D3.2 Applications porting and tuning reports
Version 1.0

Document Information

Contract Number	610402
Project Website	www.montblanc-project.eu
Contractual Deadline	M21
Dissemination Level	RE
Nature	Report
Coordinator	Nico Sanna (CINECA)
Contributors	Piero Lanucara (CINECA), Jose Gracia (HLRS), Simon McIntosh-Smith (BRISTOL), Diego Nieto (BSC), Filippo Mantovani (BSC)
Reviewer	Claudine Pelegrin-Bomel (Bull)
Keywords	Kernels, OmpSs, ARM, Porting, Tuning

Notices:
The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 610402.

Change Log

Version	Description of Change
v0.1	Initial Draft released to the European Commission
v0.2	First proposed template to the partners of the Consortium
v0.3	First proposed draft after contributions from the partners of the Consortium
v0.4	Version reviewed by the internal reviewers
V0.5	Version reviewed and restructured by Filippo and Diego
V1.0	Final version for the European Commission

Table of Contents

1. Executive Summary 4
2. Platforms 5
3. QuantumESPRESSO 6
4. COSMO OPCODE 9
5. BUDE 10
6. ROTORSIM 11
8. LBC 13
9. HTC for Bioinformatics 16
10. Cannon benchmark 18
11. Conclusions and next steps 20
12. Acronyms and Abbreviations 21

1. Executive Summary

In this document *D3.2 Applications porting and tuning reports* the activities related to T3.1 and T3.2 and T3.4 of the first 21 months of the Mont-Blanc 2 project are given in detail. During the same period also a limited part of the activities related to T3.3 (Application benchmarking) have started in order to preliminarily assess the code versions ported to the platforms made available to the consortium partners (see next section for platform disambiguation). The support activities related to T3.4 have produced significant help in the porting of Mont-Blanc applications thus paving the route for its repeated use in T3.1 and T3.2 of Mont-Blanc2.

Two Mont-Blanc applications (COSMO and QuantumESPRESSO) have been tested and preliminary optimized on the Mont-Blanc prototype where a set of three benchmarks have been installed, executed and benchmarked. This work had some overlap with Mont-Blanc project, however the same tests have been carried out on Mont-Blanc 2 mini-clusters.

Among others, we would highlight the following significant results for the WP3 activities in this period:

- For QuantumESPRESSO and COSMO, we produced for the first time a set of Extrae traces running on the preliminary and final version of the Mont-Blanc prototype and tested the initial versions of the new FFT and ZGEMM functions.
- LBC code is now fully ported to MPI/OmpSs and has been comprehensively profiled and benchmarked on the XC40 system at HLRS; now the code is ready to be migrated to the full version of the MB prototype;
- BUDE's OpenCL implementation has been optimised, now achieving 50% of peak performance on an Nvidia GTX680. BUDE's OpenCL version has now been tested and benchmarked on a wide range of embedded GPUs, from vendors including ARM, Imagination and Qualcomm and is now ready to be benchmarked on the new Mont Blanc prototype;
- ROTORSIM's functionality continues to be ported to OpenCL, and more of its features are now accelerated compared to the initial version of the code. The Domain Specific Language (DSL) for ROTORSIM is currently in development but the present version of the code was able to run up to 4096 GPU nodes on Piz Daint (CH) and Titan (US) supercomputer with almost linear weak scalability.
- We designed a feasible computing workflow of a non-HPC application in the domain of the HTC for bioinformatics with the release of a virtualized pipeline as a template for the Next Generation Sequencing (NGS) Whole Exome analysis. This has been tested so far on standard HPC architectures, and will be deployed on the Mont-Blanc 2 mini-cluster in the near future.

2. Platforms

For this deliverable we refer to two class of platforms:

1. The Mont-Blanc prototype deployed at BSC in the framework of Mont-Blanc project (during M14 and M18 of Mont-Blanc2 project)
   More details about this platform are available on: https://wiki.hca.bsc.es/wiki/MontBlanc
2. Mini-clusters. These platforms will be installed during the whole duration of the project and are subject to evolve. For this deliverable the two mini-clusters tested were:
   - Odroid XU (24 nodes)
   - Nvidia Jetson TK1 (8 nodes)

   More details about mini-clusters can be found here:
   http://montblanc-project.eu/arm-based-platforms

Since the major objective of WP3 is to analyse the programmability and performance of the hardware and software systems developed in this project, this deliverable will report activities performed during the first 21 months of the project on the platforms listed above as they were implemented and made available to the partners of the project.

