Administration details

Please see:
http://undergraduate.csse.uwa.edu.au/units/CITS3402/
for all details, including:

▶ lecture and lab times;
▶ links to the lecture recordings;
▶ links to the facebook page and YouTube channel;
▶ and resources;
We will make extensive use of available resources, in particular resources from www.openmp.org in the first part of the unit.

We will use Tim Mattson’s slides in the first part of the unit, interspersed with my own slides.

My slides will be in white (like this one) and Tim Mattson’s slides in blue background.

We may use slides from other sources, but we will always acknowledge the source in my slides.
Why High Performance Computing?

- Today’s computers are very fast.
- Most of the programs that you write, run almost instantaneously.
- But there are programs that take a long time to run.
- And people are trying to solve larger and larger problems as computers are becoming faster and faster.
- In fact some programs never complete execution, e.g., weather simulation (and forecasting) programs.
- The speed of hardware can never keep up with the demands of faster execution of programs.
The evolution of processors

- Every computer has a processor (usually an electronic chip), e.g., your desktop computer, laptop, mobile phone etc.
- Processors have become more and more powerful over the decades.
- However, they have become powerful by packing more and more electronic circuits within a small chip area.
- Electronic circuits dissipate power during their operations. Processors must be cooled for their correct operations.
- Also there are physical limits to how small the electronic circuit elements can be.
- Today’s processors have almost reached this limit.
The current trend is not to have a single powerful processor, rather several less powerful processors, so called *multicore* processors.

Each core is less powerful, but there are several of them.

The challenge is to make use of them for executing programs faster.

There are two related, but slightly different issues regarding this, *concurrency* and *parallelism*. 
Concurrency and Parallelism

- *Concurrency* is the simultaneous execution of multiple tasks.
- For example, you could be browsing the internet, writing a program and listening to music at the same time using your laptop.
- *Concurrency* could be *logical* or *physical*. Individual tasks can be executed on the same processor giving a logical concurrency, or each task may run on a different processor, giving physical concurrency.
Parallelism is breaking up a task into smaller tasks and solving these smaller tasks simultaneously.

Of course, the smaller tasks will give only partial results and we have to combine these partial results to get the final result.

Suppose you have to add 1000 numbers. You can either do this on your own, or recruit 10 friends each of whom adds 100 numbers at the same time.

You can then add these partial sums to get the final sum.

You can complete the task 10 times faster by taking help from 10 friends.
High performance computing is about parallelism

- Our aim in this unit is to study parallelism on machines that have multiple processors.
- There are many such machines, starting from your laptop, desktop, mobile phone, playstation, XBox and supercomputers.
- We will study two kinds of machines in this unit, multicore processors and cluster of workstations.
- The first one is readily available in your desktop or laptop computers, as almost all processors these days have more than one core.
- The second one is called a 'distributed memory' architecture. We will use the lab machines to create a cluster, that we will program like a parallel computer.