Questions and Answers

Last updated on 14 August 2015, at 12:00.

1. Benchmark document and general questions

- **Q#1** : Chap 1.1: For HPL and HPCG kernels, can we use the optimized version in order to get best performance? Or do we have to use the public implementations from netlib.org and hpcg-benchmark.org?
- **A#1** : Any optimized version of the above kernels could be used, providing that we can also test those versions (namely the optimized version will be released to Cineca for testing purposes).

- **Q#2** : Chap 2.3 : in the table #6, what "suggested nproc" stands for?
- **A#2** : "nproc" here reads "ncores", i.e. the total number of cores used for the simulation. Please see also Section 2.4 rule #5.

- **Q#3** : Chap 2.4. rule #5. "Each program must be executed a predefined number of times with the pertinent datasets." How many times each program has to be executed?
- **A#3** : Each program must be executed at least a number of times as reported in the README file. Usually at least four times for the scalability plot.

- **Q#4** : Chap 2.4. rule #7. "If you use floating point accelerators, you should report the performance results obtained both with the accelerators and without the accelerators using the same number of nodes". Are accelerator results optional or mandatory ?
- **A#4** : If in your offer are planned accelerators then accelerated results are mandatory only for the Quantum Espresso benchmark, and they are optional for the other codes.

- **Q#5** : Chap 3: In the table #10 is it "ncores =" instead of "nproc"? You are saying in the text that we "must report the performance obtained relative to the base number of nodes/core reported in table 6" but you are using nproc (= number of processors) ?
- **A#5** : See Q#2, "nproc" hear reads "ncores".

- **Q#6** : It is said that the benchmarks have to be run on nodes "configured as the proposed solution"? What if the proposed solution is not yet available?
- **A#6** : If not yet available, please report results extrapolated for the proposed solution.

- **Q#7** : Can turbo mode be ON?
- **A#7** : Yes, please report the turbo status (ON/OFF) as a comment together with the results.

- **Q#8** : How many benchmarks do I have to perform for each application?
- **A#8** : Please follow the instruction reported in the README files. For better clarification see the related question in the application sections here below:
    - Section 2 Q#2 for Quantum Espresso;
    - Section 3 Q#3 for OpenQCD;
    - Section 4 Q#1 for MILC;
    - Section 5 Q#1 for GENE;
    - Section 6 Q#2 for GYSELA.

2. Quantum Espresso

https://hpc-forge.cineca.it/files/gara_Tier0_2015/public/QuantumEspresso/README

- **Q#1** : In the README file the suggested command lines to execute QE benchmarks report values for -npool greater than 1 ('-npool 2' and '-npool 8'), but for both the simulation k-point is set to 1. This could lead to the following warning message: "some nodes have no k points". Is that correct?
- **A#1** : It is a typo, please consider the command line as in table #6 of the benchmark document. For the benchmark GRIR686 are allowed values of 1 or 2 for -npool, since the simulation is spin polarized.

- **Q#2** : "Please, for each dataset, provide us at least four results including the reference configurations (16 cores and 512 cores, or the maximum configuration available)". What is the reference configuration? Is it the configuration used for the benchmark?
- **A#2** : Here we are asking to provide 2 sets of results of at least 4 values each. The first set, for the small dataset, has to contain a result for 16 cores. The second set, for the big dataset, has to contain a result for 512 cores (again, please see Section 2.4 rule #5 of the benchmark document) or, in case this is not possible, the maximum core count available.

3. OpenQCD

https://hpc-forge.cineca.it/files/gara_Tier0_2015/public/OpenQCD/README

- **Q#1** : The reference results provided - what system was used to obtain the results?
- **A#1** : IvyBridge (PICO - http://www.hpc.cineca.it/content/pico-user-guide)

- **Q#2** : From the output of "check2" and "check5" seems the accurancy validation tests are not passed, even if compiler performance optimization are turned off, are we facing a validation issue?
- **A#2** : No. Our preliminary check on this issue has found that the statement appearing in the output to be misleading. Values of the order of magnitude of those reported as reference are considered valid.

- **Q#3** : The benchmark document reports 32 and 512 cores as reference configurations, but the README file says “Please provide us the result obtained for a run on 32 cores and at least other 3 values including the maximum available configuration." How many benchmarks do I have to perform and what is the number of cores with which we have to run these benchmarks?
- **A#3** : The runs on 32 and 512 cores are reported in the OUTPUT_DATASET directory as reference configurations. What we need is a weak scalability test with at least 4 point including the reference configurations or the maximum available configuretion if 512 core are not available.

