MUSINGS OF AN IDIOT

Adrian Jackson

a.jackson@epcc.ed.ac.uk

@adrianjhpc

epcc

BE NICE

BE KIND

FIND A TEAM

FIND A BALANCE

- Be wrong
- Ask questions
- Work on things you enjoy
- Work in public
- Seniority or money isn’t everything

PARALLEL COMPUTING

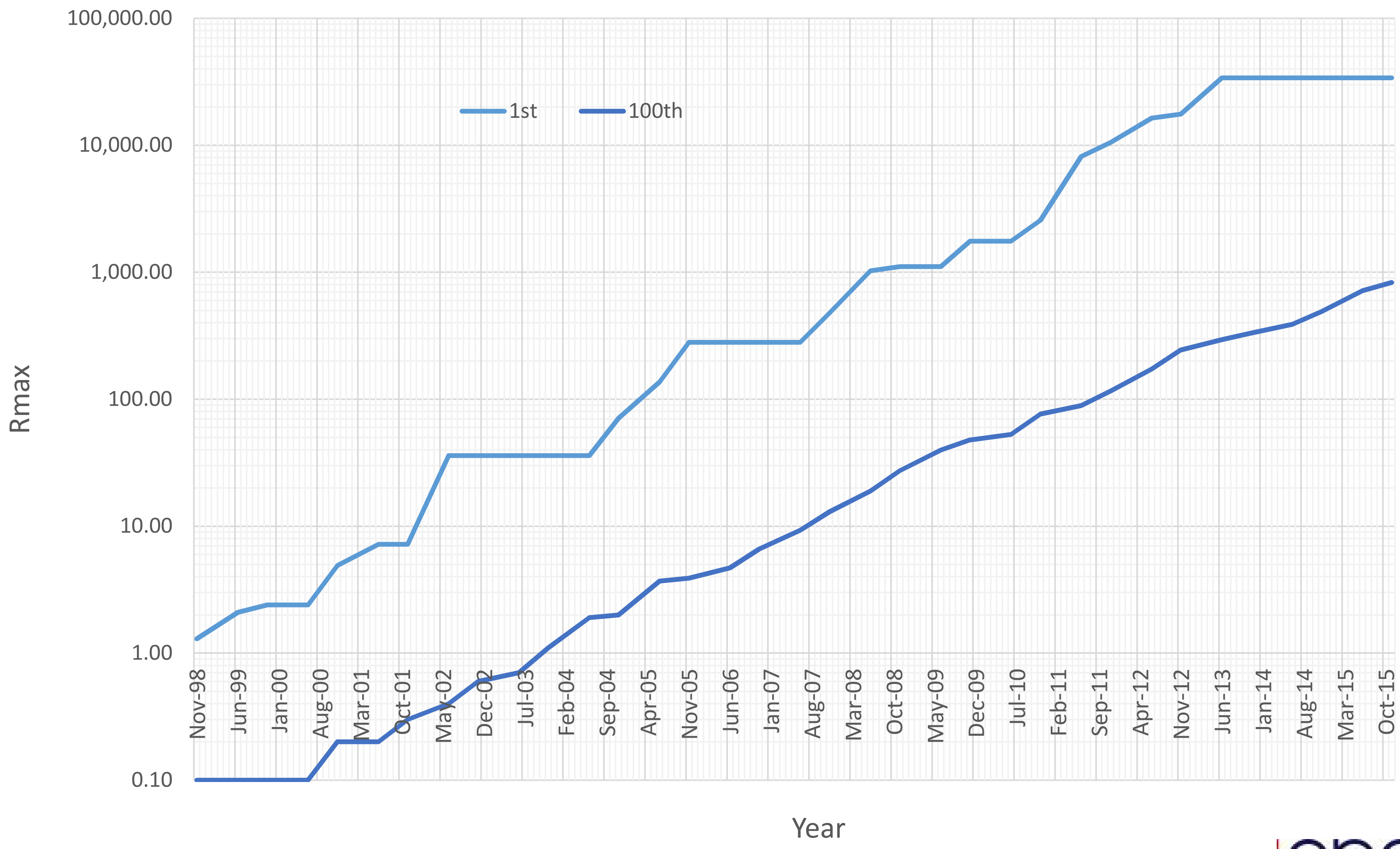
Adrian Jackson

adrianj@epcc.ed.ac.uk

@adrianjhpc

HPC - Parallelism

- Simulation science drives computing power
  - Consistently more computation power needed than available
  - Runtime of months on a single processor not uncommon
  - Parallel programs often start out as serial programs
- Why not just make a faster chip?
  - Theoretical problems
    - Physical limitations to size and speed of a single chip
    - Speed of light, size of atoms, dissipation of heat
    - Voltage reduction versus clock speed for power requirements
      - Voltages becomes too small for “digital” differences
  - Practical problems
    - Developing new chips is incredibly expensive
    - Must make maximum use of existing technology
- Solution
  - Use many CPUs cooperatively on the same problem
  - HPC computing synonymous with parallelism



