



Performance Program Coursework

Introduction

The aim of this coursework is to perform single thread performance optimisation on the back end compute nodes of **Cirrus** for a simple application code and to produce a written report on the results of this activity. Note that the target platform is Cirrus and its associated software. If you do not already have access to Cirrus please contact the course organiser.

We will be using a simple molecular dynamics code available on Learn and called MD_2019.tgz.

There are both C and Fortran versions of the code available. You should select **one** of these versions for use in the coursework, and work only on that version.

Running the program

As provided the program reads an initial state from the file input.dat and then performs 5 blocks of 100 timesteps writing an output file after each block. The output files are in the same format as the input file so you can use any output file as an input for a shorter running performance test that performs less than 500 iterations. The code reports timing information for each block of 100 timesteps and for the loop over blocks that includes file access operations.

Checking correctness

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
```

This program will not detect the presence of NaN values in the input so you should test for these explicitly.

In addition, very small numerical differences will be magnified over time, particularly once the particles start to collide, so the verification test is unlikely to pass for more than 200 time-steps from a common starting point. The verification test is intended as a guide rather than a definitive test of correctness so you need to give some thought to how you test for correctness. We suggest building tests using blocks of 100 iterations (timesteps) from a region of the simulation after the particles have started to collide.

Assignment

The assignment is to produce a report (10-20 pages including figures) on the optimisation activity. The report may contain additional appendices if you wish, though assessment will be based on the main report. The report should present the results of your work investigating and improving the performance of this code. The report should make clear recommendations as to a final improved version of the code. These recommendations should consider factors such as code maintainability and readability as well as overall performance.

Your aim is to reduce the combined run-time of all 500 timesteps while maintaining a reasonable level of code quality. File I/O times do not need to be considered and can be omitted from timing results.

The coursework is intended to assess your understanding of the course material so approaches such as multi-threaded or multi-process parallelism should not be attempted.

You are required to submit this recommended code version along with the report but the assignment marks are based on the report so the report should be a **stand-alone document** with discussions of the code being illustrated by in-line code fragments rather than by reference to the submitted source code.

Please ensure that you include your exam number in the title of both your report and your source code. This assignment will be marked anonymously so we cannot identify which report goes with which source code unless you include your exam number in the title.

Marking scheme

The report will be marked on:

- Demonstrated understanding of the performance issues: both problems in the original code and of the results of changes made to the code **(40)**.
- Discussion of the proposed optimisations: their impact on performance as well as code quality **(30)**.
- Methodology used in the assignment as demonstrated in the report. This includes general approach, tools used etc. **(10)**.
- Clarity, relevance and presentation of the report **(20)**.

Coursework is due at **16.00 on Friday 5th April**.

As per the University's Taught Assessment Regulations (for further information see link on Learn course Assessment page) assignments submitted after the deadline (unless granted an extension, see Student Support page on the Learn course) are subject to a 5% penalty per day (i.e. 24 hours) that the assignment is late after the deadline, up to a maximum of seven. Assignments handed in more than seven days late receive zero marks.