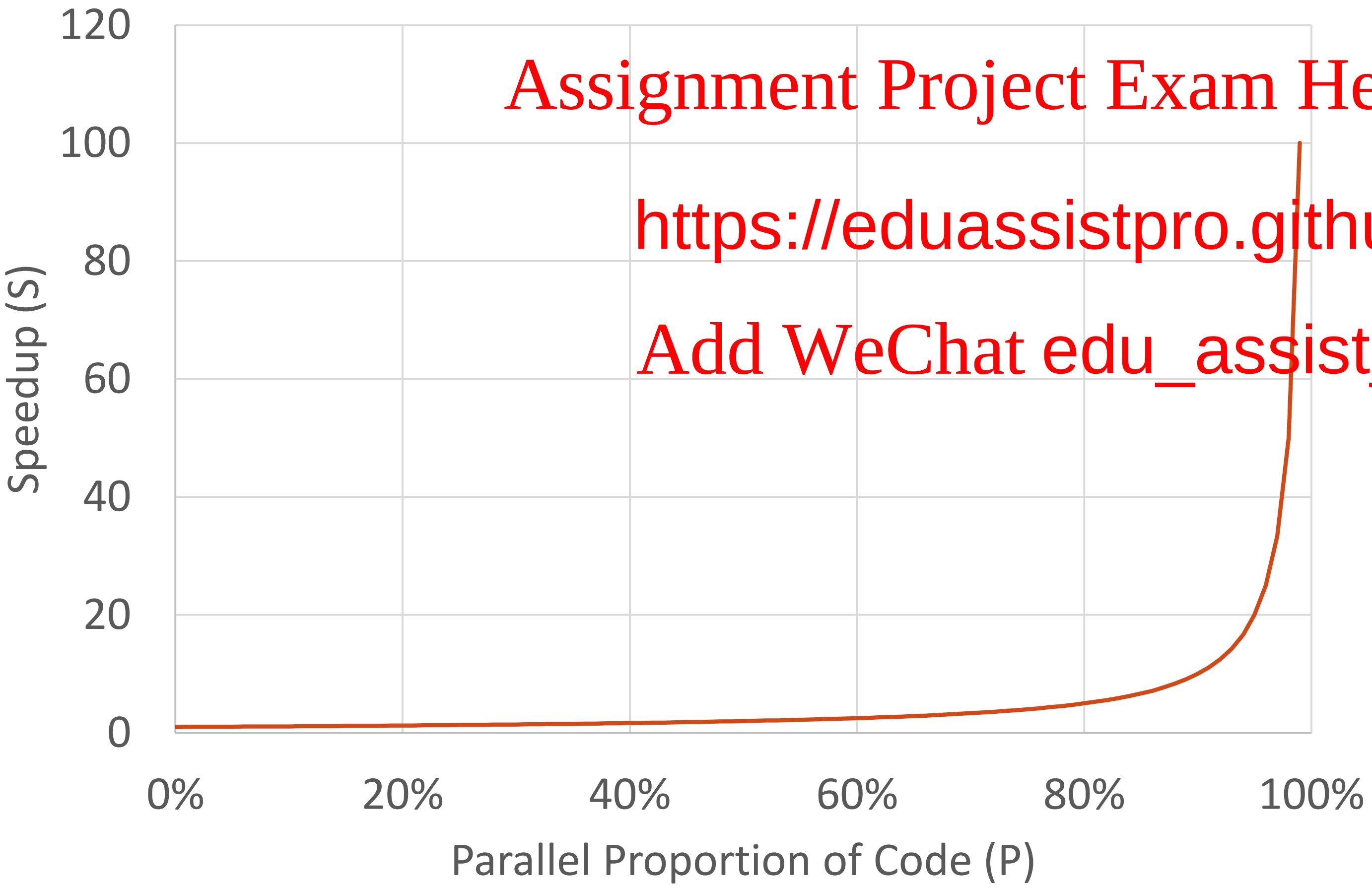
- How to use this hardware efficiently?