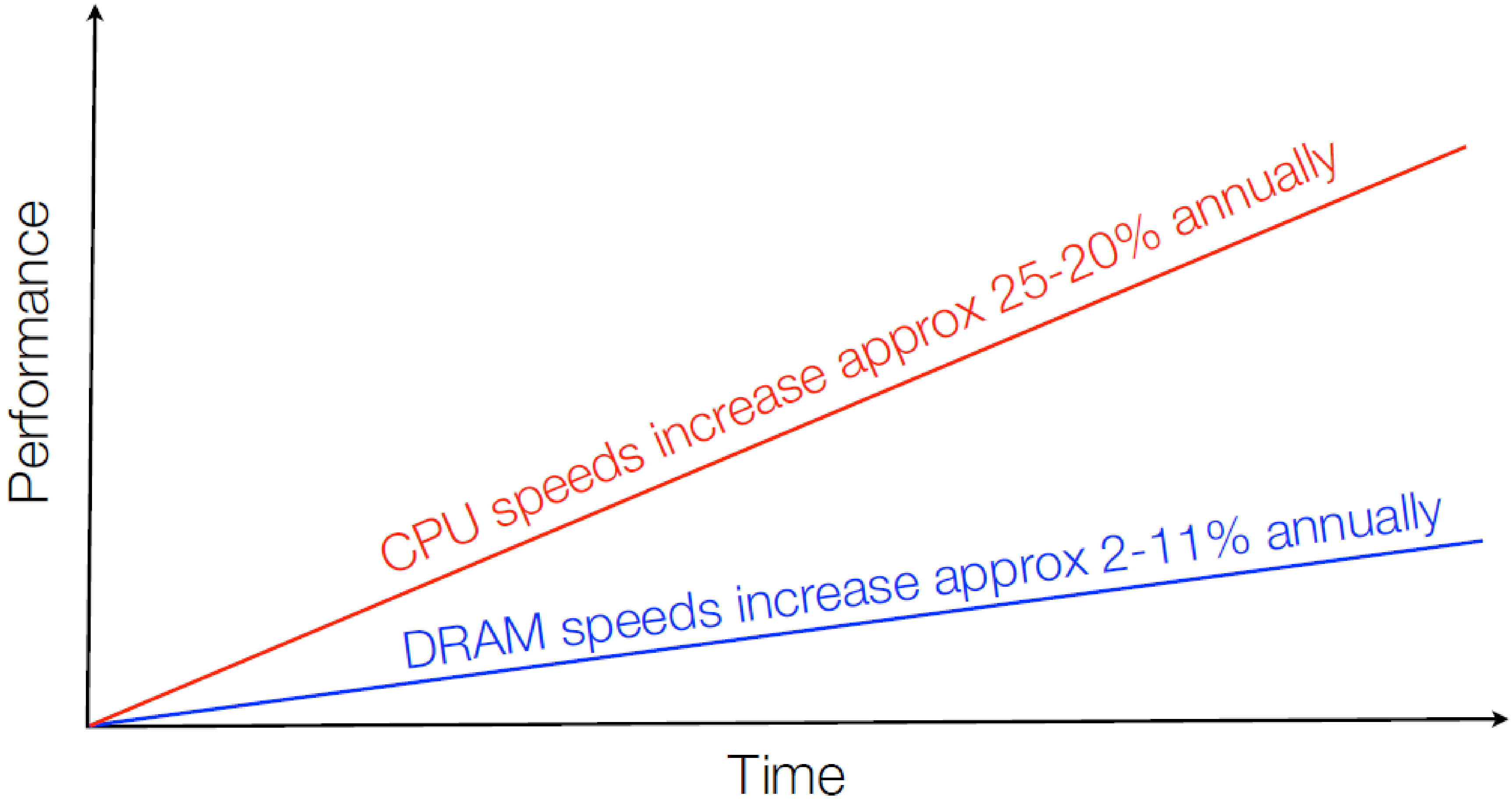
1st
100th
Rmax
100,000.00
10,000.00
1,000.00
100.00
10.00
1.00
0.10
Year
Nov-98
Jun-99
Jan-00
Aug-00
Mar-01
Oct-01
May-02
Dec-02
Jul-03
Feb-04
Sep-04
Apr-05
Nov-05
Jun-06
Jan-07
Aug-07
Mar-08
Oct-08
May-09
Dec-09
Jul-10
Feb-11
Sep-11
Apr-12
Nov-12
Jun-13
Jan-14
Aug-14
Mar-15
Oct-15

Building Blocks of Parallel Machines

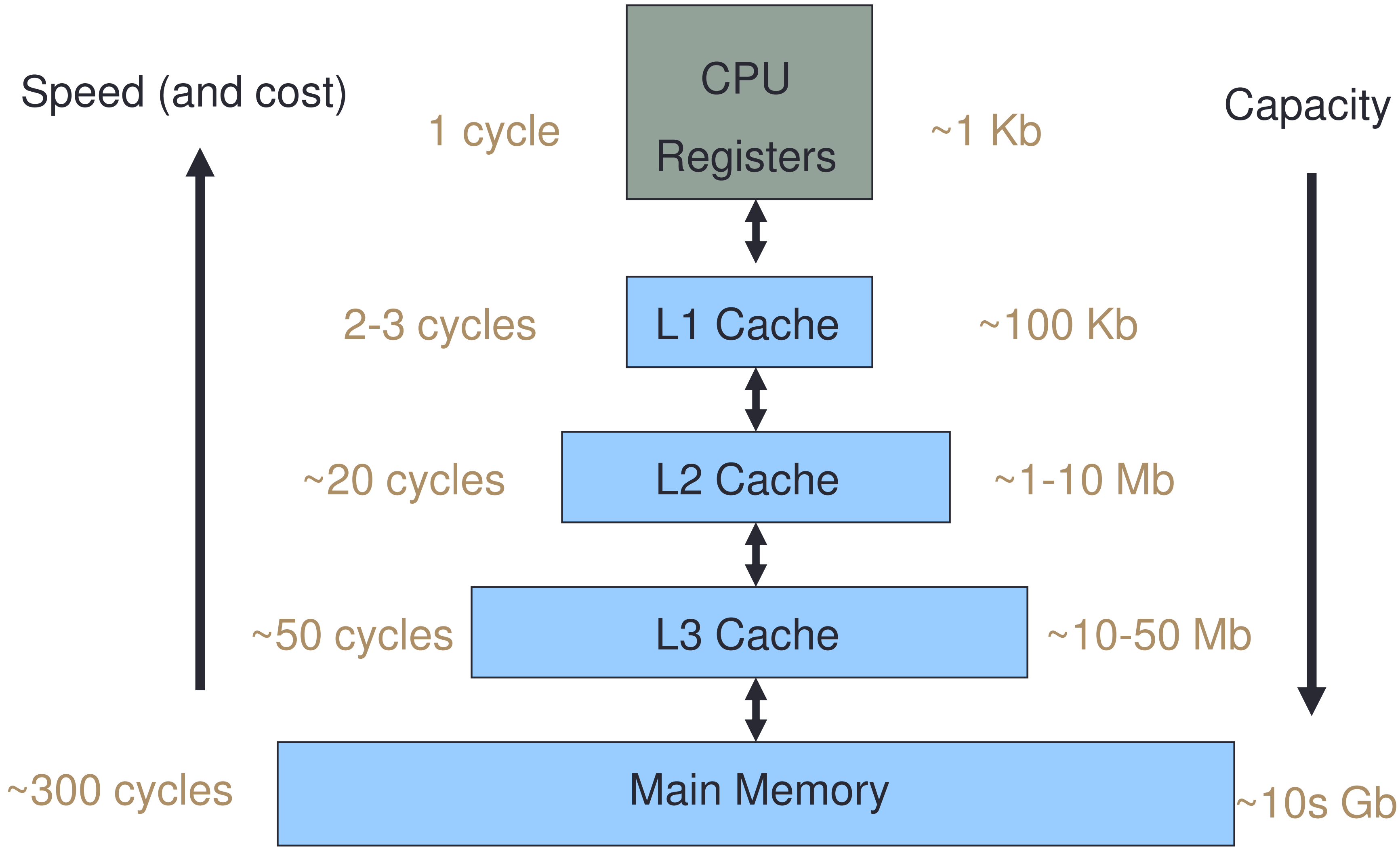
Four principal technologies which make up computers

