Matrix

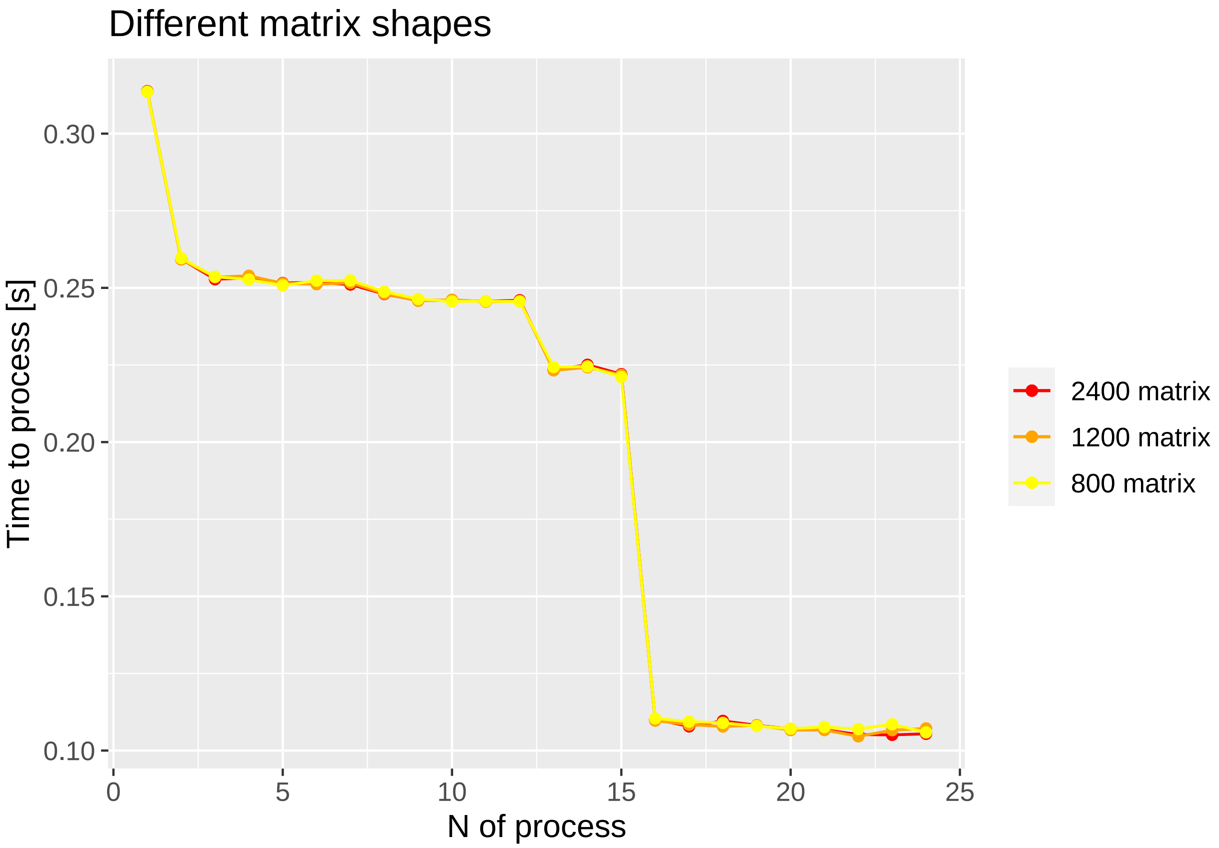
Verardo Thomas

29/12/2021

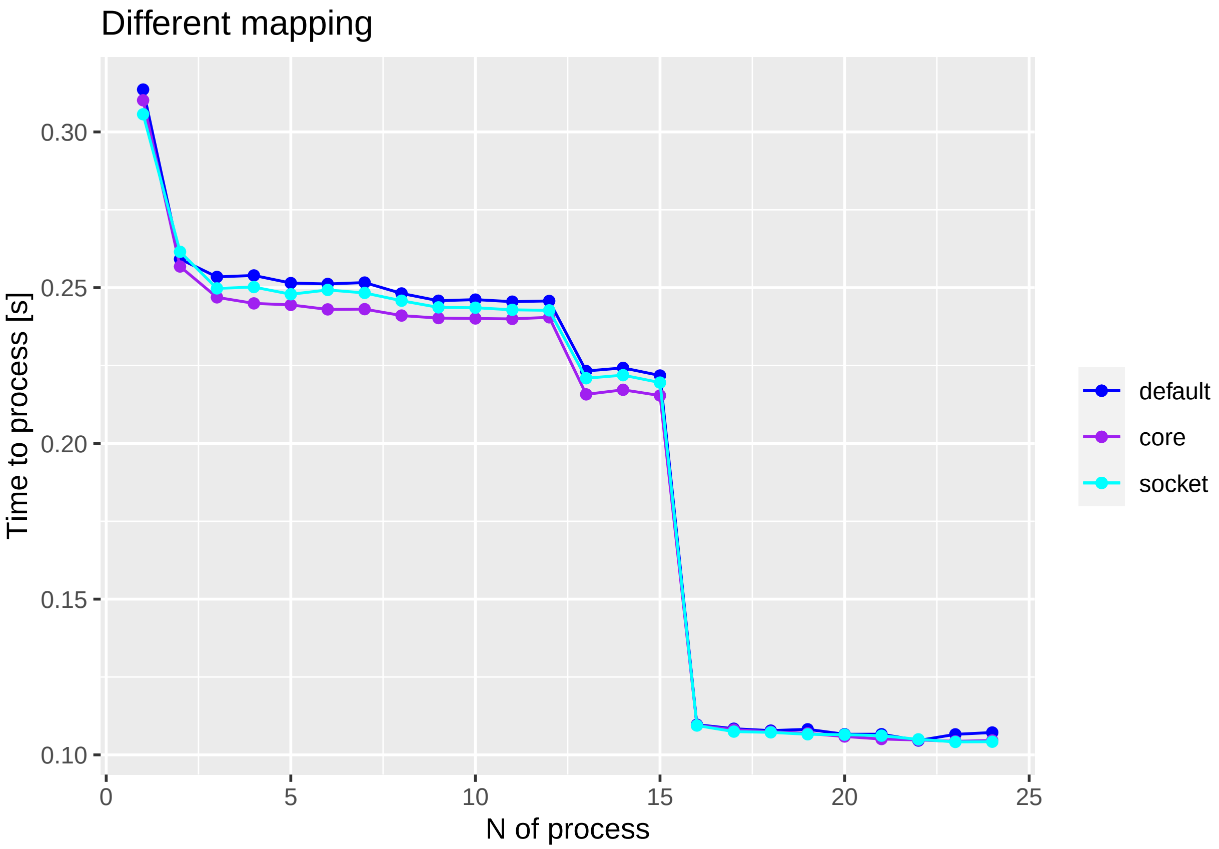
SECTION1 (Matrix)

The matrix is initialized with random number and, after building the topology, that is always (s, 1, 1) as s the numbers of processes, the matrix is first divided and then sent to all the processes. So, each process computed a part of the matrix, that is exactly the shape of the matrix ($x * y * z$) divided by the number of processes.