- Software approaches
  - Parallel languages: some limited impact but not as flexible as sequential programming
  - Automatic parallelisat rs of research hasn't solved this yet
  - **Design software with parallelisation in mind**

The University Of Sheffield.
NVIDIA GPU RESEARCH CENTER

Amdahl's Law

- Speedup of a program is limited by the proportion than can be parallelised



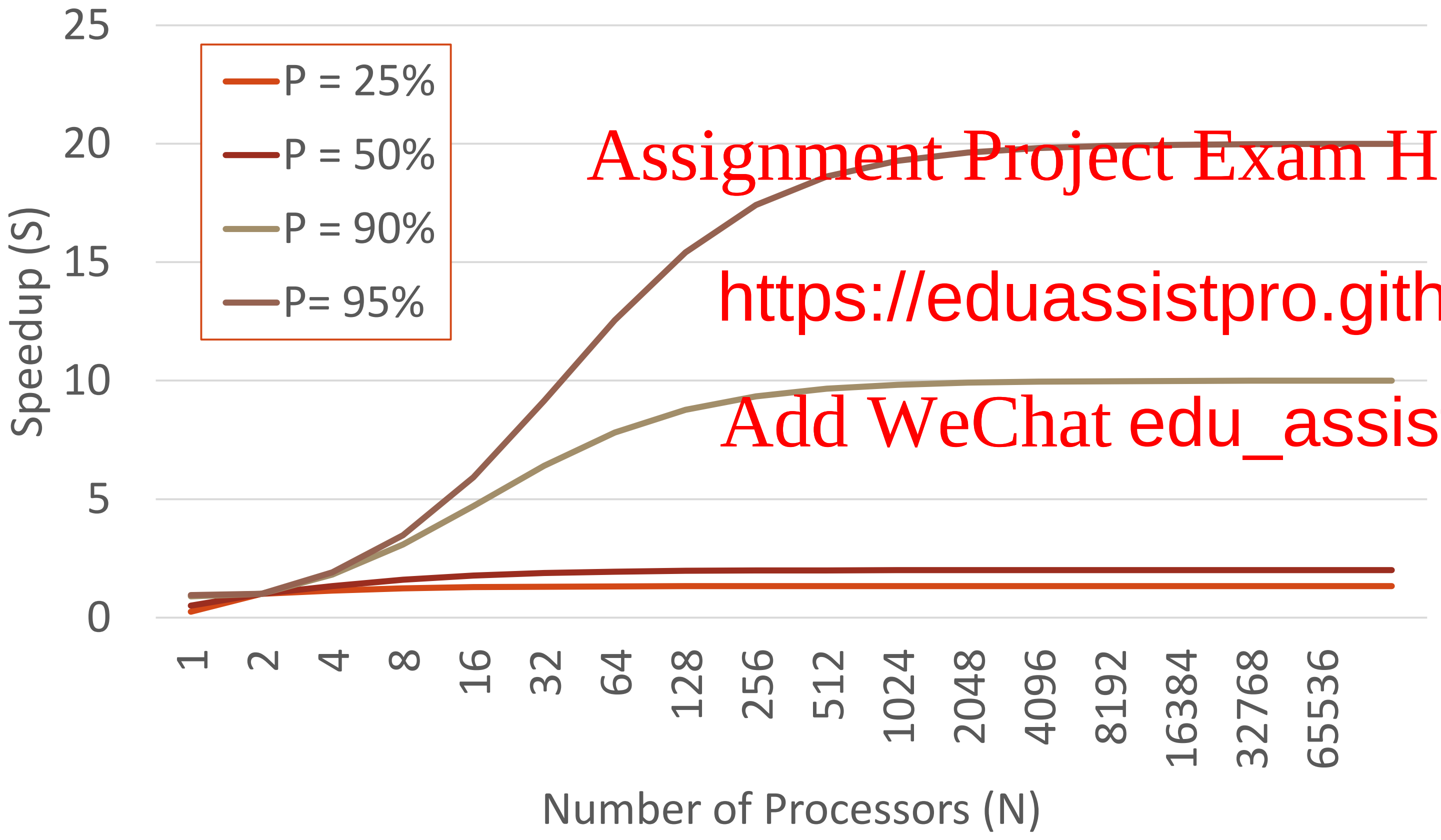
$$Speedup\ (S) = \frac{1}{1 - P}$$

The University Of Sheffield.

NVIDIA GPU RESEARCH CENTER

Amdahl's Law cont.