- Processors
  - to calculate
- Memory
  - for temporary storage of data
- Interconnect
  - so processors can talk to each other (and the outside world)
- Storage
  - disks for storing input/output data and tapes for long term archiving of data

Performance Trend

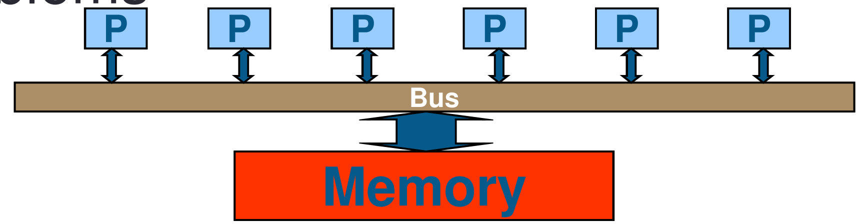


Memory hierarchy

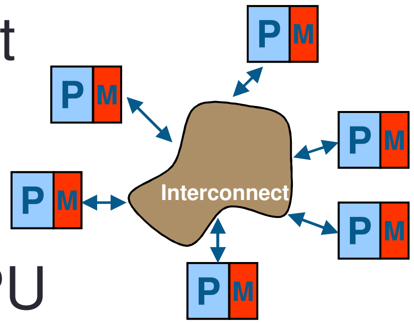


Types of HPC systems

- Shared-memory: OpenMP
  - Multiple processors share a single memory space
  - Simple to program for many problems
  - Scaling is problematic



- Distributed memory: MPI
  - Each processing unit has its own memory space
  - Excellent scaling properties
  - Can be more complex to program due to explicit communications



- Accelerators (GPU, Intel MIC)
  - Specialist processing units attached to main CPU
  - Can be difficult to extract good performance
  - (Conceptually similar to old vector architectures.)

Distributed Shared Memory (clusters)

- Dominant architecture is a hybrid of these two approaches: *Distributed Shared Memory.*
  - Due to most HPC systems being built from commodity hardware – trend to multicore processors.
  - Each Shared memory block is known as a *node*.
  - Usually 16-64 processors per node.
  - Nodes can also contain accelerators.
- Majority of users try to exploit in the same way as for a purely distributed machine
  - As the number of cores per node increases this can become increasingly inefficient…
  - …and programming for these machines can become increasingly complex

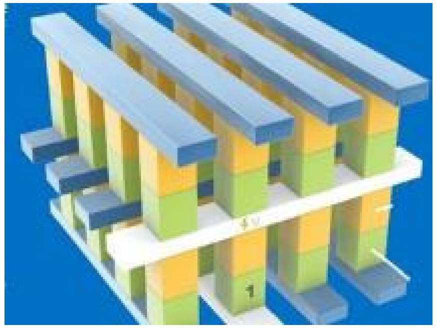
epcc

Differences from Cloud computing

- Performance
  - Clouds usually use virtual machines which add an extra layer of software.
  - In cloud you often share hardware resource with other users – HPC access is usually exclusive.
- Tight-coupling
  - HPC parallel programming usually assumes that the separate processes are tightly coupled
  - Requires a low-latency, high-bandwidth communication system between tasks
  - Cloud usually does not usually have this
- Programming models
  - HPC use high-level compiled languages with extensive optimisation.
  - Cloud often based on interpreted/JIT.

New hardware trends

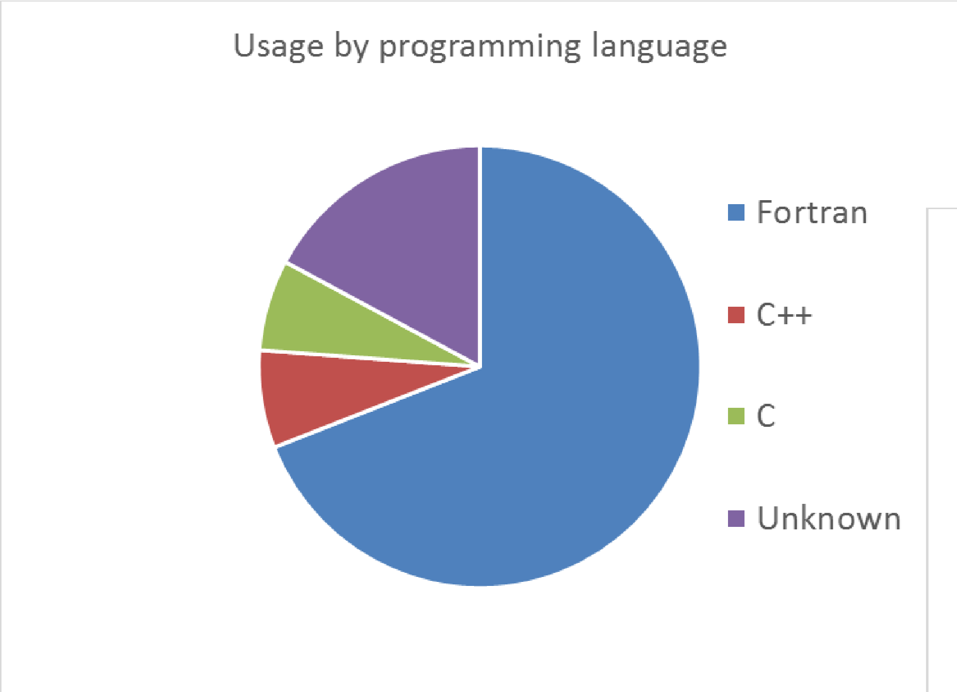
- Many-core
  - Intel KNL processors
    - 72 cores, 288 threads
  - Nvidia Pascal
    - Large scale GPU
- Memory and I/O becoming more complex
  - Different levels of memory
    - Stacked, DRAM, NVRAM
  - Different levels of I/O
    - NVRAM, SSDs, Lustre, /tmp

