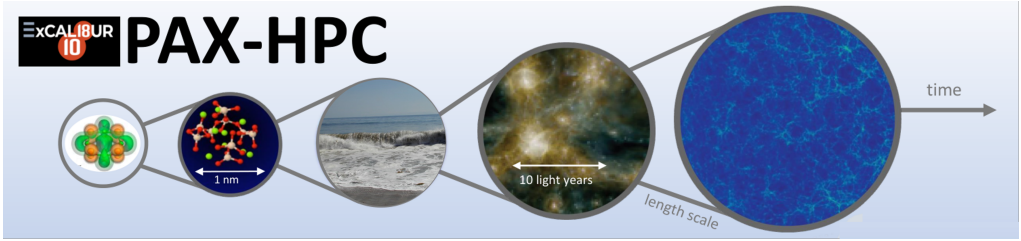
Enabling Exascale First-Principles Materials Simulations with the PAX Project

Phil Hasnip

PASC23
June 2023

Particles At eXascale



- Focuses on particle-based methods
- Smoothed Particle Hydrodynamics
  - Cosmology
  - Engineering
- Materials modelling
  - Condensed matter physics
  - Quantum chemistry
  - Materials Science

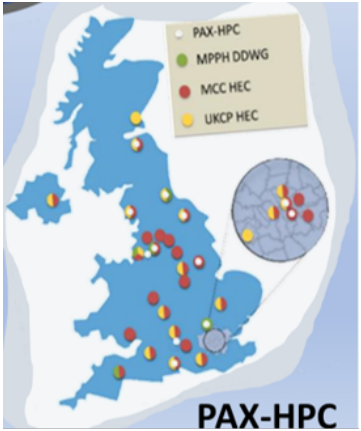
Bringing communities together

UK partners:

- Massively-Parallel Particle Hydrodynamics
- Materials Chemistry Consortium
- UK Car-Parrinello Consortium

Plus:

- Universiteit Leiden (Netherlands)
- Duke University (USA)
- NVIDIA
- Intel



Useful HPC

Want to use high-performance *computing* to deliver high-performance *science*.

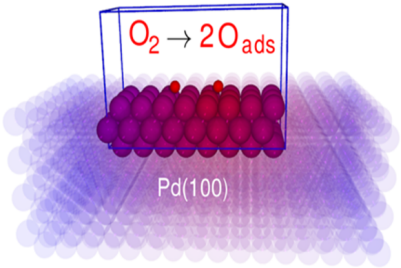
Three classes of studies identified:

- Complex workflows (“code coupling”)
  - Multiscale
  - Multiphysics
- Hero calculations (“capability”)
  - Few large calculations
  - Tightly coupled
  - Fine-grained parallelism
- High-throughput (“capacity”)
  - Many medium-sized calculations
  - Loosely coupled
  - Coarse-grained parallelism

Complex workflows

Many different classes of models across 12 codes

- Classical forcefields (DL_POLY)
- First-principles modelling
  - Quantum Monte Carlo (CASINO)
  - GW (Questaal)
  - DFT (CASTEP, CP2K, CRYSTAL)
- ChemShell can couple codes for QM/MM
  - Extending range of compute engines
  - Enabling multiple QM regions
  - Integrating ESCDF for exascale I/O
- Helped by a standardised API: bitbucket.org/byornski/dft-python-api



O_2 dissociation on Pd

First-Principles Modelling

Most common approach is Density Functional Theory (DFT); solve a form of Schrödinger equation:

$$\hat{H}\psi_{bk} = E_{bk}\psi_{bk}$$

Main basis set choices to represent ψ_{bk}:

- Local basis set
  - Hamiltonian is compact
  - Constructing Hamiltonian resource-intensive
  - E.g. CRYSTAL
- Plane waves
  - Hamiltonian is large
  - Constructing Hamiltonian is simple
  - E.g. CASTEP
- Dynamic
  - Switch between them
  - E.g. CP2K

Exascale HPC

Materials modelling software typically based on:

