Science and Technology Facilities Council

Acceleration of Electronic Structure Codes on Heterogeneous Hardware

Marcello Puligheddu, Ian J. Bush
Science and Technology Facilities Council

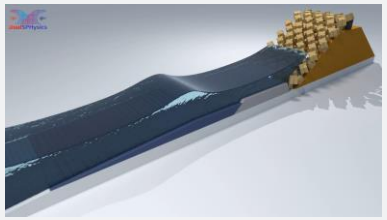
CoSeC Conference 2023, Manchester Central Convention Complex
Wednesday 6 December

EXCALIBUR 10

PAX-HPC

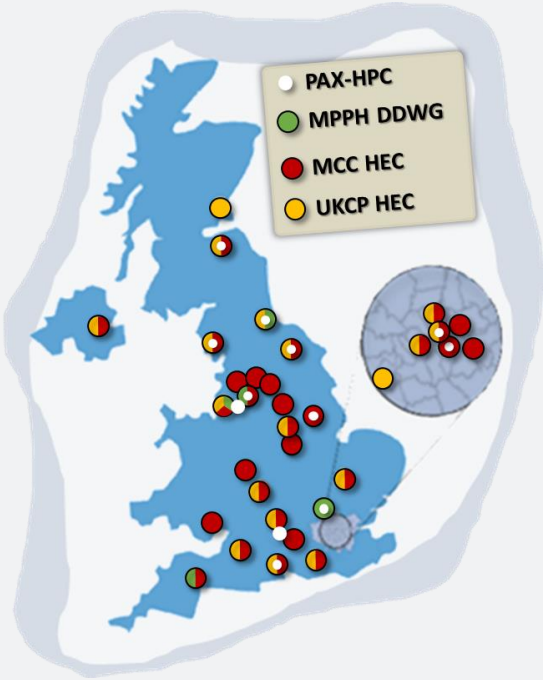
{MMM DDWG
MPPH DDWG

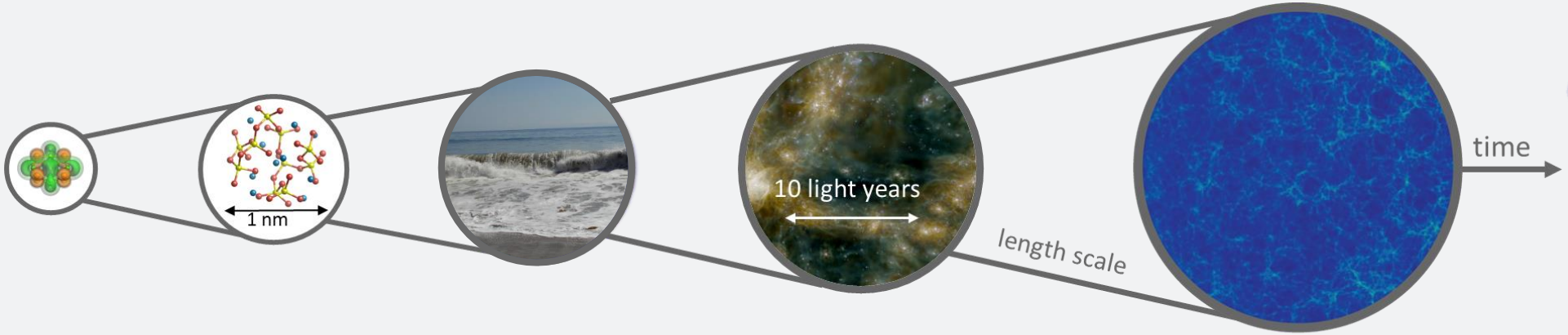
Particles at the eXascale on HPC



Keggeris et al. (2019)

Altomare et al. (2014)





CCP-QC UKCP CSCS Centro Svizzero di Calcolo Scientifico Swiss National Supercomputing Centre DiRAC nag MATERIALS AND MOLECULAR MODELLING HUB NVIDIA NREL Universiteit Leiden Duke UNIVERSITY MCC intel CCP5

Engineering and Physical Sciences Research Council
UNIVERSITY of York
Durham University
UNIVERSITY OF LINCOLN
Lancaster University
Science and Technology Facilities Council
Scientific Computing