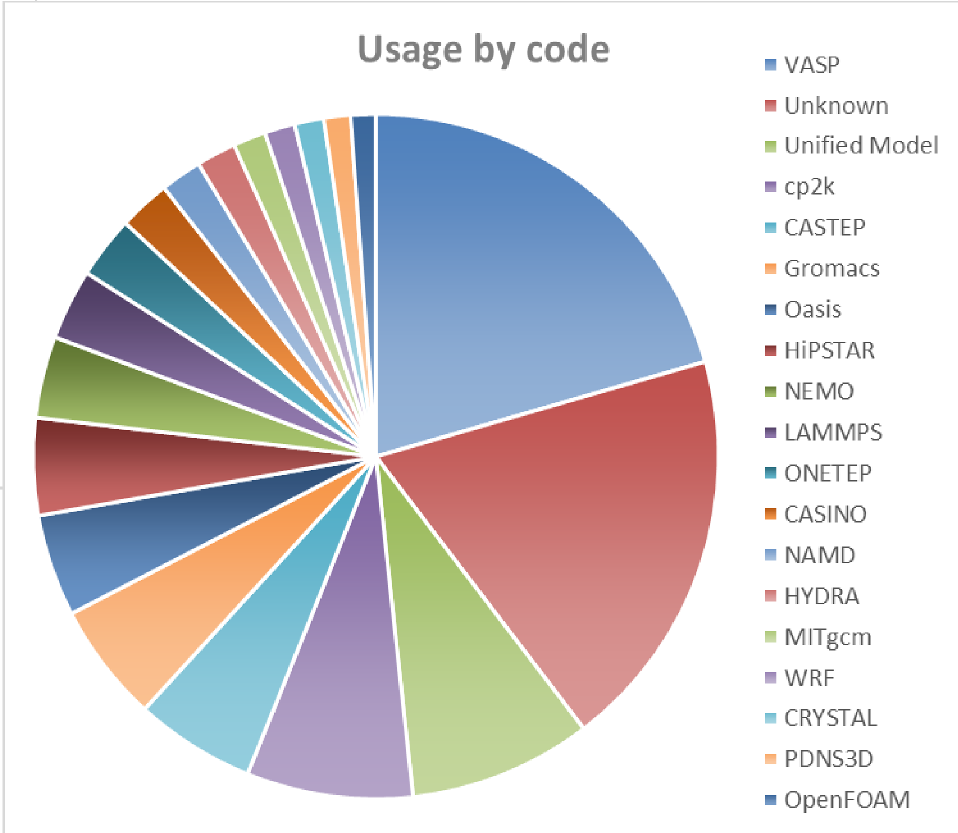


- More heterogeneity in both processing and memory systems
  - Applications are likely to have to adapt to get best performance