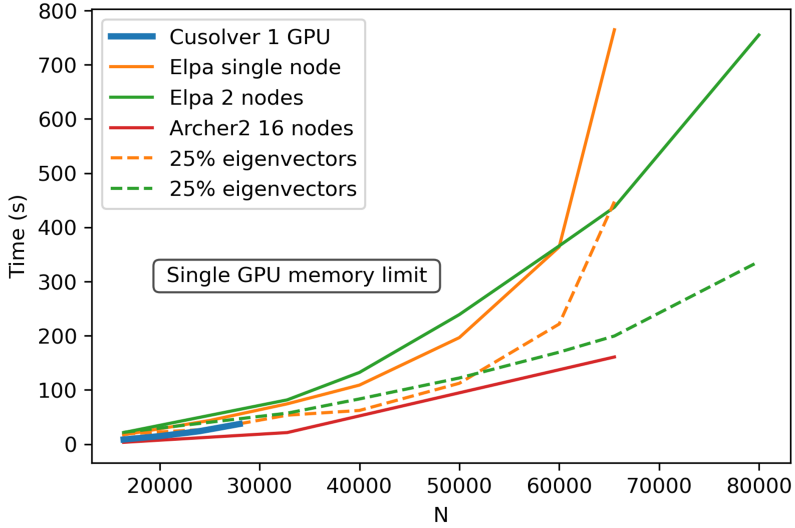
- Modern Fortran
- MPI
- OpenMP

Exascale machines characterised by:

- GPUs
- Massive parallelism

GPU kernels

Matrix diagonalisation is a key kernel.



Plane-waves

For plane-wave-based software (e.g. CASTEP), the Hamiltonian is simple to apply.

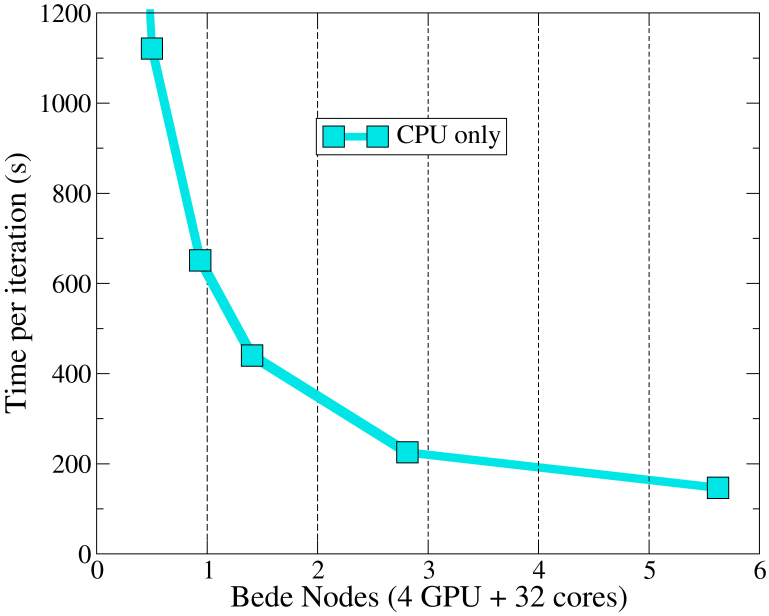
We can use standard BLAS and LAPACK routines, with optimised GPU libraries.

Fast Fourier Transforms are also key kernels, but optimised GPU libraries already exist.

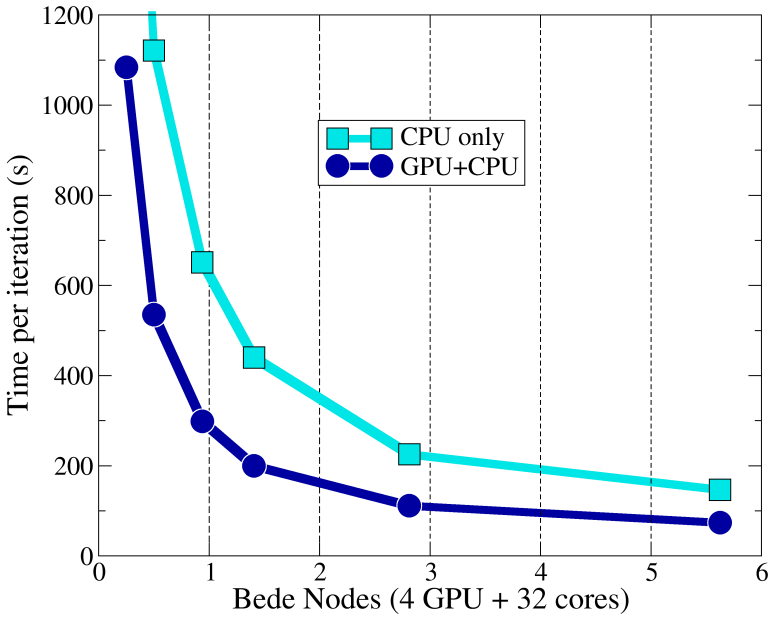
Focus on:

- Single code base
- Directives-based data movement (OpenACC)
- Use of optimised GPU libraries