- Addition of processing cores gives diminishing returns



$$Speedup\ (S) = \frac{1}{\frac{P}{N} - (1 - P)}$$

Parallel Programming Models

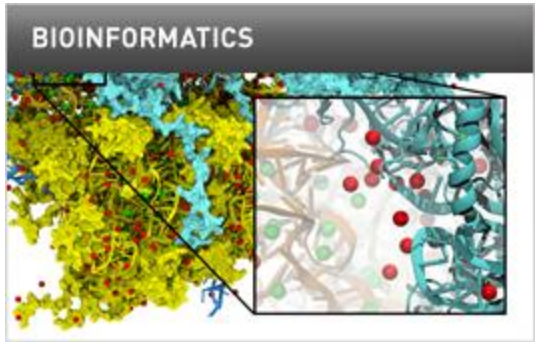
- Distributed Memory
  - Geographically distributed processors (clusters)
  - Information exchanged via messages
- Shared Memory
  - Independent tasks sha
  - Asynchronous memor
  - Serialisation and synchronisation to e tness
  - No clear ownership of data
  - Not necessarily performance oriented

Types of Parallelism

- Bit-level
  - Parallelism over size of word, 8, 16, 32, or 64 bit.
- Instruction Level (ILP)
  - Pipelining
- Task Parallel
  - Program consists of m
  - Tasks execute on asynchronous cores
- Data Parallel
  - Program has many similar threads of execution
  - Each thread performs the same behaviour on different data

Implications of Parallel Computing

- ❑Performance improvements
  - ❑Speed
  - ❑Capability (i.e. scale)



The University Of Sheffield.

GPU RESEARCH CENTER

- ❑Context and Hardware Trends
- ❑Supercomputing
- ❑Software and Parallel Computing
- ❑Course Outline

COM4521/6521 specifics

- Designed to give insight into parallel computing
  - Specifically with GPU accelerators
  - Knowledge transfers to all many core architectures
- What you will learn
  - How to program in C ... ually
  - How to use OpenMP t ... i-core CPUs
  - What a GPU is and how to program it ... DA language
  - How to think about problems in a highly parallel way
  - How to identify performance limitations in code and address them

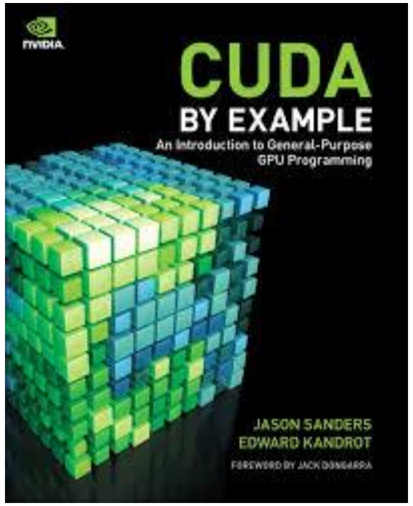
The University Of Sheffield.
NVIDIA GPU RESEARCH CENTER

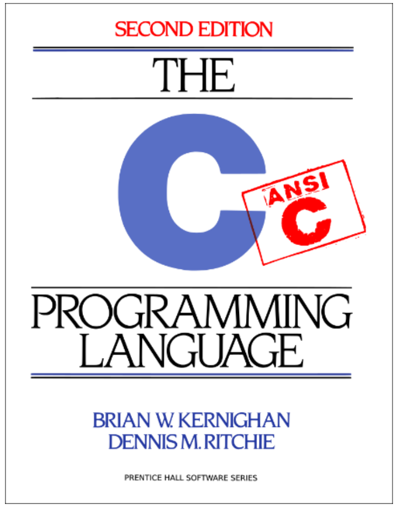
Course Mailing List

- A google group for the course has been set up
  - You have already been added if you were registered 01/02/2018
- Mailing list uses;
  - Request help outside of lab classes
  - Find out if a lecture ha
  - Want to participate in tent
- https://groups.google.com/a/sheffi forum/#!forum/com4521-group

Learning Resources

- Course website: http://paulrichmond.shef.ac.uk/teaching/COM4521/
- Recommended Reading:
  - Edward Kandrot, Jason Sanders, "CUDA by Example: An Introduction to General-Purpose GPU Programming", Addison Wesley 2010.
  - Brian Kernighan, Denn ming Language (2nd Edition)", Prentice Hal