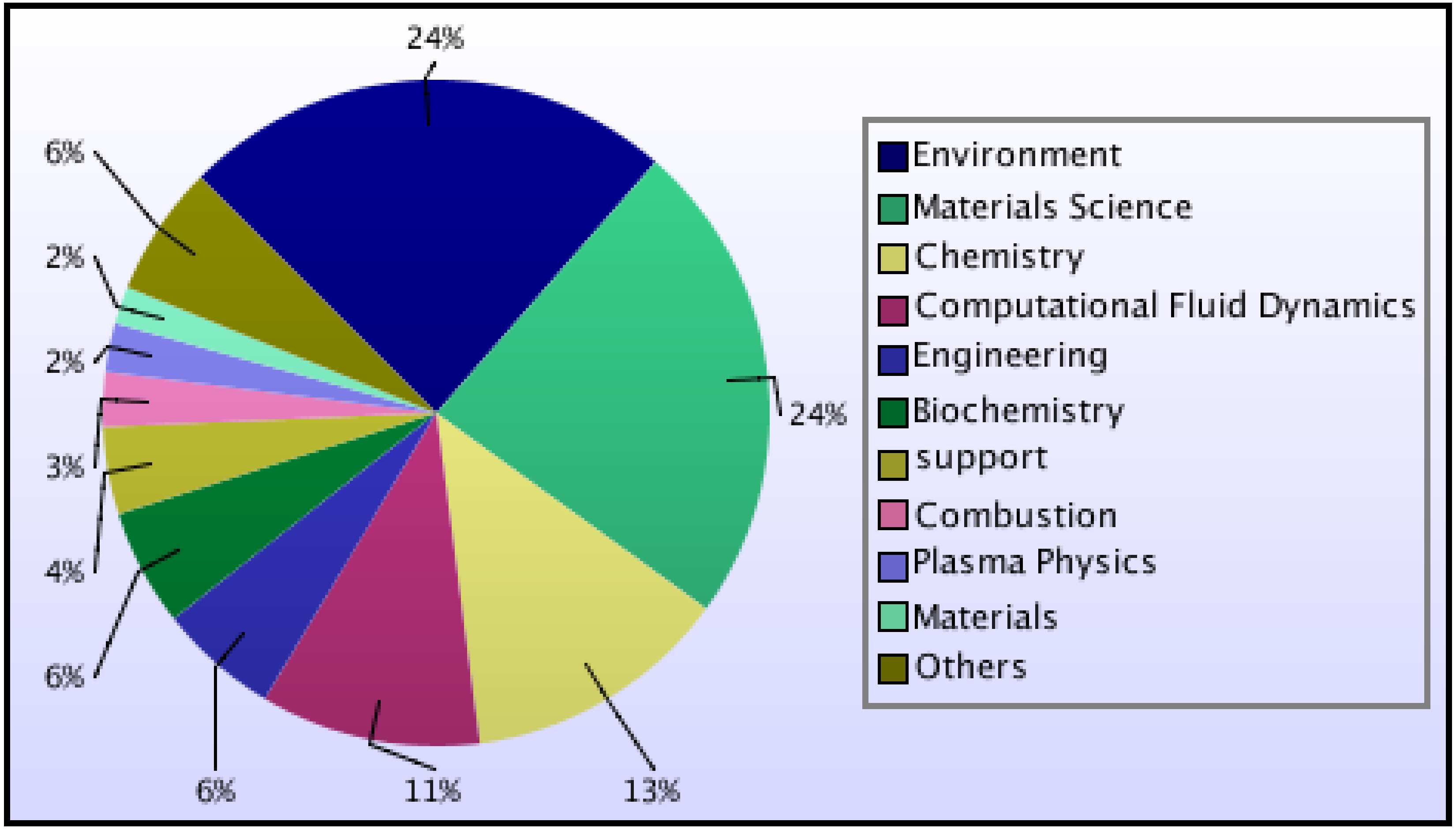
epcc

Usage by codes

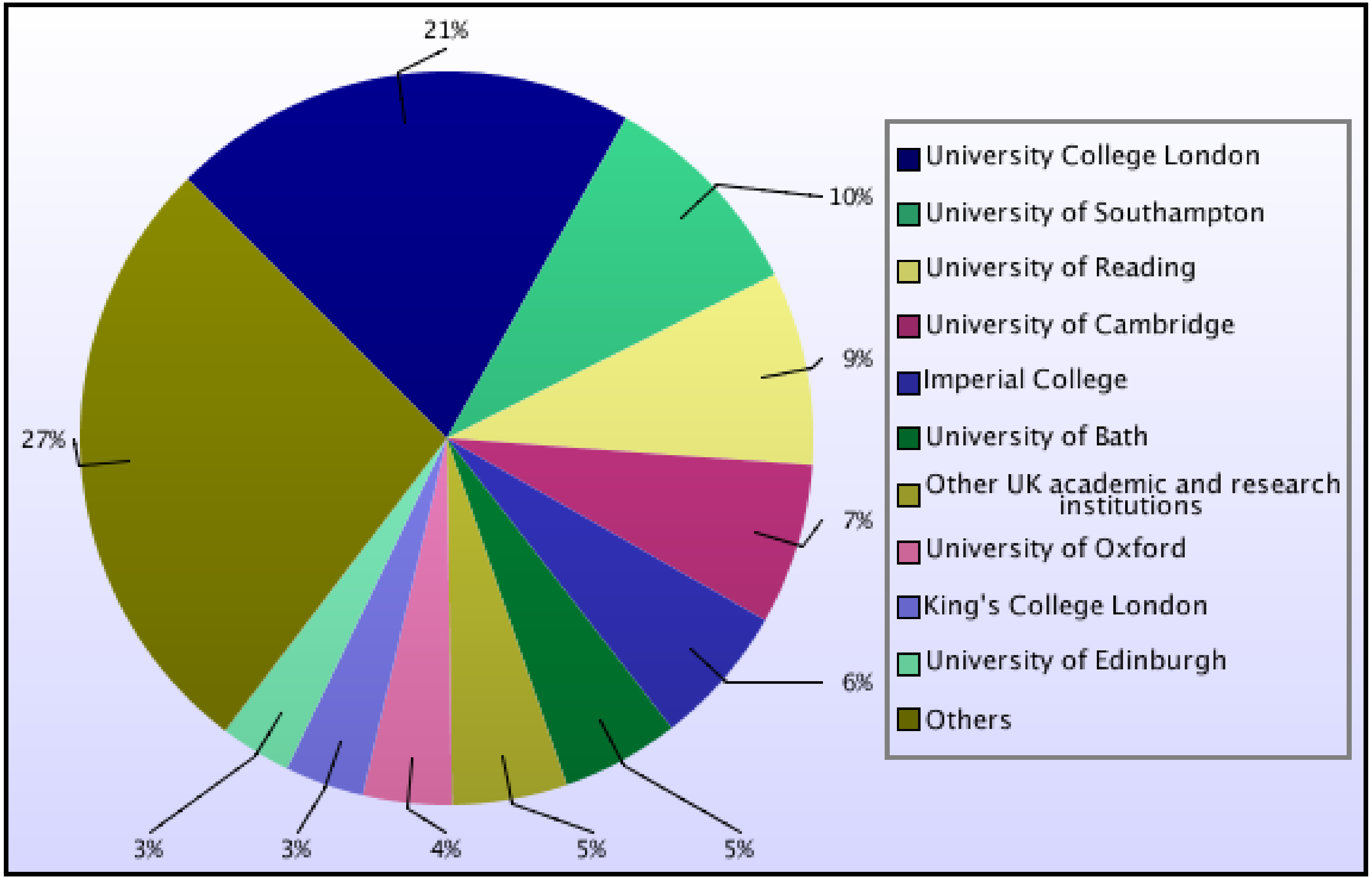




What is it used for?

