Optimisations and Parallelism off d2q9-bgk.c

George Herbert
`cj19328@bristol.ac.uk`

February 19, 2022

Abstract

`d2q9-bgk.c` implements the Lattice Boltzmann methods (LBM) to simulate a fluid density on a lattice. The original `d2q9-bgk.c` code was unoptimised and not parallelised. This report outlines the techniques I utilised to optimise and parallelise `d2q9-bgk.c`, as well as a detailed analysis of those techniques. To do so, this report is split into several sections corresponding to different iterations of my code.

1 Original code

Table 1: Performance of original code

Test case	128×128	128×256	256×256	1024×1024
Time (s)	0	0	0	0

I compiled the original, unoptimised `d2q9-bgk.c` using the GNU Compiler Collection (GCC) with the following command:

```
gcc -std=c99 -Wall d2q9-bgk.c -lm -o d2q9-bgk.
```

Figure 1 contains the results of benchmarking the ELF file produced for each of the test cases. I benchmarked the original unoptimised implementation so that I could measure the performance increase of further implementations. I produced the benchmark by taking an average of the total time of 10 runs on BlueCrystal Phase 4's (BC4's) compute nodes. Each of BC4's compute nodes is a Lenovo nx360 M5, which contains two 14-core 2.4 GHz Intel E5-2680 v4 (Broadwell) CPUs and 128 GiB of RAM [1]. It is important to take an average of multiple runs because of the variation between runs, which exists due to the inconsistent performance of compute nodes; not all compute nodes offer the same performance all of the time, due to differing placement in the data centre, amongst other reasons.

2 Serial optimisations

2.1 Compiler

2.2 Code changes

2.3 Results

3 Vectorization

3.1 Code changes

3.2 Results

4 Parallelism

4.1 OpenMP

4.2 Results

References

[1] *BlueCrystal technical specifications*. URL: https://www.bristol.ac.uk/acrc/high-performance-computing/hpc-systems-tech-specs/ (visited on Feb. 19, 2022).