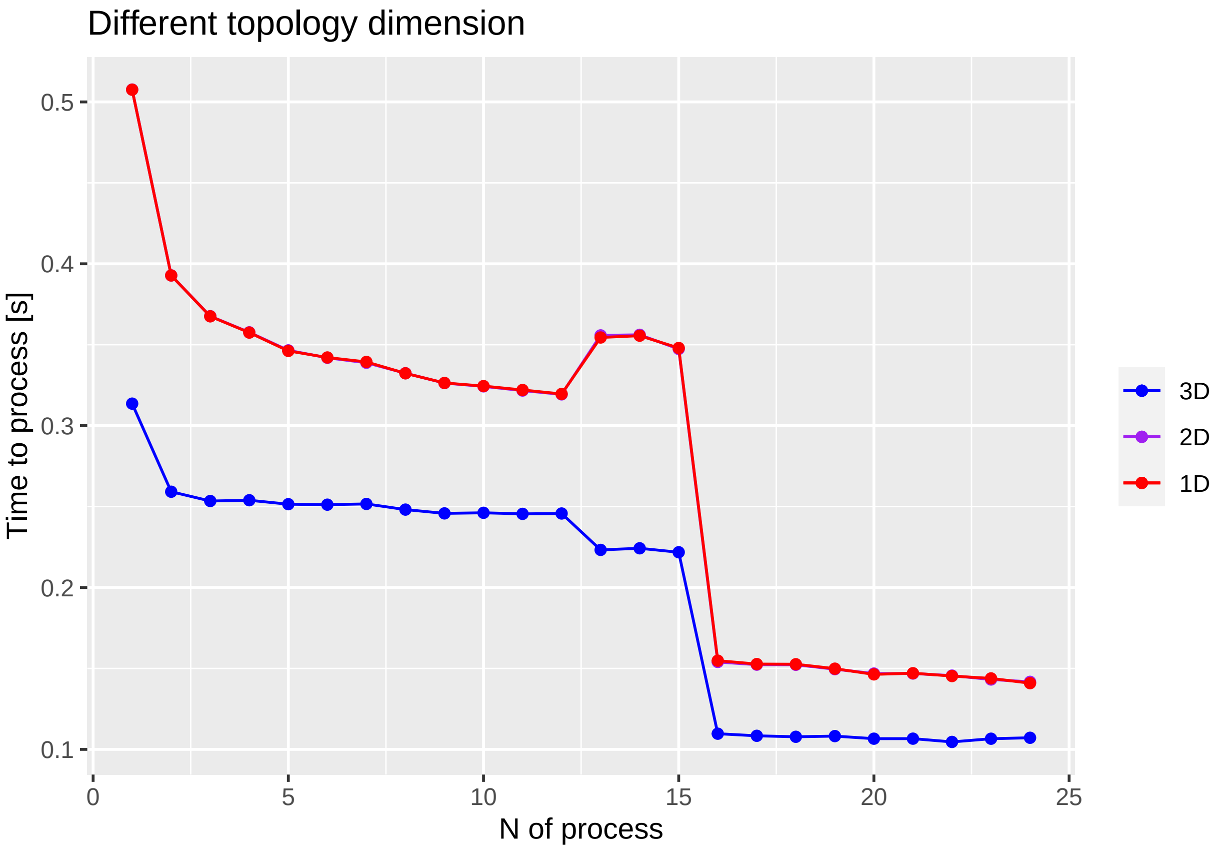
I produce two different plots: one that represent the time to compute the matrix-matrix sum for different shapes and one with the same shape but with different mapping.



In this graph, we can see that if we use different shapes, that are 2400x100x100 (red), 1200x200x100 (orange) and 800x300x100 (yellow), the time to sum the matrix is always the same, even using multiple processes. We note that for the first 15 processes executed in parallel, the time is quiet constant, but after 16 processes in parallel, the time to compute decrease drastically and remains constant until the end.



Here, there is the plot of the same matrix (1200x200x100), with the same topology (3D) but with different mapping and the result doesn't change. In fact, the time to process is always similar.



Lastly, I computed the time to do the matrix-matrix sum with the same mapping and the same shape (1200x200x100) but with different topology. The 2D and 1D topology are the same, even because the matrix is a 3D array and the two topology are built with Nx, that is tha number of processes ($[size]$ for 1D and $[size, 1]$ for 2D, as size the number of processes).