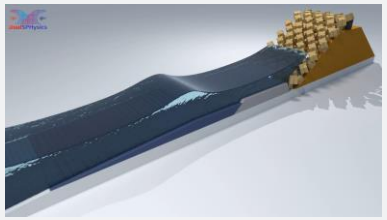
UCL
University of Hertfordshire UH
University of Southampton
KING'S College LONDON
MANCHESTER 1824 The University of Manchester
epcc

ExCALIBUR 10

PAX-HPC

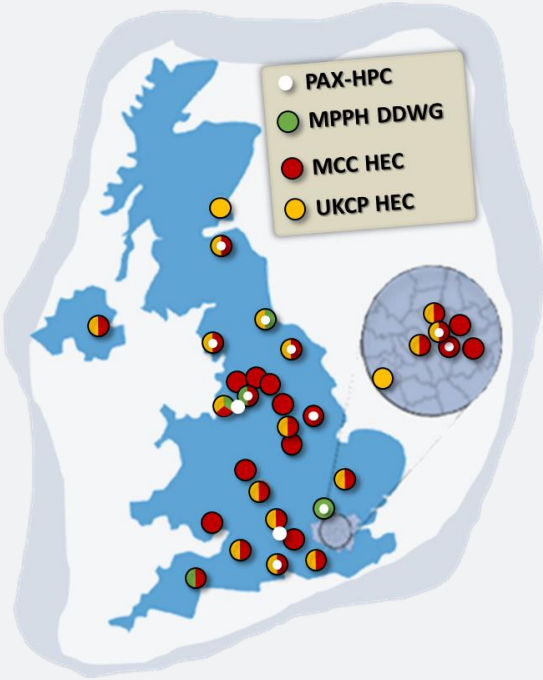
{MMM DDWG
MPPH DDWG

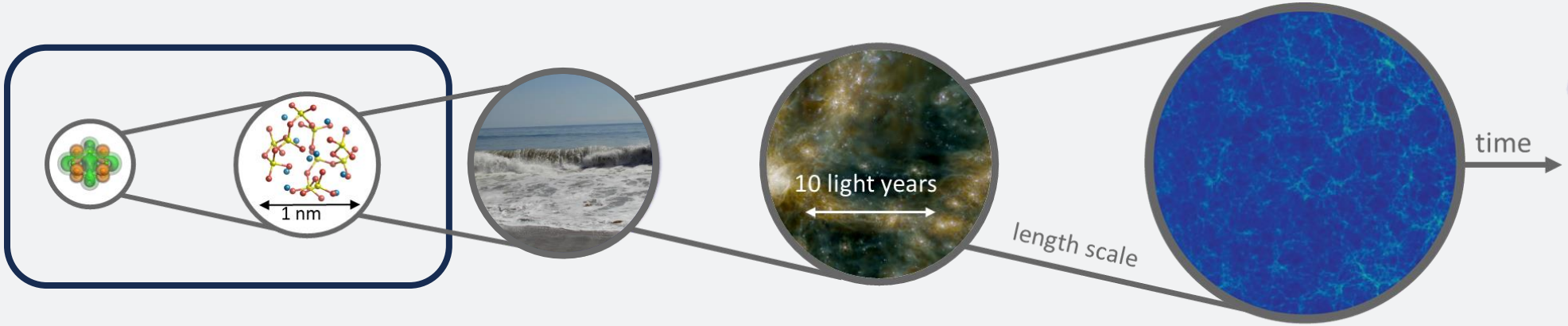
Particles at the eXascale on HPC



Keggeris et al. (2019)

Altomare et al. (2014)





CCP-QC · UKCP · CSCS Centro Svizzero di Calcolo Scientifico Swiss National Supercomputing Centre · DiRAC · nag · MATERIALS AND MOLECULAR MODELLING HUB · NVIDIA · NREL · Universiteit Leiden · Duke University · MCC · intel · CCP5

Engineering and Physical Sciences Research Council · University of York · Durham University · University of Lincoln · Lancaster University · Science and Technology Facilities Council Scientific Computing

