PAR Laboratory Assignment
Lab 5: Geometric (data) decomposition using implicit tasks: heat diffusion equation

Your names here, group, date, and include boada usernames

Add delivery date here

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

How to:

Document contents

The length of the document is expected to be maximum 8 pages (including figures and tables) with a font size of at least 11pt.

You have to add comments and observations following the questions that you have found along the document indicating **For the deliverable:** in each section (in this laboratory we have included those **For the deliverable** questions in the notebook with setences in gray under each **Comments/Observations**). DON'T include code in the document except for the fragment modified with respect to the original one.

Finally, as you know, this course contributes to the **transversal competence "Tercera llengua"**. Deliver your material in English if you want this competence to be evaluated. Please refer to the "Rubrics for the third language competence evaluation" document to know the *R*ubric that will be used.

Submission

Only PDF format for this document will be accepted. Deliverable assignment will be opened in *Atenea* and set the appropriate dates for the delivery. You also have to deliver the complete C source codes for Tareador instrumentation and all the OpenMP parallelisation strategies that you have done. Your lab professor should be able to re-execute the parallel codes based on the files you deliver. You will have to **deliver TWO files**, one with the document in PDF format and one compressed file (`tgz`, `.gz` or `.zip`) with the requested C source codes.

1

Laboratory 5 notebook

1.1 Sequential heat diffusion program and analysis with Tareador

This part refers to section 2 of the practice document.

You must include:

- The task dependency graph shown by *Tareador* for both solvers *Jacobi* and *Gauss-Seidel* for the default codes (naive version).
- The task dependency graph shown by *Tareador* for both solvers for the proposed task granularity, adding the dependences filter and new tasks per block in other parts of the code to increase parallelism.
- The excerpt of the last version of the *Tareador* code that you have modified in order to specify **one task per block** and exploit other parts of the code in addition to `solver` function.

Comments/Observations

For the naive version: Is there any parallelism that can be exploited at the naive version granularity? For the proposed task granularity: Which variable was causing the serialisation of all the tasks? How will you protect the access to this variable in your OpenMP implementation? Are you obtaining more parallelism in the proposed task granularity than in the default version? Is the parallelism achieved the same for *Jacobi* and *Gauss-Seidel* solvers?

1.2 OpenMP parallelization and execution analysis: *Jacobi*

This part refers to section 3.1 of the practice document. In addition, some indication details of the specific subsections each question refers to are specified below.

You must include:

- An excerpt of the code to show the OpenMP annotations you have added to the code after the optimizations (Optimization section - Jacobi solver).
- The *modelfactors* tables and the plot of scalability for the first implementation (Overall Analysis section - Jacobi solver) and your last optimized implementation (Overall Analysis of the Optimized Code - Jacobi solver).
- Captures of the window timelines for both first (Detailed Analysis section - Jacobi solver) and last implementation (Detailed Analysis of the Optimized Code).

Comments/Observations

What was the region of the code that was provoking the low value for the *parallel fraction* in your first parallelisation? (Detailed Analysis section) Compare the *parallel fraction* of your first and last versions (Overall Analysis of the Optimized Code section). Is the execution time reduced from your first to the last version?. Have you increased the scalability? (Overall Analysis of the Optimized Code section) Compare the timelines of both executions under the point of view of instantaneous parallelism (Detailed Analysis of the Optimized Code section).

1.3 OpenMP parallelization and execution analysis: *Gauss-Seidel*

This part refers to section 3.2 of the practice document. In addition, some indication details of the specific subsections each question refers to are specified below.

You must include:

- The plot of scalability when using 20 blocks in the `j` dimension, `nblocksj=20` (Overall Analysis section).
- An excerpt of the OpenMP code to show the modifications done: OpenMP annotations and synchronization mechanisms (once you make `nblocksj=userparam * nblocksi`) (Detailed Analysis and Optimization section).
- The plots obtained with `submit-userparam-omp.sh` using 16 threads (Finding the appropriate value for the number of blocks section) and and strong scalability for the best value of `userparam` (`nblocksj=userparam * nblocksi`).

Comments/Observations

Do you observe a linear speedup for 20 blocks in the `j` dimension? (Overall Analysis section). Reason *why* changing the number of blocks in the `j` dimension changes the ratio between computation and synchronisation (Detailed Analysis and Optimization, Number of blocks tune section) and explain the plots that you have included, comparing the strong scalability for both cases `nblocksj=20` and `nblocksj=best_userparam * nblocksi` (Finding the appropriate value for the number of blocks).

Final Laboratory Survey

We would like to get some feedback from you so that we can continue improving the practical part of the course. In particular, we are interested in your opinion about the usage of *modelfactors*. Can you please tell us briefly your opinion about it? Was *modelfactors* useful for you in order to understand the performance of your parallel application? Would you advice its use in the laboratory assignments of this course in future editions?

From 0 to 10, how would you rate:

- *modelfactors*;
- *Tareador*;
- *Extrae + Paraver*.

On the other hand, we have made an effort to reduce the volume of things to deliver in the lab documents, trying to guide more the information we require. From 0 to 10, how would you rate (0 - too much, 10 - very well) the volume of documentation to be delivered.

Feel free to include any other comment that you want to add about the practical sessions.