3. QuantumESPRESSO

Partner: CINECA
ARM Cluster: BSC (full and partial MB prototypes, Jetson TK1 cluster)
Kernel: ZGEMM/BLAS, FFT
Domain: Mont-Blanc Kernel/Application

Performed Activities

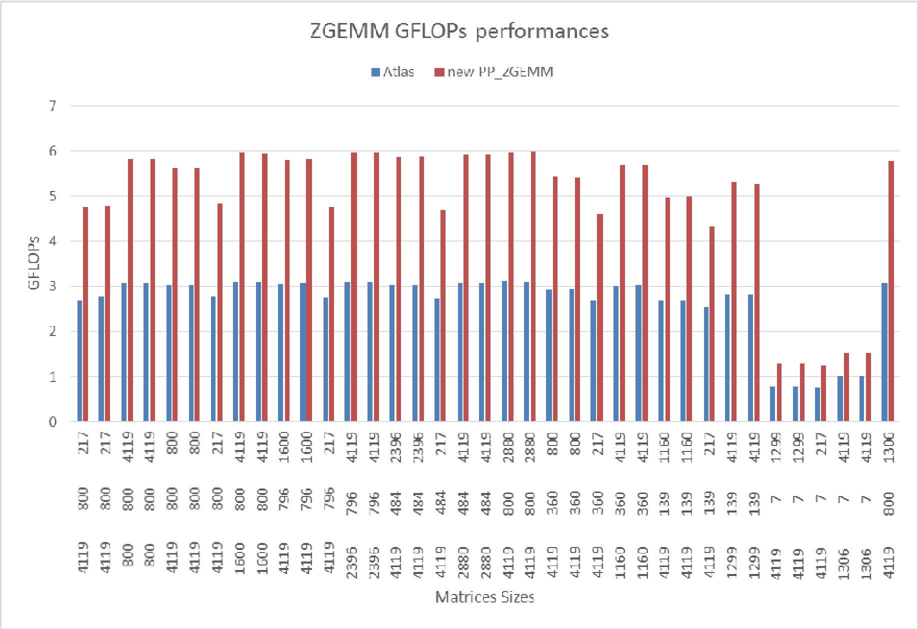
The QuantumEspresso package was successfully ported to full Mont-Blanc prototype. Previous work on partial machine was used to reduce the computing work in order to have the codes running successfully and efficiently on the new environment (both as SMP and OpenCL tasks). After the initial porting, we have prepared three version of codes: MPI (used as reference), SMP (only MPI+OmpSs) and OCL (MPI+OmpSs+OpenCL). The FFT and GEMM part can now be used in SMP mode, while the complete porting (MPI+OmpSs+OCL) is still ongoing. Scalability tests of the application were carried out, while correctness tests in term of numerical accuracy of all the 3 versions were performed but limited to reproduce some small size benchmarks on the smaller clusters. Regarding the prototype development environment, the new setup of the full Mont Blanc prototype was tested. Some problems occurred during the configuration of the application on the new environment were fixed at compilation and run time. Modules, GNU compilers, Mercurium, Nanos++, libraries (ATLAS, Extrae, Paraver, FFTW) were particularly stressed during compilation, runtime linking and execution. Command queue scripts for scalability tests and energy to solution measurement on compute nodes were developed and used.

Version of Environment tools

Mercurium (1.99.4), Nanos++ (0.7.4), Extrae (3.0.1) and GCC (4.9.1 release). Atlas (atlas 3.11.31_lapack) were used.