UCL · University of Hertfordshire UH · University of Southampton · King's College London · Manchester 1824 The University of Manchester · epcc

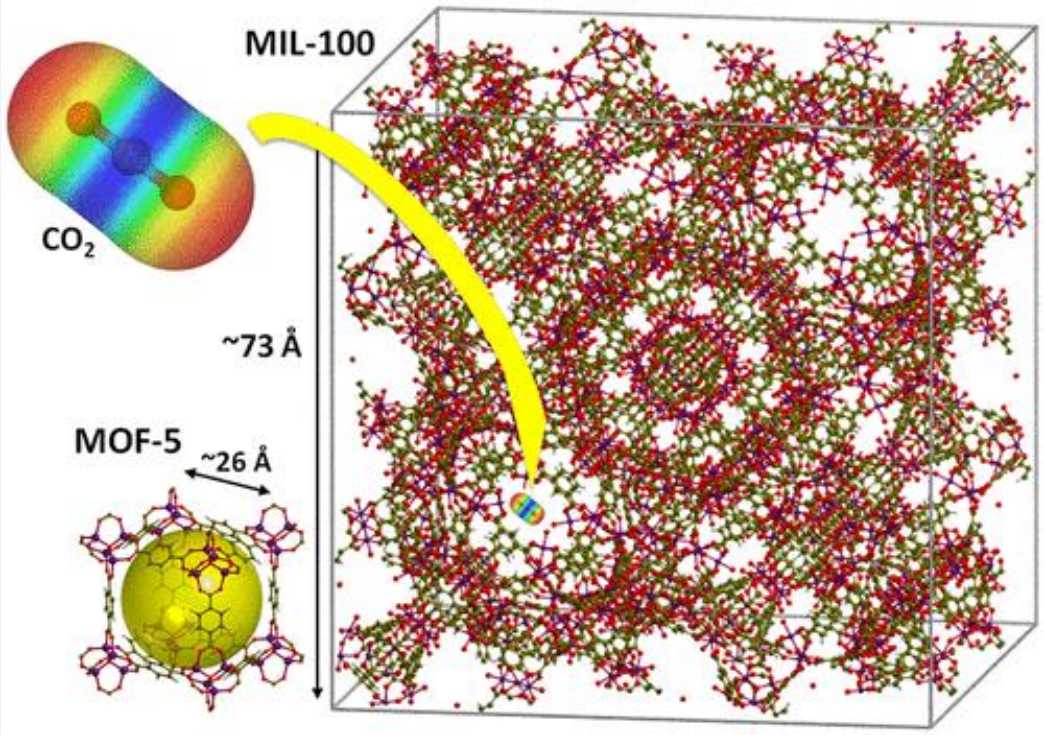
Ab initio Electronic Structure codes

Electronic structure codes compute the properties of the system under study from first principles.

They are used for:

- Modelling and rational design of materials
- Catalysis, batteries, solar cells, pharmaceuticals
- Novel materials.

Our work is focused on CP2K and CRYSTAL



Giant Metal Organic Framework MIL-100 investigated with CRYSTAL for use in carbon sequestration (https://doi.org/10.1021/acs.jpcc.9b06533)

