1 Background Information

The hands-on session is aimed to be performed on the hybrid system presented in class and schematically illustrated in the lecture's slides. The objective of this session is to acquire manual skills with the software environment required for enabling compute simulation on such systems. Particular focus is given to performance analysis of the results obtained while running similar codes on different devices. Simple codes for matrix-matrix multiplications are provided. Most of the codes are designed to perform the matrix-matrix multiplication calling the GEMM funtion [1]

2 The Software Environment

At login only the GNU compiler is available. The CUDA Toolkit [3] and the Intel compiler [5] were also installed. To set a different software environment you are request to use the "module" command. Basic information about how to use the module command are available within the Documentation web page of the HPC facility at ICTP [6]. All the provided sources and the Makefile were installed within the /opt/ex-lab directory. Please, copy it on your local directory. It contains files as listed below:

bench_dgemm_cpu.f90 - the files contains a simple F90 source code which includes a call to the DGEMM function [2]

bench_dgemm_gpu.f90 - the files contains a simple F90 source code which includes a call to the CUBLAS version of the DGEMM function. This file is proposed to show how the CUBLAS library can be directly linked to a source code using the cublas thunking interface. In this case the call to the function implementation manages all the data transfer between CPU and GPU on bot directions

bench_dgemm_cublas_gpu.cpp - the files contains a c++ source code provided by NVIDIA [4] which includes a call to the CUBLAS version of the DGEMM library along with all the data transfer management

bench_dgemm_cuda_gpu.cu - the files contains a CUDA source code provided by NVIDIA [4] to perform a matrix-matrix multiplication with data allocated on the GPU device

Makefile - makefile created to compile all the above proposed source codes

As presented during class the Resource Manager Torque is configured with three different queues: cpu-only, gpu and phi.

3 Exercise Description

The exercise consists in compiling all the available codes using the provided Makefile. Create a the submission script to submit in batch mode each of the

obtained binaries to the appropriate queue. Extract and compare the results obtained in term of time to solution.

NOTE: Thread scaling is obtainable setting the environment variables OMP_NUM_THREADS and MKL_NUM_THREADS at the desired number of threads.

4 Advanced Exercise Description

Scale the matrix size and analyze scaling behavior. Download and compile other open source multi-threaded library, implementing GEMM operations, and compere the result obtained.

References

[1] http://en.wikipedia.org/wiki/General_Matrix_Multiply

[2] http://www.netlib.org/lapack/explore-html/d1/d54/group__double__blas__level3.html

[3] https://developer.nvidia.com/category/zone/cuda-zone

[4] http://docs.nvidia.com/cuda/cuda-samples/index.html

[5] http://software.intel.com/en-us/intel-compilers

[6] http://argo.ictp.it/documentation/using-201cmodule201d-command