

IN THIS ASSIGNMENT, EXPLORE OPTIMISATIONS FOR THE SERIAL JACOBI
SOLVER CODE RUNNING ON A SINGLE CORE OF A BLUECRYSTAL PHASE 3 NODE

ASSIGNMENT 1:

SERIAL OPTIMISATIONS

Progress: 25%

ASSIGNMENT DESCRIPTION

- **Jacobi starting code:** <https://github.com/UoB-HPC/intro-hpc-jacobi>
- Using this code as a starting point, apply some of the serial optimisations from the lectures to improve performance
- Your code should only use a single core (no OpenMP yet)
- In particular, consider the following optimisations:
 - Compiler choice and flags
 - Data layout and data type
 - Loop fusion
 - Anything else you can think of!
- You should then produce a short report discussing your findings
 - Describe the optimisations that you tried
 - Explain why your optimisations improved performance
 - Provide results to back up your statements

Ballpark runtimes after applying
these optimisations:

1000x1000: <1 second

2000x2000: <8 seconds

4000x4000: <60 seconds

GUIDANCE PART 1

- To achieve a good mark of 60%+:
 - A well-written, 2 page report that clearly demonstrates you understand what you did
 - Code that successfully uses most of the optimisations we describe
- You should be able to get all these optimisations working in about 1 day (~7-8 hours)
- Your time on 1 core of Blue Crystal phase 3 should be consistent with the table on page 2 of this assignment

GUIDANCE PART 2

- To aim for a first (70%+), you'll need:
 - An excellent 2 page report
 - Code that:
 - Applies further optimisations that improve performance above those we've described. These may include code transformations beyond those discussed in class. It should certainly include effective vectorisation.
 - Achieves single core performance notably faster than those given in the table on page 2 of this assignment
- With ~3 weeks allocated to the serial optimisation assignment, 10 hours allocated to the course each week for 10 weeks, and 4 hours per week spent in lectures and labs, don't spend more than $3 * (10 - 4) = \sim 18$ hours on this assignment in total
 - It should only take 7-8 hours to do the simple version which should be good enough to earn 60%+

COURSEWORK SUBMISSION

- Your submission will be made via SAFE and should include:
 1. A **1 to 2 page** report in PDF form, which must include:
 - a. Your name and user id
 - b. A description of your serial optimisations;
 - c. Comparisons of your optimised performance vs unoptimised;
 - d. Analysis of the effectiveness of different optimisations you tried;
 2. The working code you used to generate the results in your report.
- Your code must converge to the same solutions as the starting code, after the same number of iterations (within a reasonable tolerance).

SUBMISSION REQUIREMENTS

- Your **report** which must be in a file called "**report.pdf**",
 - Lower case r: "**report.pdf**" NOT "**Report.pdf**"
- Your **source code**, i.e. "**jacobi.c**"
- Your **makefile**, called "**Makefile**"
- Don't modify the timing code in the starting code, as we'll use this to automatically extract timing information from each submission
- We must be able to reproduce any runtimes you quote in your report by compiling and running the code that you submit
- Don't zip these files up, instead submit them as separate files in SAFE

PLAGIARISM CHECKING

- The HPC assignments are all for individuals, they are **not** group work
- We will check all submitted code for plagiarism using the MOSS online tool
 - MOSS ignores the example code we give you
 - MOSS will spot if any of you have worked together or shared code, so **please don't!**
- We'll also check all submitted reports using the TurnItIn tool, which will find any shared text
- So please don't copy code or text from each other! You **will** get caught, and then **both** the copier and original provider will get a **0** for the whole assignment.

SUMMARY

- Remember, you'll get marks for:
 - A well written, comprehensive, report
 - A serial code that successfully explores most of the optimisations we suggest
- Have fun exploring how to optimise programs!