Science and Technology Facilities Council

GPUs in *ab initio* electronic structure codes

The state of the system is the solution of the Roothaan equation

$$F\epsilon = SC\epsilon$$

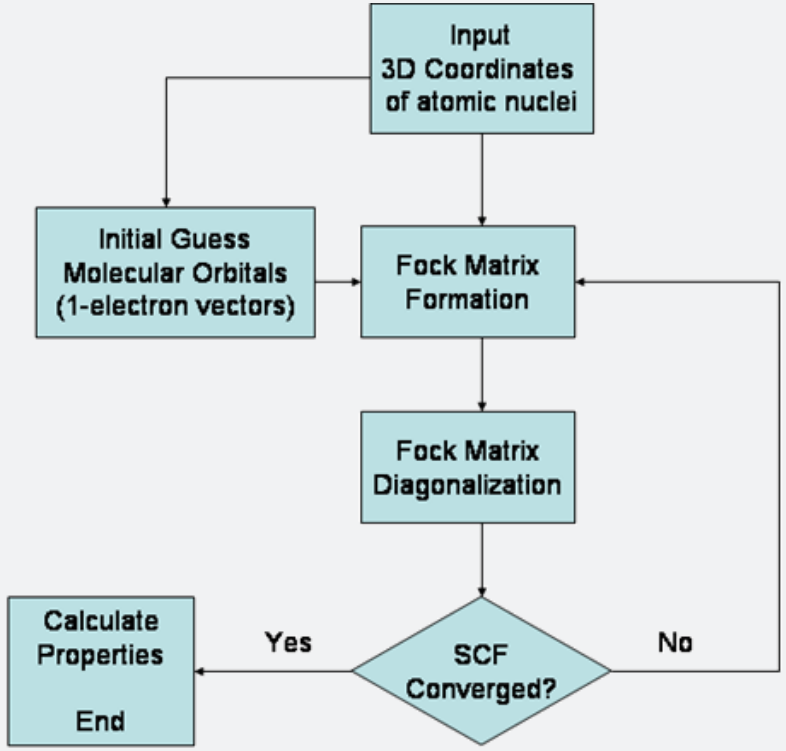
The majority of the computational work in these codes consists of two steps

•Using the density matrix C to form the Fock Matrix F
• Diagonalizing the Fock Matrix F to obtain the new density C

These steps are repeated in the Self-Consistent Fields (SCF) procedure

Step 1 requires the evaluation of an **enormous** number of integrals
Step 2 requires expensive linear algebra on large matrices

Make these calculations more affordable by exploiting the computational power provided by modern high-end



Science and Technology Facilities Council

Benchmarks of ELPA Eigenvalues Solvers on GPUs

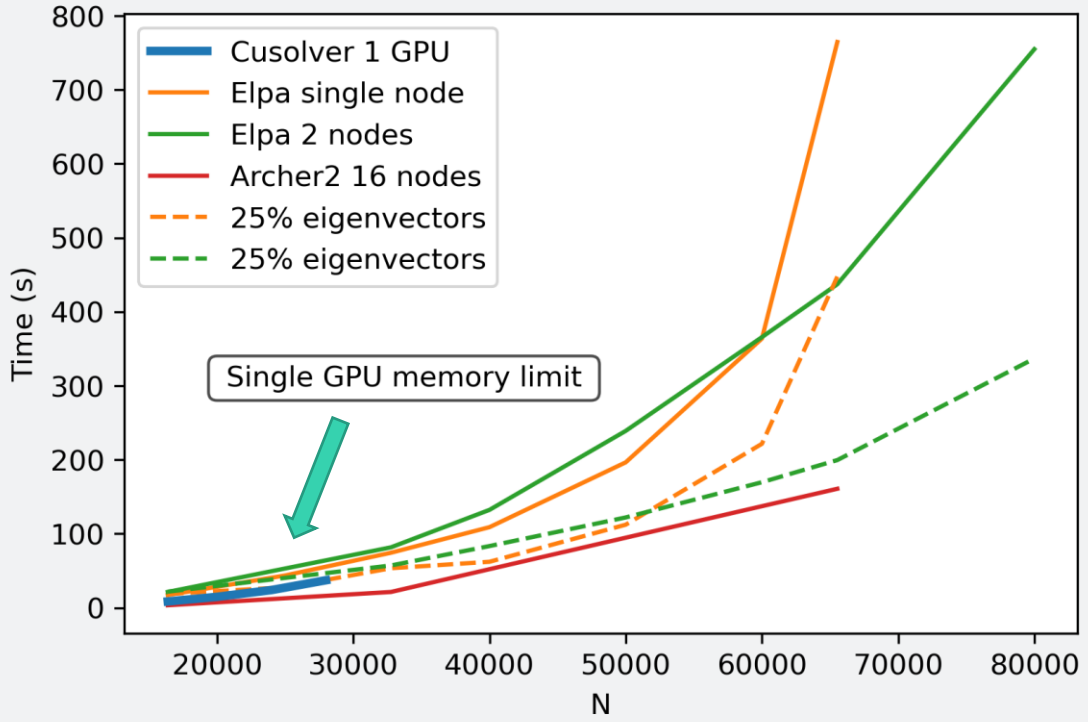
ELPA (Eigenvalue Solvers For ~~Peta~~Exaflop Applications) is a software library built to scale and to run on heterogenous architectures, including GPUs.