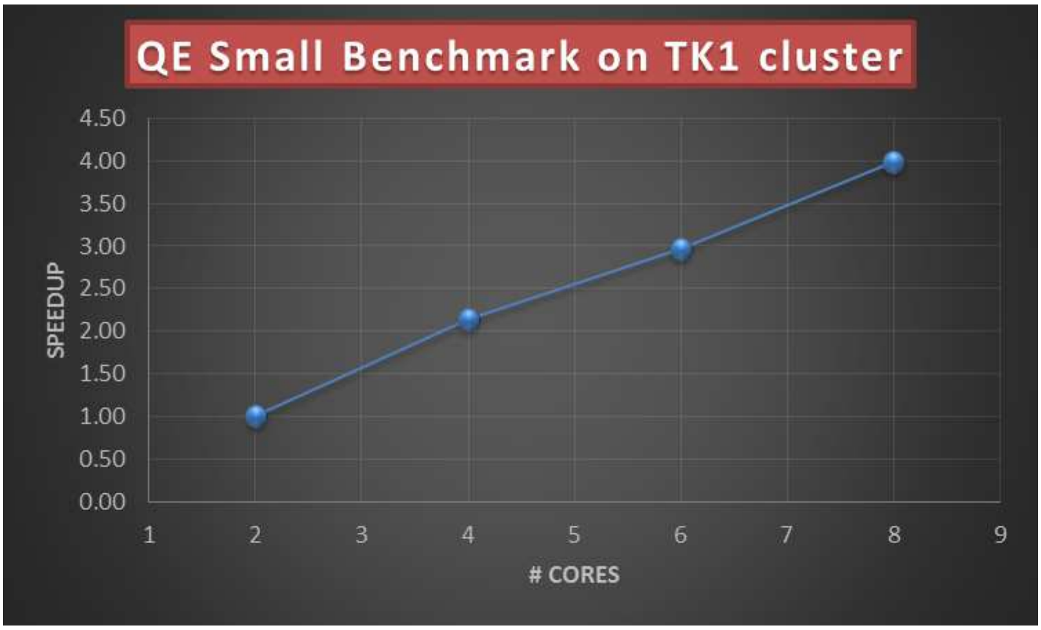
Significant Results

The QE package was preliminarily ported and tuned on a partial Mont-Blanc prototype and we used the PWSCF (***pw.x***) code as testbed for our optimization of the most time consuming part of the code: the FFT and ZGEMM set of functions. The initial approach to improve GEMM operations in ***pw.x*** was first to isolate the ZGEMM calls within the PhiGEMM and then to replace it with a stubbing function in FORTRAN performing BLAS operations in standard non-optimized mode. Then, this prototype version of ZGEMM has been taskyfied looking at the operations able to asynchronously run in parallel (with Nanos++ runtime) without any race condition or data conflict. Although we are still working on an efficient OpenCL implementation of DGEMM in the newly developed PP_ZGEMM, the OmpSs version running in SMP mode is able to get up to 90% of two ARM cores in Mont-Blanc SDB as reported in the following figure in more detail:

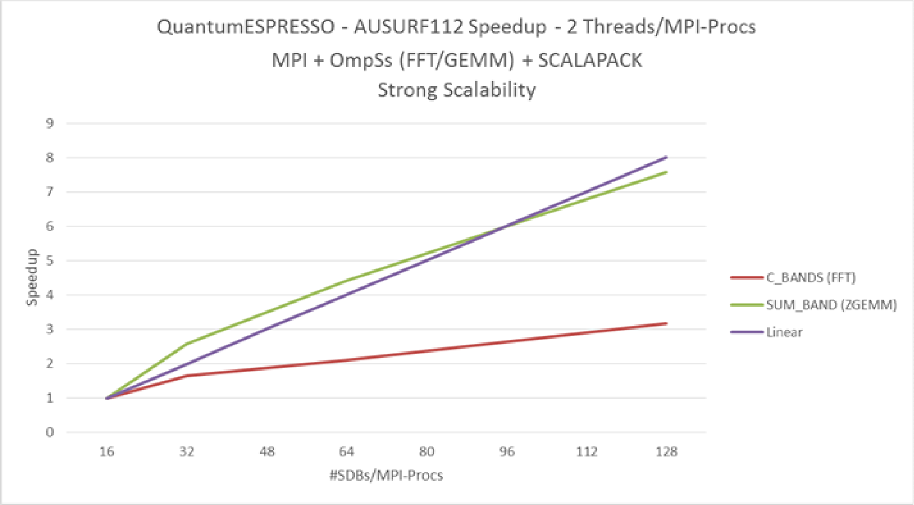


Regarding the FFT part of PWSCF, the fft_scatter routine has been modified by introducing a new communicator (*comm_dup*) capable to generate two *mpi_alltoall* from different iterations at the same time without any conflicts with the other iterations running. To implement this model of duplicated *mpi_alltoall*, we added in *fft_scatter* routine a new optional variable iter which accounts the FFT iteration index in play. Then, the new fft_scatter is called within *invfft* and *fwfft* driver functions. As a matter of fact, by using the newly developed FFT model of computation, we gained a speedup up to the maximum number of OmpSs threads per SDB set by the environment variable NX_THREADS (in our case NX_THREADS=2).

