Research Software Engineering enabling HPC

Anna Brown

WOMEN IN HIGH PERFORMANCE COMPUTING

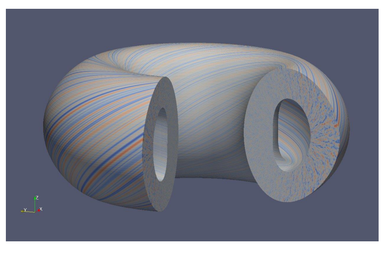
Why Research Software Engineering?

- Increasingly complex HPC architectures require dedicated expertise
- Increasingly complex codes require good software practice and time dedicated to maintenance
- An RSE combines SE expertise with an understanding of research to develop software in close collaboration with researchers

Join the community at RSE2019

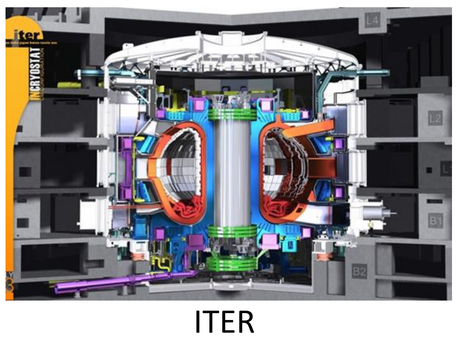
GS2 — Plasma physics simulation

GS2 is an open source Fortran 95 code for simulating turbulence in magnetized plasma on CPU, parallelised with MPI and scaling up to O(10k) cores.



Why HPC?

- The experimental fusion reactor ITER will have 10 times the plasma volume of the largest device currently in operation.
- Need to dramatically increase the performance of the existing GS2 code to model science at this scale.



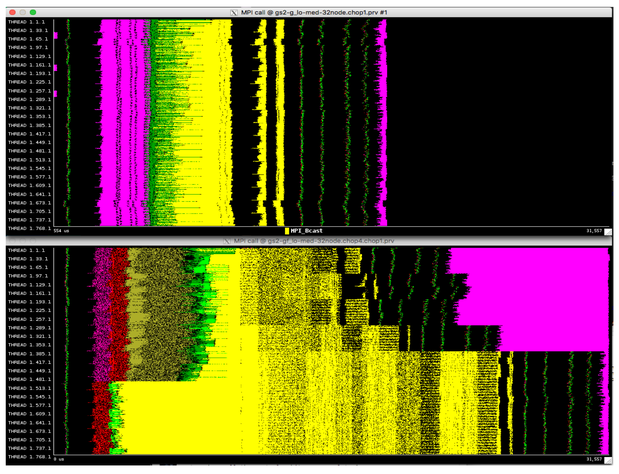
ITER

Optimising unavoidable MPI communication

- GS2 uses a distributed 7 dimensional array.
- Each timestep contains reduction steps across several dimensions, always hitting some data that is not local to a process
- Communication will always be the bottleneck here; needs to be efficient
- We used profiling tools to visualise communication patterns and bottlenecks – understood load imbalance, limiting factor

Profiling at scale

- Many communication bottlenecks only become apparent at scale
- Profiling at scale is complicated
- Special techniques, eg filtering large visualisation files and optimising parallel I/O.



Avoiding global communication

- The critical bottleneck was found to be in the duplication of a distributed data structure onto a single process for verification purposes.
- Little impact at small process counts
- A typical use case is now at the scale where this global communication has a significant impact.
- Initial tests on ~3k cores suggest that removing this bottleneck could lead to a run time reduction of 20%.

Collaborators

Joseph Parker

Science & Technology Facilities Council
Rutherford Appleton Laboratory

Colin Roach

CCFE
CULHAM CENTRE FOR FUSION ENERGY

Sally Bridgwater

nag

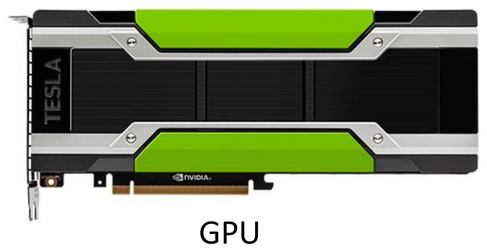
QuEST — Quantum computing simulation

QuEST is an open source library for simulating quantum computing on classical computers, written in C and parallelised with OpenMP across a single CPU node, MPI across multiple nodes and CUDA on GPU.

Why HPC?

- Memory requirements double with each additional qubit.
- Need to simulate large systems to verify real quantum computers are working correctly
- To simulate just 40 qubits takes 32 TB of RAM.
- Run time for a single quantum doubles with each additional qubit.
- Need to simulate thousands of small systems when modelling error in quantum algorithms.
- Each individual simulation needs to run fast.

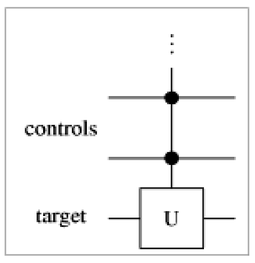
Archer national supercomputer



GPU

Hiding HPC complexity from users

- Users need access to HPC without detailed knowledge of the architecture.
- QuEST uses the same API for circuits running on single CPU, distributed across multiple CPUs and on GPU
- Users can easily scale up code developed on a laptop.
- Needs learnt through close collaboration with domain experts



Collaborators

Simon Benjamin, Tyson Jones

OxfordMaterials

S
B

Niel de Beaudrap

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE
NQIT Networked Quantum Information Technologies

Our team

Ian Bush

Jacob Wilkins

Anna Brown

EPSRC
Engineering and Physical Sciences Research Council

Funding: EP/N019067/1
Software Engineering - In Support of the Exascale

DEPARTMENT OF ENGINEERING SCIENCE
Oxford e-Research Centre
UNIVERSITY OF OXFORD