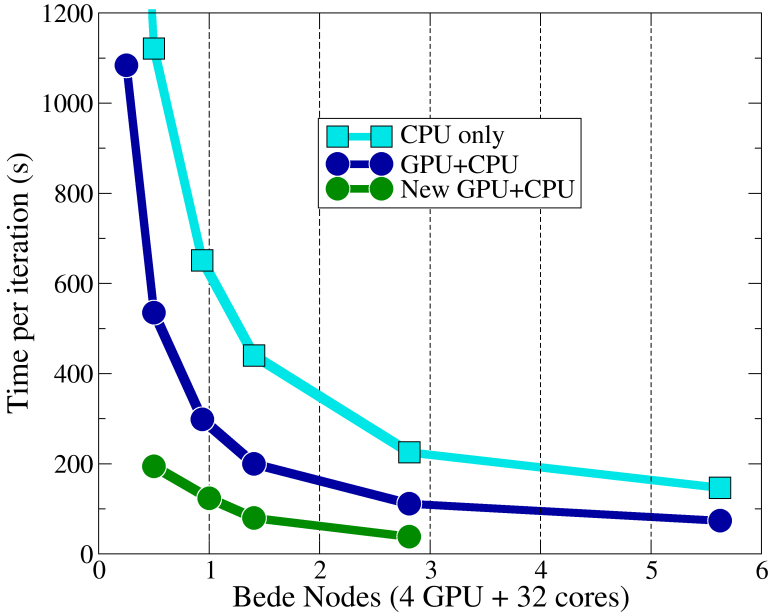
CASTEP-GPU on Bede (UK Tier-2 HPC)



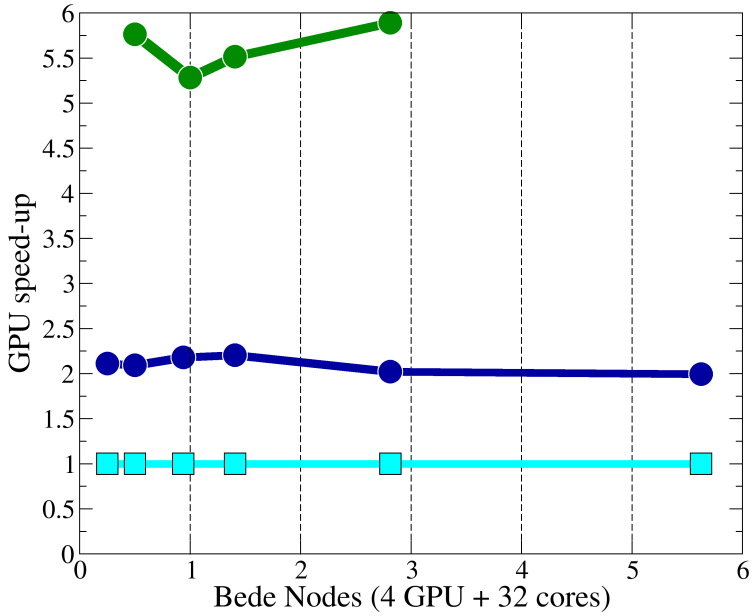
CASTEP-GPU on Bede (UK Tier-2 HPC)



CASTEP-GPU on Bede (UK Tier-2 HPC)



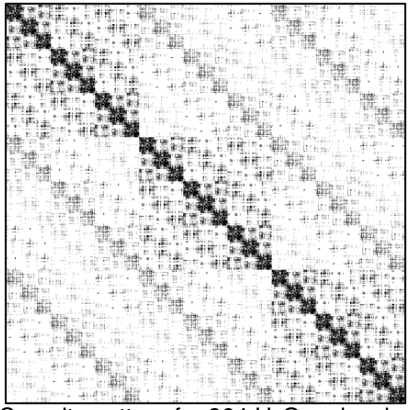
CASTEP-GPU on Bede (UK Tier-2 HPC)



Local basis set

Creating the Hamiltonian matrix is complicated in local basis set codes (e.g. CP2K, CRYSTAL), and GPU acceleration is not trivial.

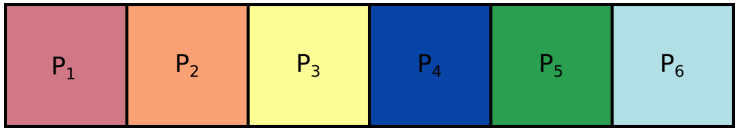
- Matrices are sparse; can we exploit that?
- Experimenting with test matrices
- Testing with CRYSTAL and CP2K
- Offloading 2-electron integrals to GPUs
- Task-based parallelism



Sparsity pattern for 864 H_2O molecules

Plane-wave parallelism problems

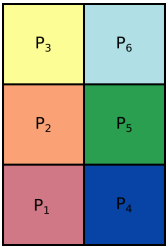
- For plane-wave DFT, the parallel FFT limits the scaling
- FFTs need all-to-all comms
- For P processes, comms time scales as P^2
- Need to re-think the data distribution...



