|epcc|

NSCCS/ARCHER CP2K UK Workshop 2014: CP2K Hands-on Exercises

1 Introduction

The hands-on sessions are designed for you to work at your own pace through a series of exercises. Instructors are available to answer questions relating to the exercises, or other questions on CP2K. Please read through this instruction sheet and decide which exercises you would like to work on, or if you have calculations on your own system you would like to attempt.

2 CP2K Exercises

A series of exercises are available at the CP2K website. These are designed for beginners, and walk through many different features of CP2K. Please note that the exercises refer to a parallel machine called 'Brutus'. You will need to adapt the instructions to execute CP2K on either the local lab PCs or ARCHER.

`http://www.cp2k.org/exercise`

The exercises grouped under 'Lecture 1' are a good introduction to doing classical calculations using CP2K. 'Lecture 2 - 3' cover some further classical calculations, and also analysing results with some scripts. 'Lecture 4 & 5' covers Molecular Dynamics in NVE and NVT ensembles, and 'Lecture 6' and beyond use DFT calculations with Quickstep.

Some more detailed tutorials can also be found at:

`http://www.cp2k.org/tutorials`

These take a closer look at the fundamental tasks needed to set up a reliable, well-converged Quickstep calculation e.g. "How to Calculate Energy and Forces" and "How to Converge the CUTOFF and REL_CUTOFF".

Look through both pages and choose some exercises that match your level of experience and interests.

3 CP2K Case Study Scenarios

A set of more extended case-study type exercises are available, aimed at users with some existing experience of using CP2K, who wish to explore particular functions of CP2K in more depth. These consist of a set of CP2K input files, supporting data, and some suggestions for activities to explore. The relevant files can be downloaded from the course web pages:

`https://www.archer.ac.uk/training/course-material/2014/08/CP2K/Exercises`

- Metadynamics: `marcella_mtd1.tar.gz`
- NMR: `marcella_nmr.tar.gz` (no instructions, just input files)

- X-ray Spectroscopy: `marcella_xray.tar.gz`
- Normal Mode (vibrational) Analysis: `Flo_Normal_modes.tar.gz`
- QM/MM: `kuo_FIST_and_QMMM.tar.gz`
- Thermostatting MD: `bussi_thermostats.tgz`

4 CP2K Parallel Scaling Exercises

Note: these exercises require use of ARCHER, so can only be done on Thursday 28th

CP2K is distributed with some input files which serve as benchmarks for Quickstep, consisting of MD calculations on varying sizes of liquid water systems ranging from 32 molecules up to 2048. The benchmarks are located in the `cp2k/tests/QS/benchmark` directory. You can either download the CP2K distribution from `http://www.cp2k.org/download` or browse the input files online at:

`http://sourceforge.net/p/cp2k/code/HEAD/tree/trunk/cp2k/tests/QS`

Performance data for some of these benchmarks is published at

`http://www2.epcc.ed.ac.uk/~ibethune/files/cp2k_cug2014.pdf`

which should guide your investigations. There are also other benchmarks for LS-DFT, HFX and MP2, but these require a large amount of nodes to complete in a reasonable time. If you wish to experiment with these, you might be able to reduce some accuracy parameters to reduce the runtime.

The following investigations are suggested:

- For a fixed problem size e.g. `H2O-32.inp` vary the number of MPI processes and plot the run time of the job against the number of processes - be careful to stop when you have reached the point at which scaling stops and the calculation gets slower.
- From the output files from your scaling runs, select the routines with the largest '`SELF TIME`' in the timing report, and plot these against the number of MPI processes. Do some parts of the code scale better than others? If so, why? Is load imbalance a noticable effect?
- Having identified a fixed total number of nodes, use the mixed-mode MPI/OpenMP CP2K executable `cp2k.psmp` and experiment with varying numbers of OpenMP threads per MPI process. To do this, replace the `aprun` line in your batch script with:

  ```
  export OMP_NUM_THREADS=XX
  aprun -n YY -N ZZ -d XX $CP2K/cp2k.psmp H2O-32.inp
  ```

  `XX` is the number of OpenMP threads for each MPI process, and so should be between 1 and 24. `YY` is the number of MPI processes. You must ensure that `XX` $\times$ `YY` is less than or equal to the total number of cores you have available (24 $\times$ number of nodes). `ZZ` is the number of MPI processes to place on each node, and should be set to 24 / `XX`, rounding down. Does OpenMP improve the performance of CP2K on this number of nodes? What happens when you use more nodes? Can you identify from the timing report which routines have been parallelised with OpenMP?
- Keeping the number of nodes fixed (and number of MPI processes and/or OpenMP threads), vary the problem size from `H2O-32` to `H2O-2048`. How does the overall runtime scale with the problem size? From the timing report, compare what fraction of time is spent in different parts of the code for each problem size? Which parts of the code dominate for small problem sizes? Which for large sizes?
- For a particular input file on a given number of processes, investigate how changing the accuracy parameters of the calculation affect the performance. This is best done by switching from Molecular Dynamics (`RUN_TYPE MD`) to single-point energy calculation (`RUN_TYPE ENERGY`), and

comparing the "Total FORCE_EVAL (QS) energy". Variables that might be of interest are the realspace grid parameters CUTOFF, REL_CUTOFF and NGRIDS. The accuracy of Quickstep can be directly tuned using the EPS_DEFAULT parameter, and you can also experiment with the SCF procedure e.g. EPS_SCF and method for solving the KS equations (OT or DIAGONALIZATION).

5 Thousands more examples...

The CP2K distribution contains over 2500 input files in the cp2k/tests/ directory, which serve as examples of the wide variety of functionality in CP2K, as well as being used as a regression test suite to ensure all the functionality is working! You can either download the CP2K distribution from http://www.cp2k.org/download or browse the input files online at:

http://sourceforge.net/p/cp2k/code/HEAD/tree/trunk/cp2k/tests/.

The tests are designed to run quickly (most will run in under 5 seconds on a single core), so convergence and accuracy parameters will likely be looser that would be required for a real calculation. However, the tests do provide good examples of which keywords are relevant for different types of calculation, and are grouped logically e.g.:

- tests\QS : directory for Quickstep
- tests\QS\regtest-gpw-* : GPW input files
- tests\QS\regtest-gapw-* : GAPW input files
- tests\QS\regtest-mp2-* : MP2 input files
- tests\SE : directory for Semi-empirical calculations
- tests\Fist : directory for classical calculations
- etc.

Feel free to use any of these files as a starting point for experimentation!