

## Assignment #7, Module: MA5621

---

Gustavo Ramirez

March 22, 2017

# 1 PART 1

## 1.1 MAKEFILES

In the sub-directory './makefiles/' can be found the 4 combinations of compilers and BLAS libraries; in this case, time only permitted to run the basic 4 combinations, gcc with ACML and MKL, and intel11 with ACML and MKL.

## 1.2 ON THE AUTOMATION SCRIPTS (BASH-PYTHON SCRIPTS)

As mentioned before, only the 4 basic combinations (gcc with two flavours of BLAS, and intel11 with same two flavours of BLAS) were tested.

**Although only those 4 were executed, a few Bash and Python scripts were written to automate the execution for all combinations for each number of cores.** The way I executed it was simply by submitting a job through 'sbatch sbatch.sh' (note that sbatch.sh is located in ./src/; **instructions on how to use those scripts are in ./src/README**). The most significant point here is that those scripts allow for the execution using gcc and all intel versions (11, 12, 15).

The makefiles generated from executing ./src/general.sh don't include the modules initially, so added them at the end by hand, although it is very easy to automatically add them from that Bash script.

## 1.3 ROUGH CALCULATION OF $N$

Executing 'free -m | grep Mem', from that line is seen that the available RAM is 7869 B. Then (through a very informal calculation, syntax-wise):

$$N^2 \cdot 8 \sim 0.8 \cdot RAM = 0.8 \cdot 8GB \Rightarrow N \sim 2.8 \cdot 10^4$$

# 2 PART 4

The results for performances can be found in the file performances.csv, for the four basic simulations.

Also, some extra time allowed to run a few executions for 16 cores; the results are shown in the file slurm-outputs/slurm-245975.out, where evidently the performance doubles (as the number of cores doubled).

### 3 PART 5

For arch=lonsdale\_gcc\_openmpi\_mkl, the bigger the value of NB, the better the performance; of course, within the four values tried for NB, and this is in average, as the 3rd value in ascendant order gave the best results, i.e. NB = 140. Even more, in general NB = 140 gave the best results.

BLAS-wise, ACML showed the best performances.

Also, gcc gave slightly better results than icc.

In general, the submission of jobs to the queueing system ran smoothly, no problems at all, due in great part to the implemented Bash and Python scripts (located in ./src/).