$O(P^2)$

Rethinking the distribution

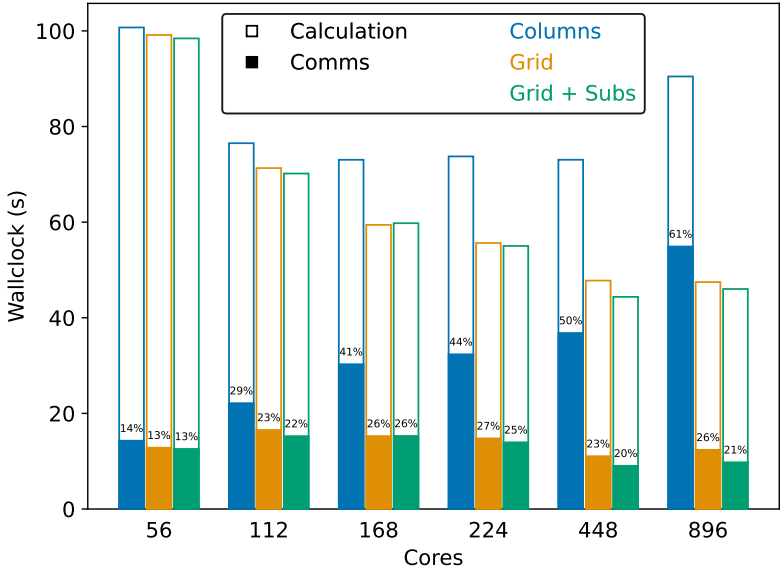
- Key idea: arrange processes in a logical process grid, side $\sim \sqrt{P}$
- Each data transposition only involves processes in either the same process row or column



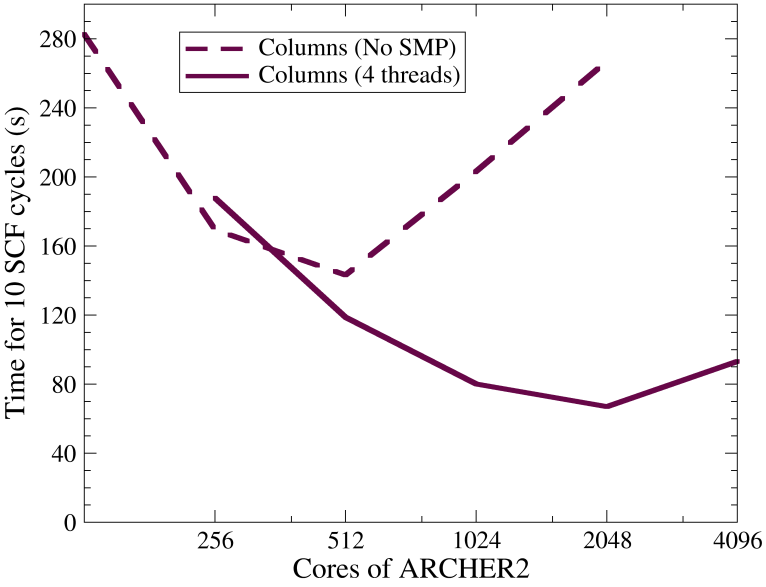
$O(P)$

- Restrict communications to sub-communicator rows and columns of process grid

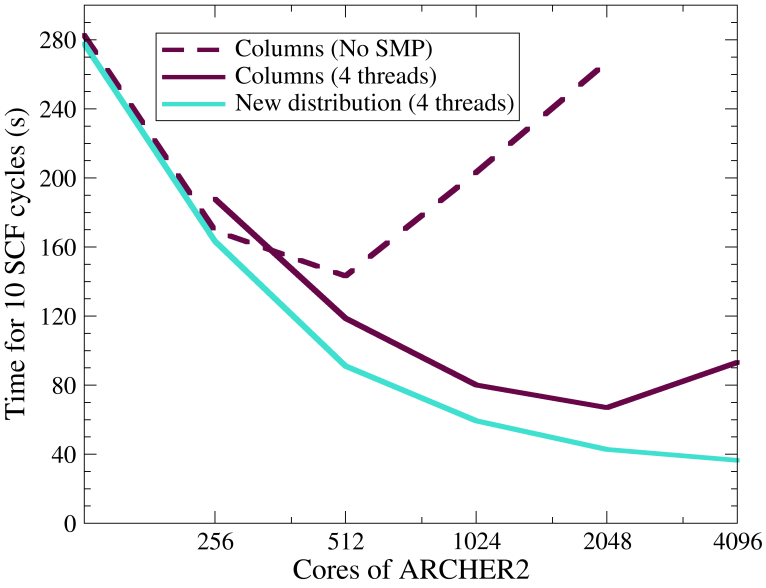
Performance: Time (Solid benzene on CSD3)



