Developing a User-Centric CP2K Benchmark Suite

Fiona Reid - fiona@epcc.ed.ac.uk
Iain Bethune – ibethune@epcc.ed.ac.uk

Background

CP2K [1] is a freely available atomistic simulation program, able to study a wide range of molecular and bulk materials with methods from classical potentials through Density Function Theory (DFT) and QM/MM to Hartree-Fock and Møller-Plesset 2nd order perturbation theory (MP2). In particular, CP2K contains a highly efficient linear-scaling DFT implementation based on the Gaussian and Plane Waves (GPW) method [2] known as QUICKSTEP. Built on top of this are a wide range of tools such as molecular dynamics, Monte Carlo, geometry optimisation, Nudged Elastic Band, path integrals and metadynamics.

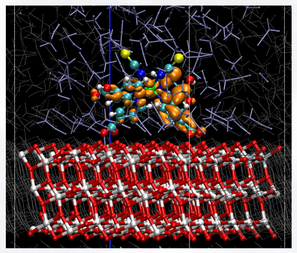

Figure 1: Dye-sensitised Solar Cell treated using CP2K. Image from J. VandeVondele.

Developed for over 10 years by a distributed team led by Prof. Jürg Hutter (Uni. Zürich) and Prof. Joost VandeVondele (ETH Zürich), CP2K contains over 800,000 lines of Fortran 95 code, including a sophisticated mixed-mode parallelisation strategy, scalable sparse matrix multiplication library and GPGPU support. It achieves excellent performance on HPC architectures from Linux clusters through to the Cray XE/XK and IBM BlueGene. CP2K is the second most heavily-used code on the UK national service HECToR, dominates usage on the Cray supercomputers at CSCS, the Swiss national HPC centre, and is also widely used across the various systems of the PRACE Research Infrastructure.

[1] CP2K Website, www.cp2k.org
[2] Quickstep: fast and accurate density functional calculations using a mixed Gaussian and plane waves approach, VandeVondele *et al*, Comp. Phys. Comm. 167, 103 (2005)

The CP2K Benchmark Suite

The aim of the CP2K benchmark suite is to provide a suite of benchmarks which can be used to guide scientists towards the best CP2K configuration (e.g. machine, number of MPI processors, OpenMP threads) to use for their particular problem. The benchmark suite consists of five benchmarks chosen to be representative of the range of calculations which can be run with CP2K.

The PSMP (mixed mode MPI/OpenMP) version of the code is used to measure performance. All possible combinations of MPI processes and OpenMP threads are tested. As many HPC systems charge by the node, full nodes are utilised at all times in our tests.

We present the results of our benchmarks on two different Cray systems, HECToR, a Cray XE6 and ARCHER, a Cray XC30. Respectively these machines are the previous and latest UK National Supercomputing Service. Table [1] gives a comparison of the architecture of the two systems.

Feature	HECToR	ARCHER
Processors	AMD Interlagos 2.3GHz	Intel Ivy Bridge 2.7GHz
Cores per node	32 (4× 8-core NUMA)	24 (2× 12-core NUMA)
Memory per node	32 GB (1 GB/core)	64GB (2.66 GB/core) 128GB (5.33 GB/core)
Nodes	2816 (90,112 cores)	3008 (72,192 cores)
Interconnect	Cray Gemini	Cray Aries
Topology	3D Torus	Dragonfly
Peak performance	1645.7 TFlops/s	829 TFlops/s

Table 1: Architecture comparison of HECToR and ARCHER.

Benchmark Results

We present the results of several of the benchmarks in the suite. The H2O-64 benchmark is a short molecular dynamics simulation running for 10 timesteps in an NVE ensemble at 300K. It consists of 64 water molecules (192 atoms, 512 electrons) in a 12.4 Å^3 cell. The system uses QUICKSTEP DFT with the LDA functional, a TZV2P basis set and 280 Ry cut-off. The H2O-64 system is a typical small simulation that would routinely be run on ARCHER.

Fig 2 shows the runtime of H2O-64 on ARCHER and HECToR plotted against the number of nodes. For each number of nodes we tested all feasible process/thread combinations and picked the best (fastest) one. Above each data point is the number of threads that gave the best performance where MPI means that the single thread time was fastest. Below each data point is the ratio of ARCHER/HECToR runtime Comparing the performance of the two systems we can see that ARCHER is around twice as faster as HECToR and also scales better.

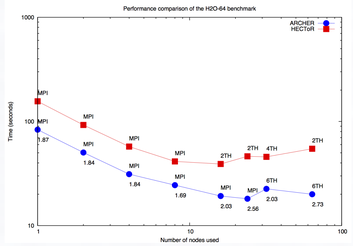

Figure 2: Performance of the H2O-64 benchmark.