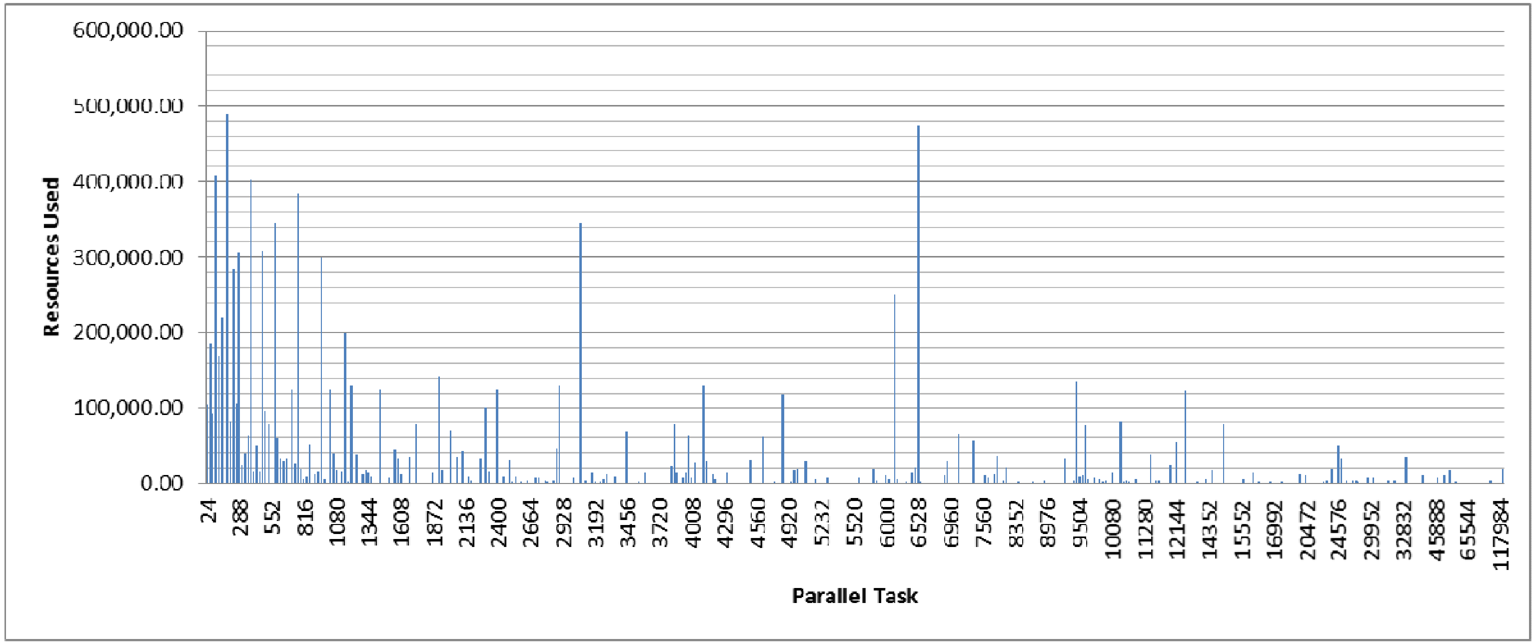


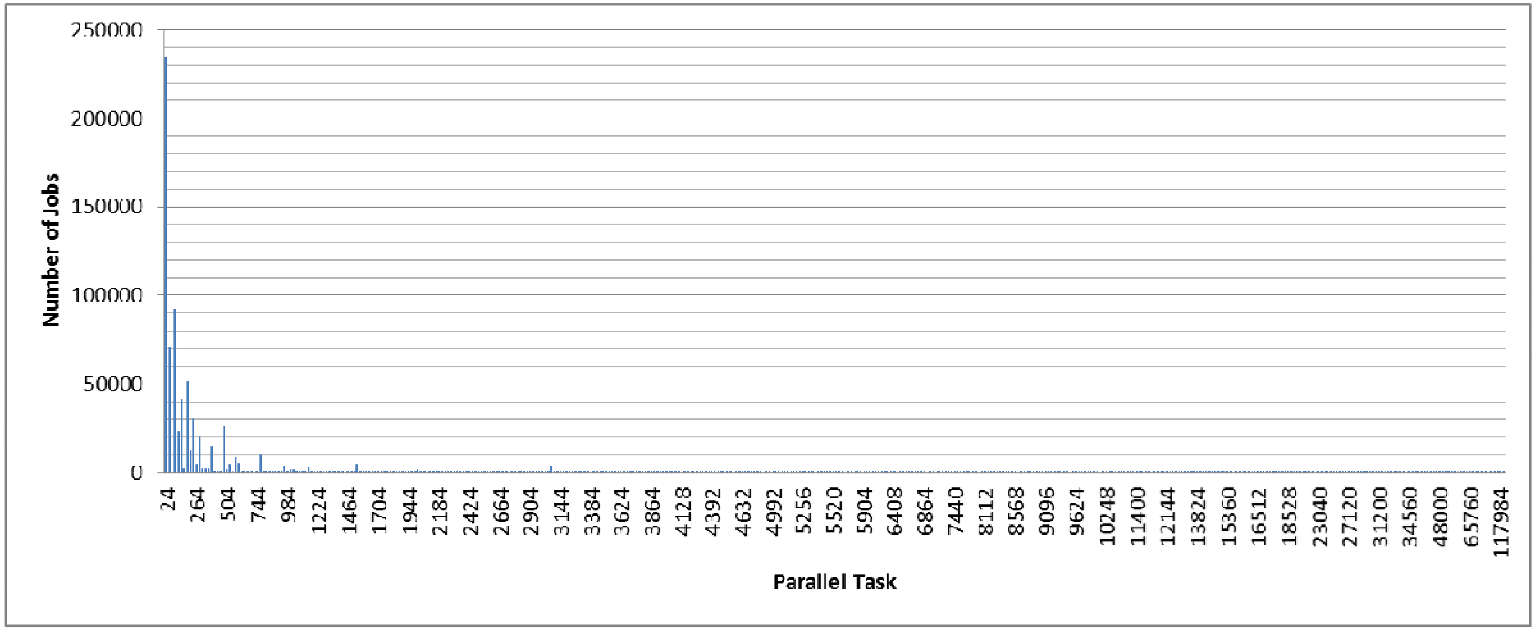
Institutional use



epcc

Machine Usage





What is a software developer?

- Coder
  - Writes code that compiles and runs
  - Inefficient, full of bugs, difficult to understand, fix and extend
- Programmer
  - Uses tools to develop correct code more easily, in less time, with less pain
  - Efficient, correct, modular, easy to understand, fix and extend
- Software developer
  - Gathers requirements
  - Creates designs and assesses alternatives
  - Writes, fixes, improves and extends programs
  - Writes user doc
  - Prepares releases
  - Trains users and provides support
- But what if my goal is to become a good researcher?