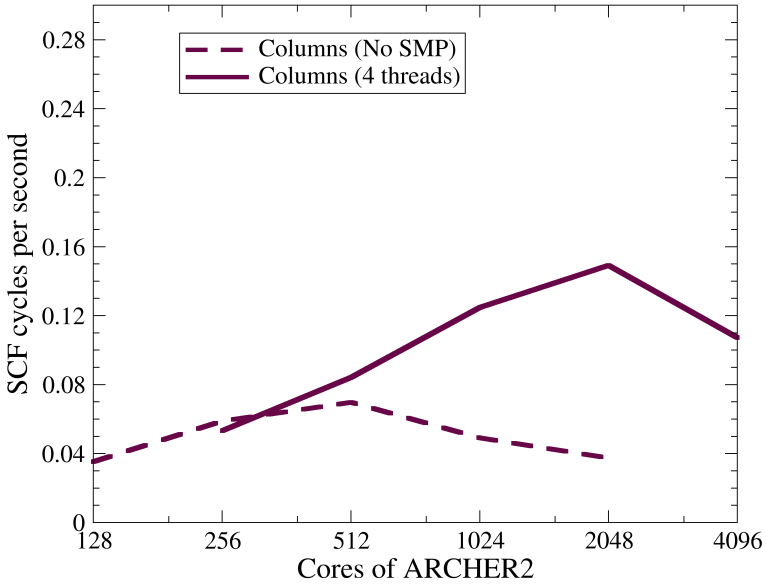
Performance: Time (Al_2O_3 on ARCHER2)



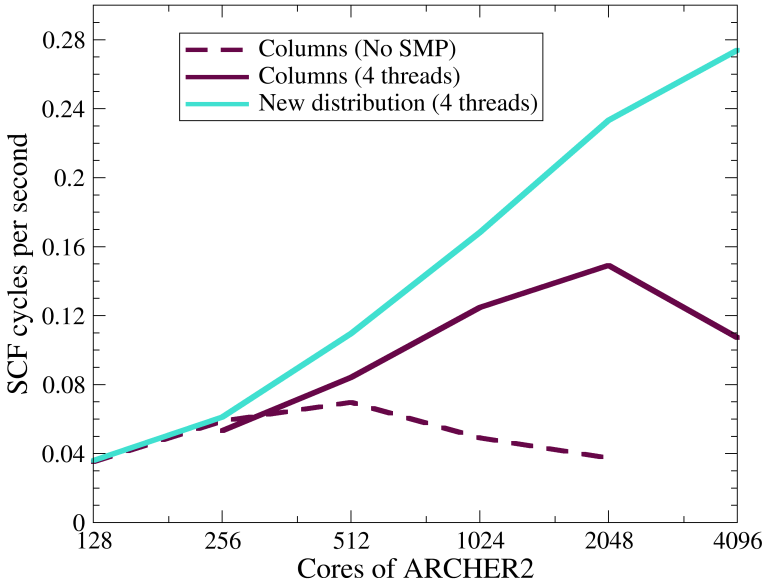
Performance: Time (Al_2O_3 on ARCHER2)



Performance: Speed (Al_2O_3 on ARCHER2)



Performance: Speed (Al_2O_3 on ARCHER2)



Re-imagined parallelism

- Emphasis on scaling (improved 4x)
- Faster on all core counts
- Gives a 1.6x speed-up on 2048 cores
  $\longrightarrow$ 63% of the time-to-science

Conclusions

- PAX is an interdisciplinary collaborative initiative
- Working with 12 materials simulation codes
- Investigating novel hardware
- Porting and optimising for GPUs
- Re-thinking parallel decompositions
- Many challenges...
  but already seeing significant performance improvements!

Exascale materials modelling:
T. Keal et al, *Comput. Sci. Eng.* **24**(1) 36-45 (Jan-Feb 2022); doi: 10.1109/MCSE.2022.3141328
CASTEP GPU:
M.J. Smith et al, *Comput. Sci. Eng.* **24**(1) 46-55 (Jan-Feb 2022); doi: 10.1109/MCSE.2022.3141714

We are grateful to EPSRC for funding PAX, and additional funding from the ARCHER2 eCSE programme.