University of Toronto Engineering

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Module 8 – High-Performance Computing

Using the SciNet Supercomputer

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

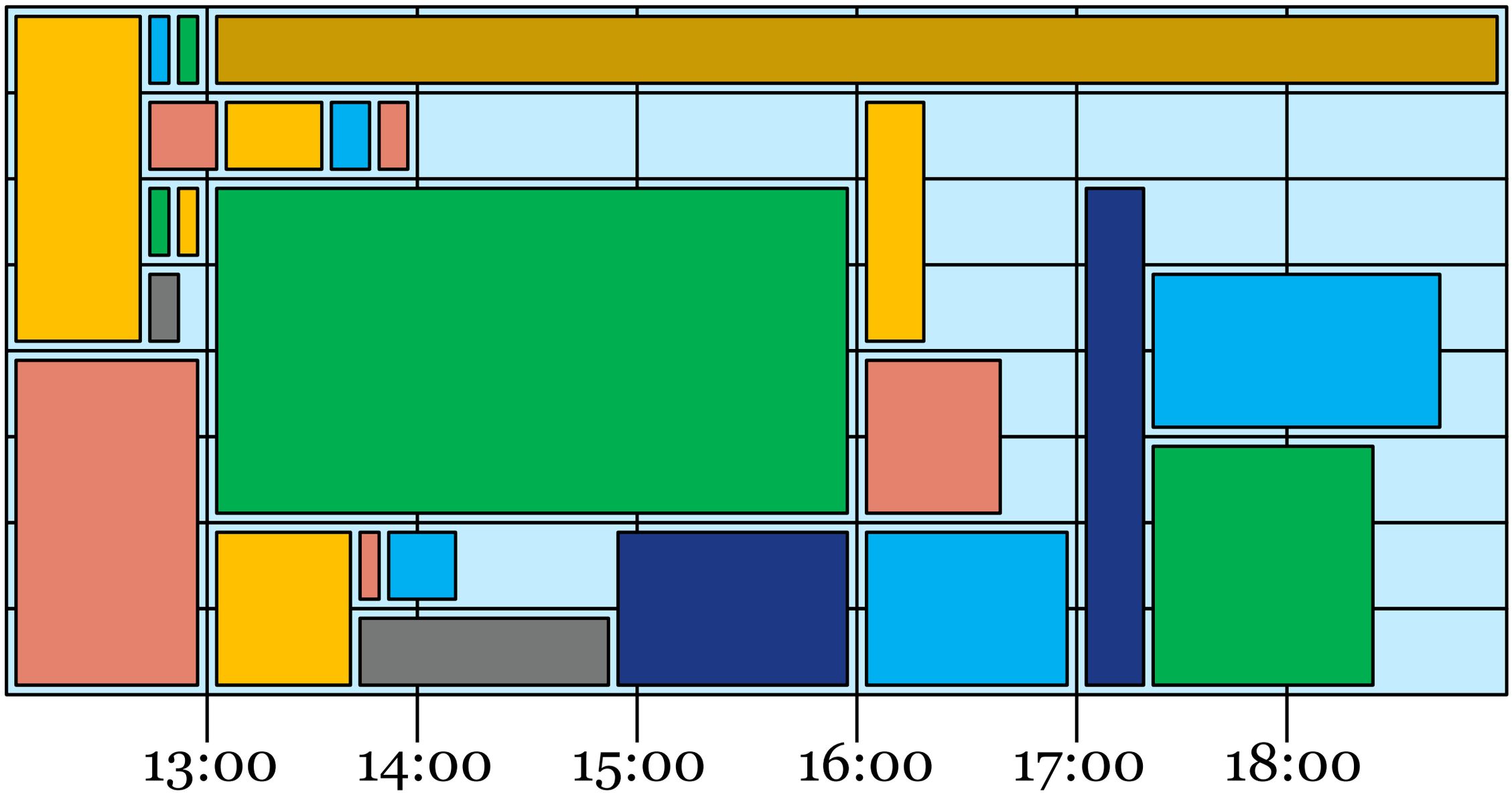
Module 8 – High-Performance Computing

What is a supercomputer?

- It's usually just a large number of smaller nodes
  - *These are connected using ethernet, infiniband or fibre-optic*
  - *Each node is like a very powerful desktop*
- Example specifications (SciNet):
  - *16GB RAM*
  - 1.4PB total permanent storage
  - 2.53GHz
  - 3,780 nodes
  - 30,912

Module 8 – High-Performance Computing

HPC Clusters share resources with jobs in a queue



Module 8 – High-Performance Computing

Once a job is running, how do we use multiple nodes?

