Parallel Computing 2022/23

Assignment2: Heatdiffusion using MPI

This assignment is based on the previous assignment. This time, your task is to

- develop an MPI version and explore the effectiveness of your version
- perform an extensive MPI performance study that includes different configurations, ideally including variations in the number of ranks per node as well as in the number of nodes involved.
- a performance comparison between your openMP solution from assignment 1 and your MPI solution from this assignment
- investigate whether you can run openMP within MPI, and if so, how that performs in comparison to the previous experiences
- discuss briefly which approach for parallelising stencils you would recommend to someone who has access to the `csedu` cluster.

This assignment should be done by teams of 2-3 students. How you distribute the work within the team is up to you. However, you need to declare who did which part. Ideally, you perform all performance measurements on a single system. If that constitutes a logistic problem, you can use different hardware provided you clearly specify which hardware has been used for which measurements.

As for the previous assignment, make sure that you:

- specify exactly what hardware is being used (CPU version, clock frequency, memory, etc.)
- specify exactly what software is being used (compiler version, compiler flags, etc.)
- specify exactly which parameter (size n and number of iterations) you are using
- repeat each experiment at least 5 times and report average time as well as the variability (error bars)

Task 1: MPI implementation

Rewrite the program to leverage MPI. Try to optimise your MPI version considering data distribution, communication effort, choice of communication method, etc. Repeat the evaluations from Assignment 1 for your MPI version(s) quantifying the effects that your own optimisation attempts have (scheduling, placement, communication pattern, etc.). Make sure that you request sufficient memory for the number of MPI ranks that you attempt to allocate. It is highly recommended to parallelise the optimised version! If you do this, it is not necessary to also parallelise the naive one.

Task 2: Performance comparison

Provide a discussion of your overall findings. This should include figures reporting speedups, scaling (strong and weak), efficiency. Use the roofline plot to explain the maximum performance you can expect from the naive and optimised versions. You should try to explain your findings and try to come up with possible further directions of investigation. Try to be brief; this should not take more than one page.

Hand-out: 10/05/2023 Submission deadline: 24/05/2021 8:30

Task 3: Newbie advice

Try to summarize your parallelisation experience by formulating some advice for a user that is given access to the `csedu` cluster and who wants to use it in order to make the sequential version faster. Which technology should the user in your opinion go for and why? Explain the difficulties that the user may have to deal with, based on your own experiences, and also do give the user an idea what kind of performance can be expected. Bear in mind that the user very likely has a specific problem size in terms of n and $iter$ and that the user also would like to know which fraction of the cluster would be a good choice for the given problem.

Task 4: Team

Provide a short description on how you divided up the work, i.e., who did what? Attribute percentages to your overall contributions.