Using MPI to efficiently distribute GEMM computations – Master (2 person job)

As you have implemented several variations of a distributed GEMM algorithm, you are able to benchmark them.

The script `bench.sh` produces execution speed measures stores in `bench.csv`[1].

> You have, of course, to write other scripts according to your plan(s) of experiments and also use your own visualization solutions (python?) to display the results.

The benchmark does not test scalability as handed out to you: a number of nodes is chosen -p and q - and the problem - the values of m, n, k - is changed to observe a comparison of these algorithms.

Your goal is to assess how weakly (**OR** strongly)[2] scalable the different algorithms are and whether their different approach (collective operations, non blocking communications) have a strong impact on performance.

In order to meaningfully assess the scalability of the algorithms, you should carefully chose values of m, n and k as (**OR** while) you increase p, q. The choice of these values can be motivated by analyzing the distribution and the complexity of the computation over the machines. Your analysis can be completed by observing `VITE` traces to ensure the performance and behavior of the algorithms are coherent.

This analysis (how and what you plan on experimenting, results of your experiments, conclusions from your results) should be reported in a **15 pages document (max)**.

NOTE I: the source file (`main.c`) is able to read the 3 dimensions of the problem rather than decide $m = n = k$ as we did during the hands-on. This should ease the choice of m, n, k to match with p, q[3].

NOTE II: with Simgrid, you can easily change your computational platform. It could be interesting to compare the behavior of the three implementations against the platforms (see directory *platforms*).

Deposit terms

You must submit your report (with the names of both students) on Moodle before January 30.

[1] Read it carefully to understand what are the results it produces

[2] See the slides of Introduction to this course by Alfredo Buttari

[3] Be careful: because for the hands-on, $m = n = k$, perhaps, if you now take different values, there will be some errors with your code. You should check the dimensions carefully for each operation, because now these dimensions can be different