Programming skills and research

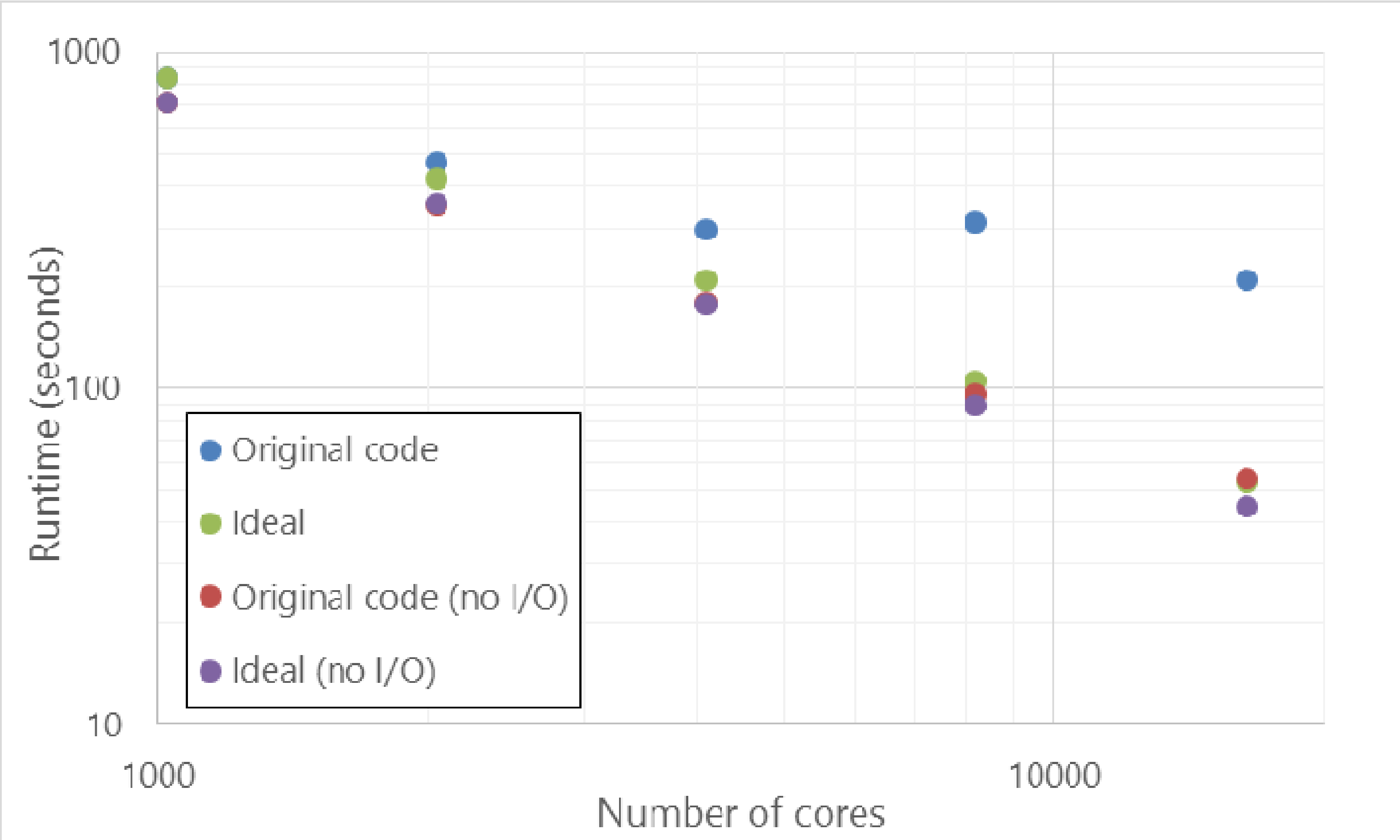
- Good programming skills are good research skills
- Record which version of your software produced the data you used in a paper, presentation, poster or thesis
  - Version control
- Help others to validate what you did by peer review, and to replicate, reproduce and reuse what you did
  - Good programming practice
- Ensure that your algorithms, implementation and data are correct and so that your research is correct
  - Unit testing, debugging, profiling
- Free up time for research
  - Automation and build tools, test frameworks