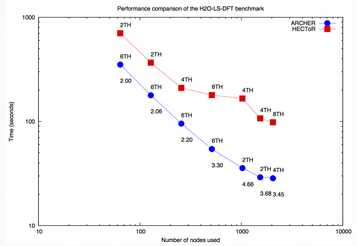

Figure 3: Performance of the H2O-DFT-LS benchmark.

The H2O-DFT-LS (see Fig 3) benchmark is a single-point energy calculation using linear-scaling DFT. It consists of 6144 atoms in a 39 Å^3 box (32 water molecules in a 4x4x4 supercell). An LDA functional is used with a DZVP MOLOPT basis set and a 300 Ry cutoff. For larger systems the linear-scaling approach will be much cheaper computationally than standard DFT and allows scaling to 100,000's of atoms. Again ARCHER is always more than two times faster than HECTOR and scales better at larger (>128 node) processor counts due to the improved performance of the Dragonfly interconnect over the Gemini Torus on HECToR.

The H2O-64-ri-mp2 benchmark is a single-point energy calculation using 2nd order Møller-Plesset Perturbation Theory (MP2) to calculate the exchange-correlation energy. The system consists of 64 water molecules in a 12.4 Å^3 cell. This type of model is currently at the cutting-edge of scientific research and has a computational cost around 100 times more than standard DFT. The performance of H2O-64-ri-mp2 (see Fig 4) again shows ARCHER being around twice as fast as HECToR.

We have presented some preliminary results from the CP2K benchmark suite. In future we plan to add results from more HPC systems and also to make the benchmarks and associated scripts available to users such that they can use the benchmarks to work out the best process and thread combinations for their own problems.

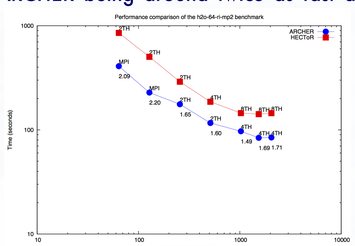

Figure 4: Performance of the H2O-64-ri-mp2 benchmark.

CP2K-UK: User Support and Training

CP2K-UK is an EPSRC-funded project to foster the growth of an active and vibrant community of CP2K users and developers in the UK. Led by Iain Bethune (EPCC), Prof. Lev Kantorovitch (KCL), Dr. Ben Slater and Dr. Matt Watkins (UCL), our vision is to enable increased uptake and more effective usage of CP2K within the UK atomistic simulation community.

Running from 2013 to 2018, the project will improve the usability of CP2K by the provision of improved documentation and case studies, making it much easier for new users to quickly achieve successful results with their calculations. We will also implement a number of new algorithms in CP2K with the aim of enabling the simulation of larger and more complex systems required to tackle key UK research challenges, for example: understanding the function of large biomolecules, porous systems such as zeolites and Metal-Organic Frameworks (MOFs), design of new materials for fuel cells, nanotechnology including scanning probe microscopies, and next-generation catalysts to create fuels from solar radiation. Finally, we will provide support, training and networking activities for CP2K users, share best practice through a series of annual user group meetings and training courses, and as a result enable the community to not only make effective use CP2K but also to develop and extend the code to support their individual research goals.

The CP2K-UK network already has members from eight research groups around the UK, and we aim to grow! If you use (or would like to use) CP2K in your research please get in touch with us – ibethune@epcc.ed.ac.uk.

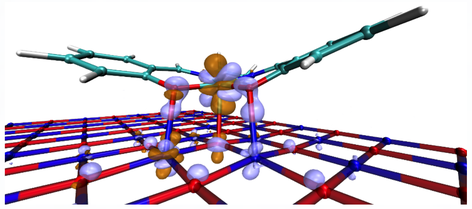

Figure 5: Charge density difference induced by Co-Salen molecule interacting with NiO(100) surface. From A. Schwarz *et al*, J. Phys. Chem. C, 17 1105 (2013). Image courtesy M. Watkins.

Acknowledgements

CP2K-UK is supported by the Engineering and Physical Sciences Research Council [grant number EP/K038583/1].

This work made use of the facilities of HECToR, the UK's national high-performance computing service, which are provided by UoE HPCx Ltd at the University of Edinburgh, Cray Inc and NAG Ltd, and funded by the Office of Science and Technology through EPSRC's High End Computing Programme. This work used the ARCHER UK National Supercomputer Service (http://www.archer.ac.uk).

The support of Prof. Jürg Hutter, Prof. Joost VandeVondele and staff at CSCS for providing access to HPC systems used for testing the code is gratefully acknowledged. Alfio Lazzaro is also acknowledged for providing us with an optimised version of libsmm which was used on the ARCHER machine.