Scalability tests of PWSCF were performed on Jetson TK1 (JTK1) cluster as well as on the partial and final Mont-Blanc prototype(s). Strong scalability benchmark on JTK1 reported in the following figure clearly shows its almost linear behavior on small input set (Si18)



while the same test (using a larger input dataset – AUSURF112) ran on the final Mont-Blanc prototype shows a drop in performance after 32/64 nodes mostly due to the FFT part of the code:



Future Plans

We are still working on the substitution of unoptimized BLAS (*dgemm* within *zgemm*) calls in PhiGEMM library used by QuantumEspresso, with a more performant OpenCL version. Porting and optimization of part of this library is in progress and we expect to use this or similar OpenCL libraries also for the FFT porting in QE. When a stable and fairly performant version of QE will be obtained, we will perform a comprehensive profiling with the collection of a set of Extrae traces (reference MPI, SMP and OCL) in order to best tuning the code(s) on the Mont-Blanc/2 prototype(s).

4. COSMO OPCODE

Partner: CINECA
ARM Cluster: BSC/Bull
Kernel: Dynamics/Physics
Domain: Mont-Blanc Kernel/Application

Performed Activities

The OPCODE/COSMO original code structure was ported to full Mont-Blanc prototype. Previous version working on partial Mont-Blanc prototype machine was used to have the codes working successfully on the new environment.

After the initial porting, three version of codes should be used: MPI (used as reference), SMP (only MPI+OmpSs) and OCL (MPI+OmpSs+OpenCL) for the dynamical core and for the physics part of OPCODE. Himeno Benchmark was used to test the dynamical core part of OPCODE in order to boost its development.

Preliminary scalability tests of the Himeno Benchmark (MPI, SMP and OCL) were carried out while correctness tests in term of numerical accuracy of all the 3 version of COSMO OPCODE will be performed up to a large number of Mont-Blanc nodes (SDBs).

OmpSs support has been provided by BSC during the whole duration of the porting operations. This included hands-on sessions with CINECA during the week of 1-5 Sept. 2014 for the finalization of the porting and optimization of the medium-size kernel applications of COSMO and QuantumESPRESSO.

Version of Environment tools

As reported above for QuantumESPRESSO.

HDF5 and NetCDF libraries were compiled during the porting of OPCODE to the full Mont Blanc prototype.

Significant Results

The OPCODE/COSMO code structure was preliminarily ported on partial Mont-Blanc prototype. Himeno Benchmark was tuned on the full Mont-Blanc prototype (previous tuning on Eurora was only partially used due to the different computing architecture of EURORA machine in CINECA with respect to the Mont Blanc prototype). To this end, an initial tuning was performed varying the size of the input problem and performing some preliminary weak scalability tests aimed at preventing performance bottlenecks (due to I/O and collective message passing phases) during the benchmark execution.

Future Plans

Complete the evaluation of the Mont-Blanc prototype focusing on OPCODE tuning by improving the dynamical core scaling and better deploying some routines of the physics part at present at the status of initial porting on OmpSs+OpenCL.

5. BUDE

Partner: BRISTOL
ARM Cluster: In-house systems
Kernel: BUDE
Domain: New Mont-Blanc2 HPC Application

Performed Activities

The work to port BUDE to OmpSs it is just started at M18, but in the meantime, BUDE's OpenCL implementation has been further optimised, now achieving 50% of peak performance on an Nvidia GTX680. In addition, a cut-down version of the code has been produced which reduces the dependency on OpenCL from v1.2 to v1.0, in order to enable testing of the performance of the main OpenCL kernel on FPGA-based platforms. BUDE's OpenCL version has now been tested and benchmarked on a wide range of embedded GPUs, from vendors including ARM, Imagination and Qualcomm.