- Each node also has 8-32 cores (processors). We should use these all too!
- To do this, we write parallel code
  - *Parallel code runs at the same time on different cores/nodes*

```
read_data()                    read_data()
c=compute_data()               c=compute_data()
list.append(c)                 list.append(c)
```

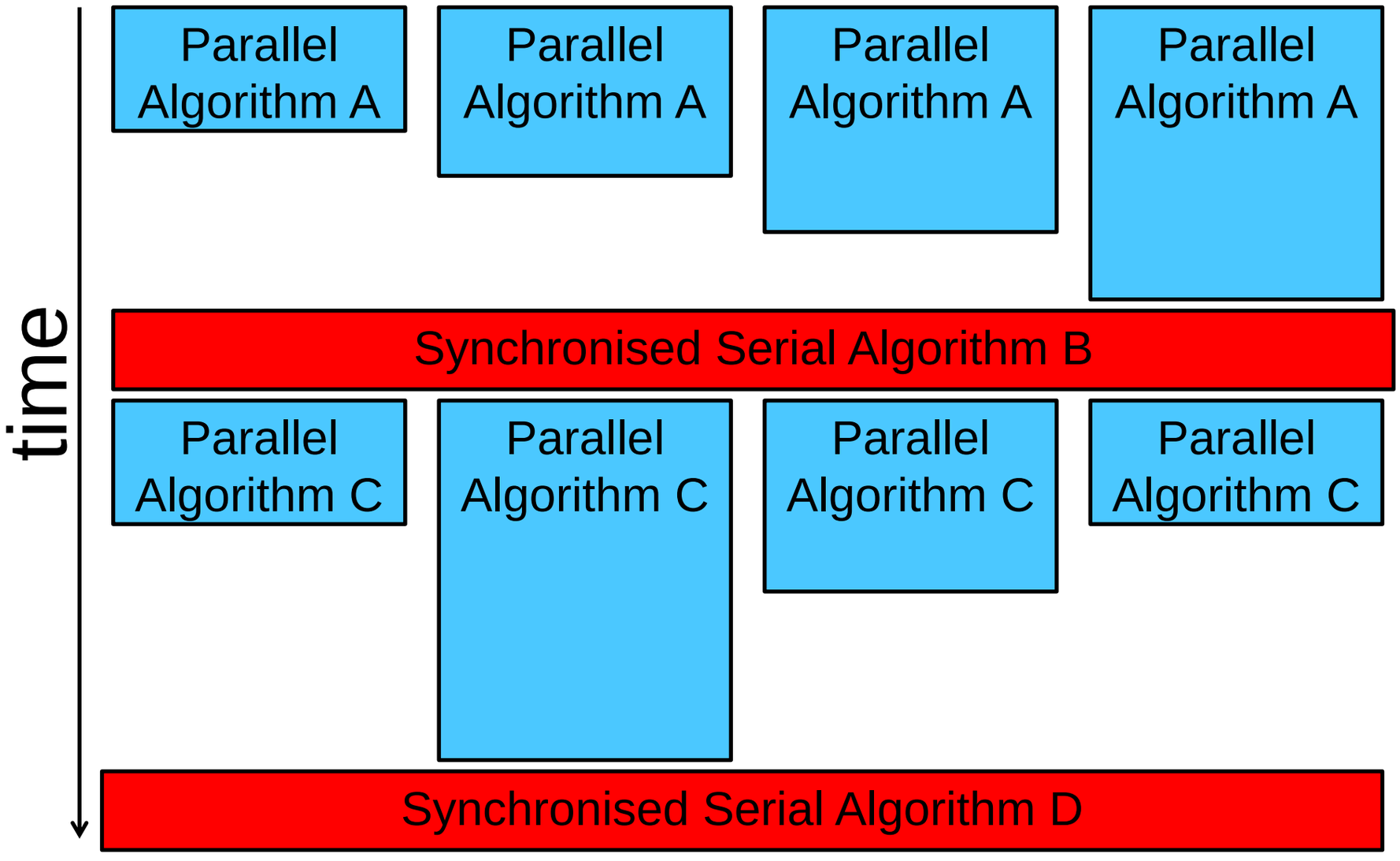
  - *Which one appends first?*
  - *We don't know: this is a race condition*
- Parallel code is counterintuitive!

Module 8 – High-Performance Computing

- An embarrassingly parallel algorithm can be parallelised perfectly
  - *Two people can make two sandwiches twice as fast as one person can make two*
  - *The sandwich making doesn't depend on the other sandwiches!*
- Most algorithms have a part that can be parallel, and a part that must be done sequentially
  - *Frequently the sequential part is aggregating data*
- *If* α represents the proportion (0-1) of time that must be sequestial, and p is the number of processors, then the time taken is $(\alpha+p*(1-\alpha))* t_s$
- The speedup is the serial time over the parallel time= $t_s / (\alpha+p*(1-\alpha))* t_s$
- As $p \rightarrow \infty$, the speedup$\rightarrow 1/\alpha$
  - For instance, if 20% of a programme is serial, we can never have better than 5x speedup!
- This is known as **Amdahl's law**

Module 8 – High-Performance Computing

We can also lose efficiency with synchronisation delays:



We will be using SciNet for these computations!

Start with an SSH client

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Module 8 – High-Performance Computing

To log in to SciNet:

- In PuTTY, enter the username and password you were given
  - *Once at the prompt, enter "ssh gpc01"*
  - *"module load python"*
  - *"python"*
- Write a simple script to perform a calculation
- Save as calc.py

Module 8 – High-Performance Computing

Now we're going to use the queueing system to run the script

- -l: number of nodes, processors per, walltime
- -q: the queue to use
- -I: use interactive mode

qsub -I -l nodes=2:ppn=8,walltime=1:00:00

qsub -l nodes=1:ppn=8,walltime=0:15:00 ./run.sh

Go ahead and run a job!

Module 8 – High-Performance Computing

We can check our jobs in the queue:

- showq>q
- showstart JOBID
- checkjob JOBID
- canceljob JOBID

Thanks!

Questions?

Email: b.graydon@ieee.org

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING