Optimisation Practical

Adrian Jackson

Introduction

The purpose of this practical is to gain experience of serial optimisations of programs.

Obtaining the Code

Once you have logged into the host machine you need to obtain a copy of the source code. You can obtain this using *wget* as follows:

```
wget https://www.archer.ac.uk/training/course-material/2016/06/
xeonphi_soton/exercises/exercise5/md.tar.gz
```

There is one directory in this archive:

- MD

This directory contains both C and F versions of the code, use whichever is most suitable for you. They have working Makefiles targeted at the platform we are using, if you want to develop on different platforms/machines you may have to change the Makefiles. The current makefile does not include flags for Xeon Phi so you will need to modify the makefile to ensure any code that is built will work on the hardware (although these practicals can be done on both the host processors and the Xeon Phi).

Optimisations

The aim of this practical is to perform single processor performance optimisation for a simple molecular dynamics code. We suggest starting with investigating compiler optimisations, profiling, and then examining the code to see if there are any other optimisations that can be undertaken.

Note that optimising the code may change the floating point results slightly, so a simple `diff` on output files is not a useful verification test. The subdirectory `Test` contains a C program which, when compiled, can be used to test that two output files from the MD code are the same to within an acceptable tolerance. The syntax for this is

```
diff-output file1 file2
```