- Programming Language is irrelevant

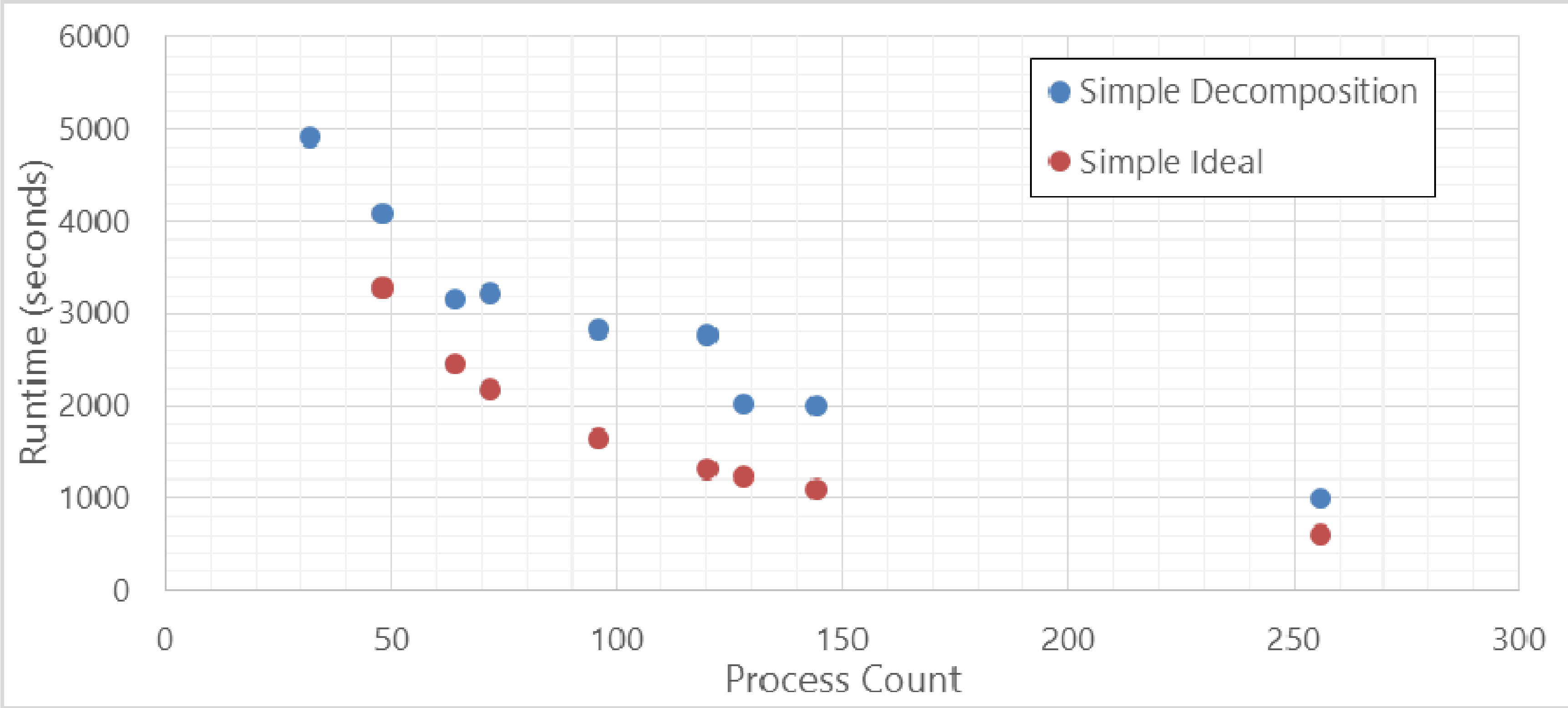
- Learn at least 2 languages

|epcc|

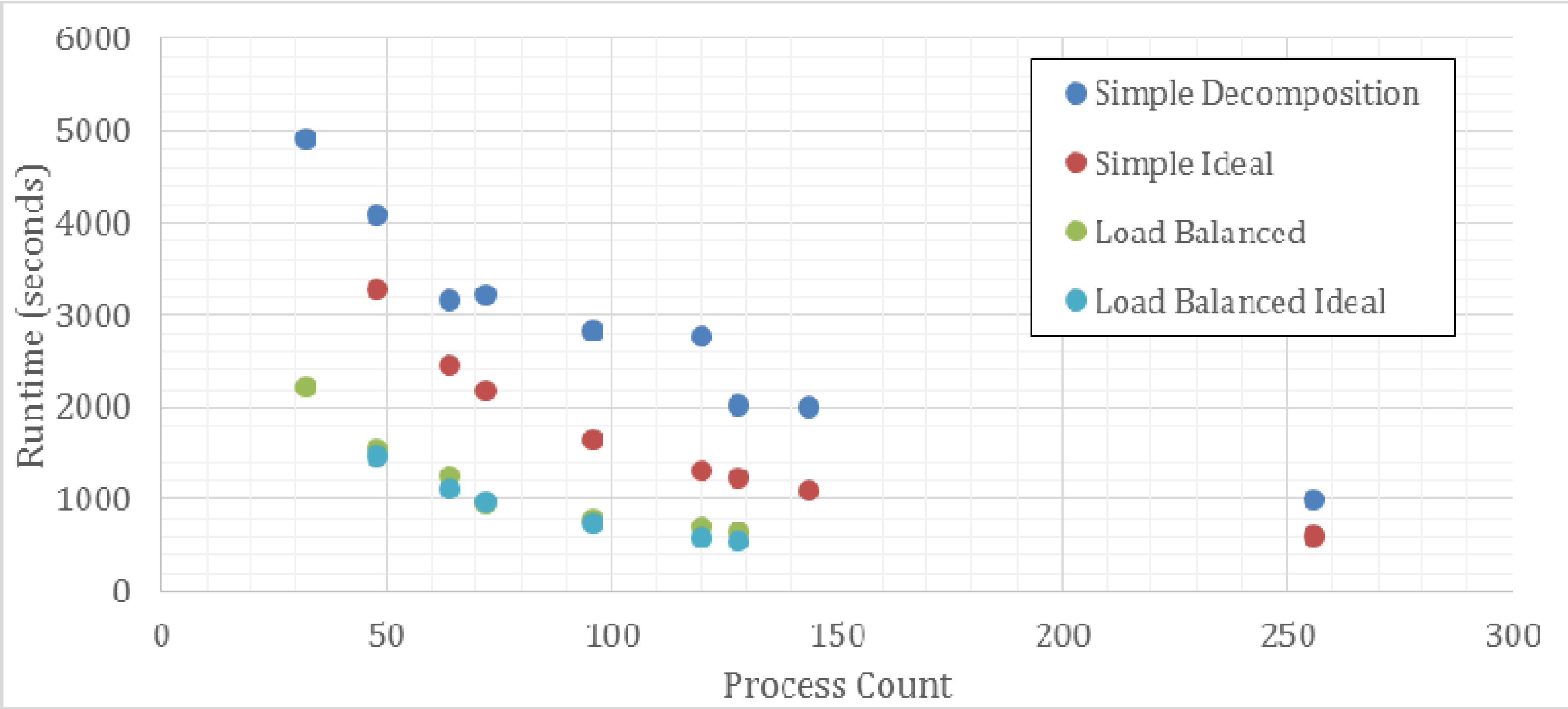
Fully science workflow is important



Fully science workflow is important



Improved workflow



- 256 block grid, high load imbalance
  - 90% difference in grid points between largest and smallest blocks

Summary

- Programs run on computers (doh!)
- Understanding computers helps understanding programs
  - Running and writing
- Programming takes effort
  - Learn more than one programming language
  - The more perspectives you have the better you'll be

What now?

- You can attempt the ARCHER driving test
  - www.archer.ac.uk/training/course-material/online/driving_test.php

- On successful completion, eligible users can apply for
  - account on ARCHER
  - 1,200 kAUs of time (80,000 core-hours) over 12 months

- Further information
  - Helpdesk: support@archer.ac.uk
  - Training: http://www.archer.ac.uk/training/.

- IPCC
  - https://www.epcc.ed.ac.uk/research/computing/intel-parallel-computing-centre

Women in HPC

- http://www.womeninhpc.org.uk

WHPC WOMEN IN HIGH PERFORMANCE COMPUTING

Working towards equal representation in HPC

Software sustainability institute

- http://www.software.ac.uk/

Software
Sustainability
Institute