The interface is similar to ScaLAPACK, different routine for matrix diagonalization

ELPA can also compute a partial solution. This can result in a large saving of computational time

Tested on 4 A100 40GB per node at scarf

- NVIDIA cuSolver can be faster than ELPA running on multiple GPUs (*)
- Solving the partial problem significantly reduces the time to solution.



Science and Technology Facilities Council

GPU Acceleration of the Calculation of the Fock Matrix

The creation of the Fock Matrix is a complex process. Accelerating its formation using GPUs is not trivial.

What is the best approach to the efficient calculation on GPUs?
How to integrate into existing codes?
How to balance the load across multiple GPUs?

Start with the simpler processes and move our way up to the most complex components.

- Apply lesson learned along the way.
- First steps: use the Overlap matrix as a simple test-bed
- At the end, compute the 4 centres – 2 electrons integrals on GPUs

No free lunch -> (often) the routines that take more time are also the harder to accelerate using GPUs

Science and Technology Facilities Council

GPU Acceleration of the Calculation of the Fock Matrix

We focus on the contribution to the Fock matrix from the exact Hartree-Fock Exchange

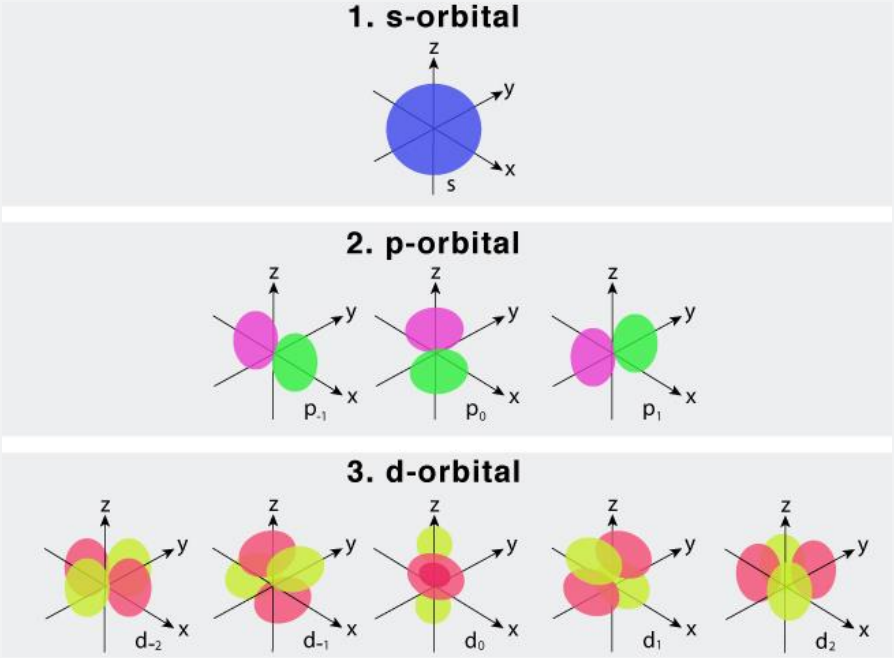
$$F_{ik} \mathrel{-}= D_{jl}\ (ijkl)$$

We wish to use to GPUs to accelerate the calculation of the 4 centres – 2 electrons integrals

$$(ijkl) = \int d^3x \int d^3x' \frac{\varphi_i(x)\varphi_j(x)\varphi_k(x')\varphi_l(x')}{|x - x'|}$$

(Also called more generally Electronic Repulsion Integrals ERI)

Where the φ are Gaussian orbitals of the type $\varphi = Y_L^m\ e^{-\alpha\, x^2}$



Science and Technology Facilities Council

GPU Acceleration of the Calculation of the Fock Matrix

Integrals of the type $(ssss)$ can be done analytically.
Integrals of higher moments are iteratively built from the $(ssss)$ and their ~derivative
We use the Obara-Saika recurrence relations and Head-Gordon-Pople method