4. MILC

https://hpc-forge.cineca.it/files/gara_Tier0_2015/public/MILC/README

- **Q#1** : "Please, for each dataset, provide us at least four results including the reference configurations (16 cores and 512 cores, or the maximum configuration available)". What is the reference configuration? Is it the configuration used for the benchmark?
- **A#1** : Here we are asking to provide 2 sets of results of at least 4 values each. The first set, for the small dataset, has to contain a result for 16 cores. The second set, for the big dataset, has to contain a result for 512 cores (again, please see Section 2.4 rule #5 of the benchmark document) or, in case this is not possible, the maximum core count available.

5. GENE

https://hpc-forge.cineca.it/files/gara_Tier0_2015/public/GENE/README

- **Q#1** : The parameters file contains a setup to run a strong scaling for multiples of 576 cores. Below, it says: 'please provide us at least four results including the reference configuration (512 cores ...)'. Can you specify exactly which core counts we should use?
- **A#1** : It is a typo, please consider the input file provided. For this benchmark provide us at least four results including the reference configuration at 576 cores.

- **Q#2** : Are we allowed to modify the "&parallelization" namelist in the parameters file?
- **A#2** : You are allowed to modify the parameters input file to obtain at least other 3 results as reported in the README file.

- **Q#3** : The domain decomposition specified in the 'parameters' file is chosen to work well with 36 tasks/node (n_procs_x is set to 18, so the communication in x direction is intra node). For different numbers of cores per node (e.g. 28 on our benchmark system), the n_procs parameters have to be adapted to get good performance. Are we allowed to change the n_procs freely?
- **A#3** : We suggest to try to relax the parallelization parameters to get the best combination from the performance point of view. Namely:

  ```
  &parallelization
  n_procs_s =  -1  ! can be either 0 or any negative number for relaxed parallelization
  n_procs_v =  -1
  n_procs_w  = -1
  n_procs_x = -1
  n_procs_y = -1
  n_procs_z = -1
  min_npx=16  ! constraint in x direction
  min_npy=4    ! constraint in y direction
  /
  ```

  With the above configuration, the parallelization is given free with two constraints: 1. Number of processes in x direction must be equal to or larger than 16 (it only works for integer divisors of the 288 points in this direction) 2. Number of processes in y direction must be equal to or larger than 4. The auto-parallelization will sample all the possible configurations. As there are sometimes a large number of possible parallelizations this can take a long time. GENE tries already to skip parallelizations which are expected to be slow, from some heuristics of previous runs. One could reduce the number of allowed combinations by additional setting of max_np* variables: for each direction only the valid min_np* to max_np* numbers are tested. After having launched a test with the above configuration, you can find all the configurations sampled and their performance in the file autopar.dat. For example, one could be:

  parallelization: 2 1 1 18 4 4 nblocks: 18432

  then you can modify the parameters file accordingly (in the example above n_procs_s =2, n_procs_v=1, n_procs_w=1, n_procs_x=18, n_procs_y=4, n_procs_z=4) .

- **Q#4** : The following message appears in console output:

  ```
  WARNING: Von-Neumann b.c. not implemented yet for GDAGGER_G
  compile with #undef GDAGGER_G in switches.h
  ```

  Actually GDAGGER_G is defined in switches.h. Should we undefined it?
- **A#4** : Do not change anything in the switches.h, because this will alter the results. You can safely ignore this warning for the benchmark purpose, it is only relevant for physical production runs.

- **Q#5** : The following message appears in console output:

  ```
  ################################################################
  WARNING: Memory per core neither set in makefile
  nor environment variable MEMORY_PER_CORE is set!! Using 500 MB
  ################################################################
  Using a maximum of   500.00 MB per core.
  ```

  Should we set appropriate value for MEMORY_PER_CORE?
- **A#5** : Yes, it is possible to set a default value in the makefile setting the variable MB_PER_NODE, or set the environment variable MEMORY_PER_CORE just before to launch the executable. The values have to be set in MB.

6. GYSELA

https://hpc-forge.cineca.it/files/gara_Tier0_2015/public/GYSELA/README

- **Q#1** : On GYSELA README, it is written that: ""It is possible to change the number of threads inside each MPI process in modifying simultanously ""bloc_phi"" and ""Nbthread"" variables in the DATA file. The acceptable values for the number of threads are: 8, 16, 32.""
  Could we used a number of threads per MPI task different then 8, 16, 32? For example for a 14 cores CPU, could we use a number of threads equal to 7 or 14 and run on 504 or 511 cores instead of 512?
- **A#1** : The allowed number of threads per MPI task that you can set with the ""bloc_phi"" and ""Nbthread"" variables are the common divisors of NPHI (64) and NTHETA (128). For the strong scaling benchmark, the number of cores actually used can be less than 512 (for

the reference configuration), but the result must be associated to 512 cores.

- **Q#2** : For the GYSELA code, is it mandatory only the STRONG scalability test?
- **A#2** : Yes.