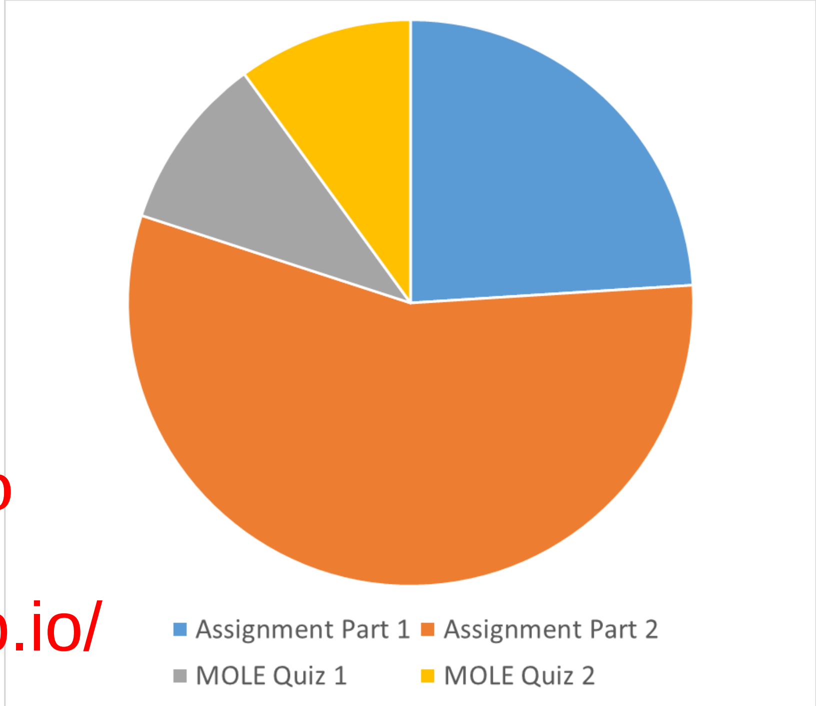
Timetable

- 2 x 1 hour lecture per week (back to back)
  - Monday 15:00 until 17:00 Broad Lane Lecture Theater 11
  - Week 5 first half of the lecture will be in DIA-LT09 (Lecture Theatre 9)
  - Week 5 second half of the lecture will be MOLE quiz in DIA-206 (Compute room 4)
- 1 x 2 hour lab per week
  - Tuesday 9:00 until 11:00 Diamond DIA-206 (Compute room 4)
  - Week 10 first half of the la z DIA-206 (Compute room 4)

- Assignment
  - Released in two parts
  - Part 1
    - Released week 3
    - Due for hand in on Tuesday week 7 (20/03/2018) at 17:00
    - Feedback after Easter.
  - Part 2
    - Released week 6
    - Due for hand in on Tuesday week 12 (15/05/2018) at 17:00

Course Assessment

- 2 x Multiple Choice quizzes on MOLE (10% each)
  - Weeks 5 and 10
- An assignment (80%)
  - Part 1 is 30% of the assignment total
  - Part 2 is 70% of the as
- For each assignment
  - Half of the marks are for the program a written report
  - Will require understanding of why you have implemented a particular technique
  - Will require benchmarking, profiling and explanation to demonstrate that you understand the implications of what you have done



The University Of Sheffield.

NVIDIA GPU RESEARCH CENTER

Lab Classes

- 2 hours every week
  - Essential in understanding the course content!
  - Do not expect to complete all exercises within the 2 hours
- Coding help from lab demonstrators Robert Chisholm and John Charlton:
  - http://staffwww.dcs.s olm/
  - http://www.dcs.shef.ac.uk/cgi-bin/ma Charlton
- Assignment and lab class help questions should be directed to the google discussion group

Feedback

- After each teaching week you MUST submit the lab register/feedback form
  - This records your engagement in the course
  - Ensures that I can see what you have understood and not understood
  - Allows us to revisit an her examples
  - This only works if you
- Submit this once you have finished b exercises
- Your feedback will be used to clarify topics which are assessed in the assignments
- Lab Register Link: https://goo.gl/0r73gD
- Additional feedback from assignment and MOLE quizzes

Machines Available

- Diamond Compute Labs
  - Visual Studio 2017
  - NVIDIA CUDA 9.1
- VAR Lab
  - CUDA enabled machines – same spec as Diamond high spec compute room
- ShARC
  - University of Sheffield H
  - You will need an account (see HPC docs w
  - Select number of GPU nodes available (se ing.shef.ac.uk)
  - Special short job queue will be made avail
- Your own machine
  - Must have a NVIDIA GPU for CUDA exercises
  - Virtual machines not an option
  - **IMPORTANT**: Follow the websites guidance for installing Visual Studio

Summary

- Parallelism is already here in a big way
    - From mobile to workstation to supercomputers
- Parallelism in hardware
    - It's the only way to use increasing number of transistors
    - Trend is for increasing

- Supercomputers
    - Increased dependency on accelerator
    - Accelerators are greener
- Software approaches
    - Shared and distributed memory models differ
    - Programs must be highly parallel to avoid diminishing returns