E.g. $(psss) = \left(\bar{R}_{ij} - \bar{R}_i\right)(ssss) + \left(\bar{R}_{ijkl} - \bar{R}_{ij}\right)(ssss)'$

We repeatedly apply these and other operations to increase the moment of the integrals

E.g. The $(dddd)$ integrals require 86 steps

We precompute a plan of execution and refine it to:

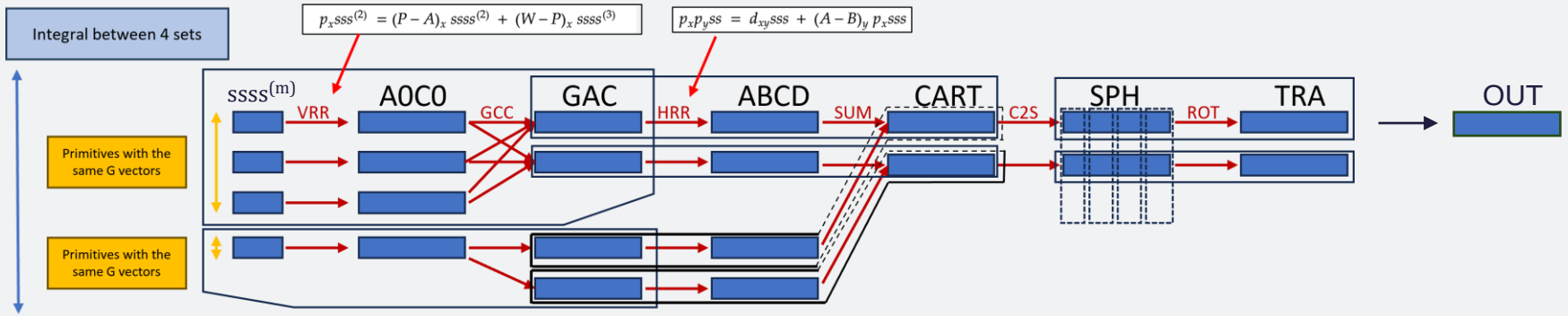
- Fuse similar operations to reduce the number of steps
- Identify transformations that can be done concurrently -> only synchronize when necessary
- Precompute memory offsets and array sizes

We prepare this plan on the CPU, save it and run it on the GPU.

Science and Technology Facilities Council

GPU Acceleration of the Calculation of the Fock Matrix

The control logic on the GPU code itself is relatively simple, the complex code is kept on the CPU side.



The calculation is done in phases. The parallelisation of work between SM is not trivial.

Some operations can be split naturally to 1 Integrals -> 1 SM
Other operations are better split between SM

- 1 Streaming Multiprocessor per GAC, 1-32 threads per Primitive
- 1 Streaming Multiprocessor per HRR transformation, 1 thread per element

Science and Technology Facilities Council

GPU Acceleration of the Calculation of the Fock Matrix

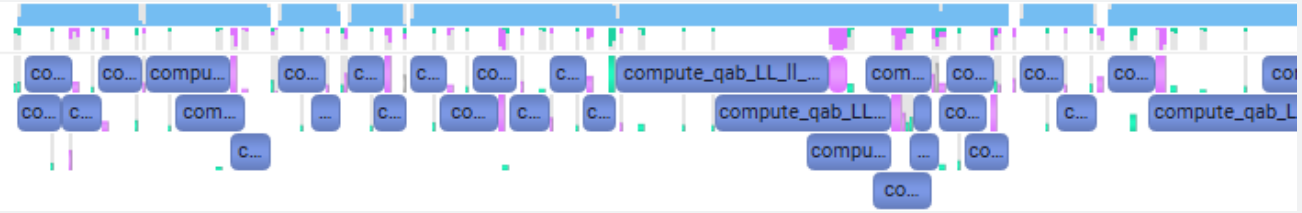
I only described the innermost core of the ERIs calculation

(1) I described a simplified view. Real ERIs are symmetric, periodic, screened linear combinations of integrals