Significant Results

BUDE's main OpenCL kernel has been demonstrated on an Altera FPGA and has achieved respectable performance and performance per watt: an Altera Stratix V A7 achieves 3.04 billion atom-atom interactions per second for the cut-down version of BUDE. This is comparable to an Intel integrated graphics processor, such as an Intel HD graphics 4600, which achieves 3.07 billion atom-atom interactions per second. An Nvidia Kepler K600 Quadro (a discrete embedded GPU for laptops) achieves 3.24 billion atom-atom interactions per second. This Altera Stratix V A7 is quite a modest FPGA and does not have many on-chip resources, so this is an impressive result, especially considering the relative immaturity of their very new OpenCL drivers.

Future Plans

It is planned to port BUDE to OmpSs during the second half of 2015. We also plan to benchmark the current OpenCL version of BUDE on the new final Mont Blanc prototype in 2015.

6. ROTORSIM

Partner: BRISTOL
ARM Cluster: In-house systems
Kernel: ROTORSIM
Domain: New Mont-Blanc2 HPC Application

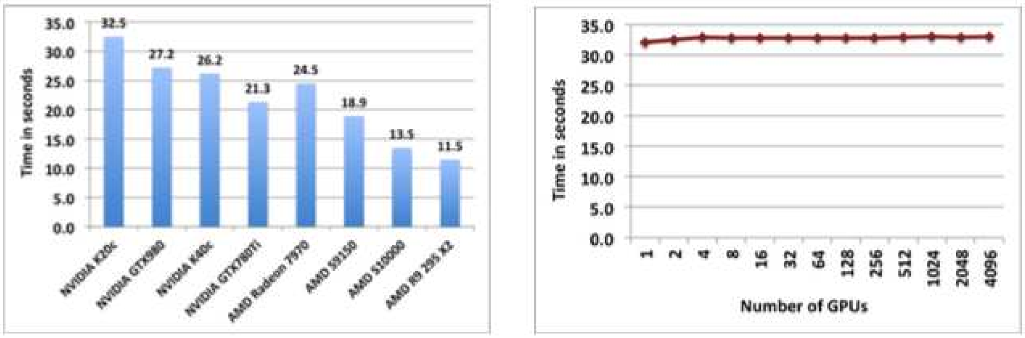
Performed Activities

ROTORSIM's functionality continues to be ported to OpenCL, and now most of its features have been accelerated compared to the last report. Further optimisations have also been applied, and the code is now faster in many more cases than before. ROTORSIM's latest OpenCL feature set has been developed to be more performance portable, and these results were published in a paper at ISC 2014, Leipzig.

In order to be able to compare and contrast the performance of the ROTORSIM OpenCL code to other versions, such as CUDA or OpenMP, a Domain Specific Language (DSL) for ROTORSIM is currently in development. This looks very like OpenCL, but abstracts away some of the details and leaves a very simple language for the expression of computational kernels for CFD applications such as ROTORSIM. We have a prototype compiler which then takes these simple, language independent CFD kernels and from then generates OpenCL, CUDA or OpenMP.

Significant Results

ROTORSIM has been successfully run on Piz Daint at CSCS in Switzerland, currently the fastest supercomputer in Europe. Piz Daint is a 5,500 node Cray XC30, complete with one Nvidia K20X GPU per node. ROTORSIM was successfully executed across 4,096 nodes (and GPUs), and showed nearly perfect weak scaling. We had hoped this would be possible but it was gratifying to achieve, and demonstrates that the development work invested in ROTORSIM is positioning it well for Exascale. These results will feature in an upcoming paper at ParCFD in Montreal, May 2015. Below we show, on the left, relative performance results for ROTORSIM, comparing a range of OpenCL-capable devices (times per iteration are shown and so shorter bars are better), while on the right we show the near perfect weak scaling results on up to 4,096 GPUs on Piz Daint.



(a) Performance on various GPUs.

(b) Weak scaling on Titan.

We submitted an abstract for a follow-on ROTORSIM paper at AIAA Scitech next January 2016. As part of this we've had ROTORSIM running on up to 4,096 nodes of Titan, the largest GPU machine in the world (Cray XK7 at Oak Ridge, the nodes each have one 16-core AMD Opteron and 1 Nvidia K20X GPU). Weak scaling was near perfect which was as good as we could possibly have hoped for the present version of the code.