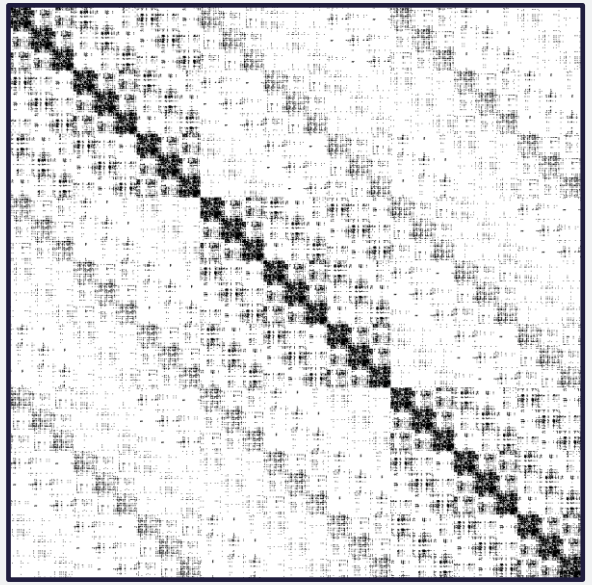
Each integral is relatively small - nanoseconds to maybe fraction of a millisecond –
How we pack the integrals to expose parallelism on the GPU may be just as important
Complexity hinders parallelism

(2) We don't exactly need the ERI themselves. We need $F_{ik} -= D_{jl}\ (ijkl)$

Doing this contraction on the GPU significantly reduces memory transfer



Profiler trace of the (much simpler) calculation of the Overlap matrix on 4 Cuda streams.



Sparsity pattern of F for an 864 H_2O system

Science and Technology Facilities Council

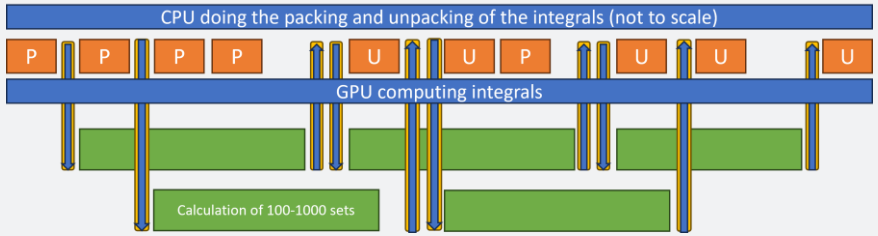
Conclusions

We discuss the use of Graphical Processing Units (GPUs) in local basis set *ab initio* electronic structure codes CP2K and CRYSTAL

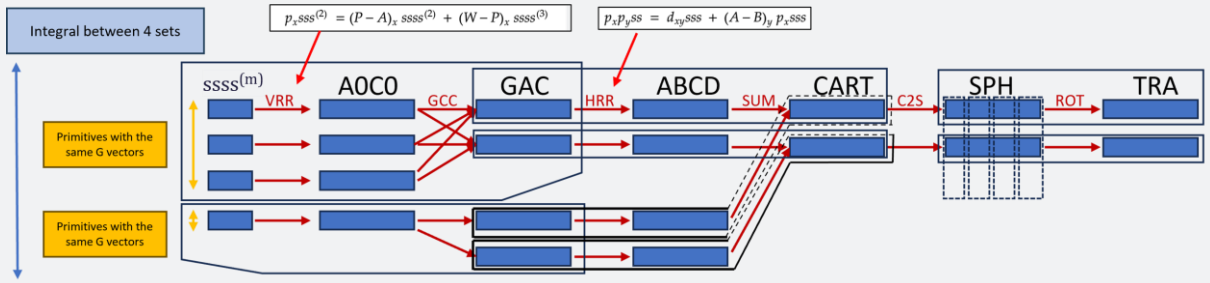
We briefly discussed some of the ways GPUs can be used to accelerate these codes

We presented some results about the diagonalization of the Kohn-Sham / Fock matrix

We discussed some of the opportunity and challenges in the calculation of the exact Hartree-Fock Exchange



Scheme for the buffered calculation of the HF exchange



Simplified representation of a ERI calculation on a generalized contracted basis set

UK Research and Innovation

ExCALIBUR

PAX-HPC