Future Plans

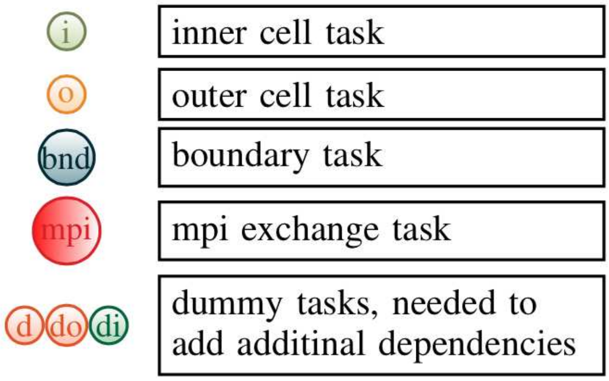
We are working to extend ROTORSIM's DSL and code generator to emit 7.code. This work will be carried out in the second half of 2015. We plan to benchmark the OmpSs/OpenCL version of ROTORSIM on the new final Mont Blanc prototype also in second half of 2015. A second bigger scaling run on the largest supercomputer in the US (Titan at Oak Ridge) is planned during second half of 2015. Titan includes over 18,000 Nvidia K20X GPUs, so this will be a very significant scaling test for ROTORSIM in its preparation for Exascale.

8. LBC

Partner: HLRS/USTUTT
ARM Cluster:
Kernel: Lattice Boltzmann
Domain: New Mont-Blanc2 HPC Application

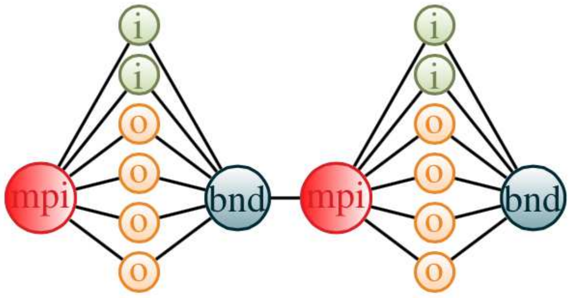
Performed Activities

Starting from a previously developed and reported version we fixed some problems in cooperation with BSC (these problems were related to the Fortran Mercurium compiler an the usage of modules in Fortran and the priorities of tasks) to get a hybrid MPI/OmpSs version.

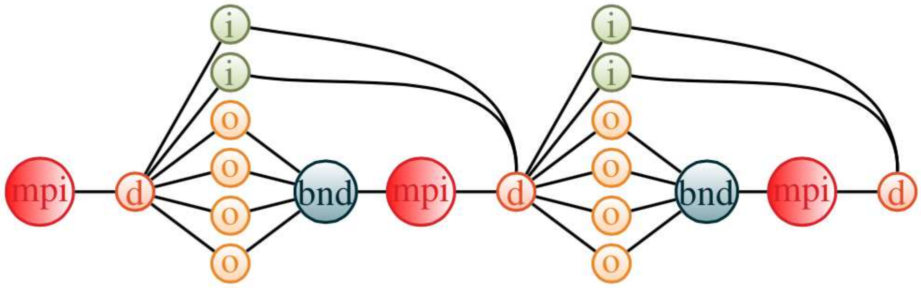


At the base of this we developed three different versions. In the figure at the left side the different tasks are explained. Only the outer cells (o) are needed for the boundary calculation (bnd) and the MPI exchange (mpi).

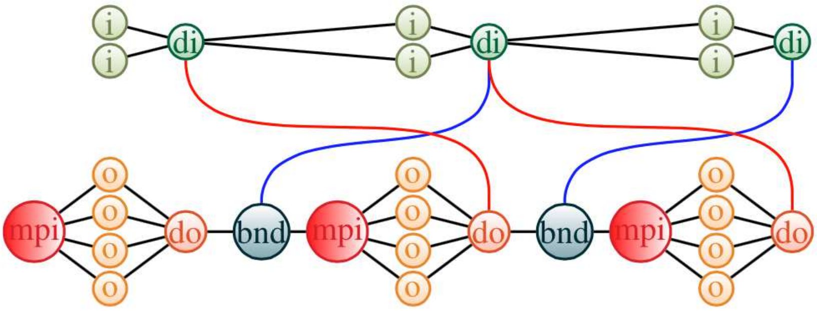
First a version we called “fork & join” because after every iteration we have to fork. In this version we are not overlapping communication and computation, we used this version later on as reference for benchmarking and tuning.



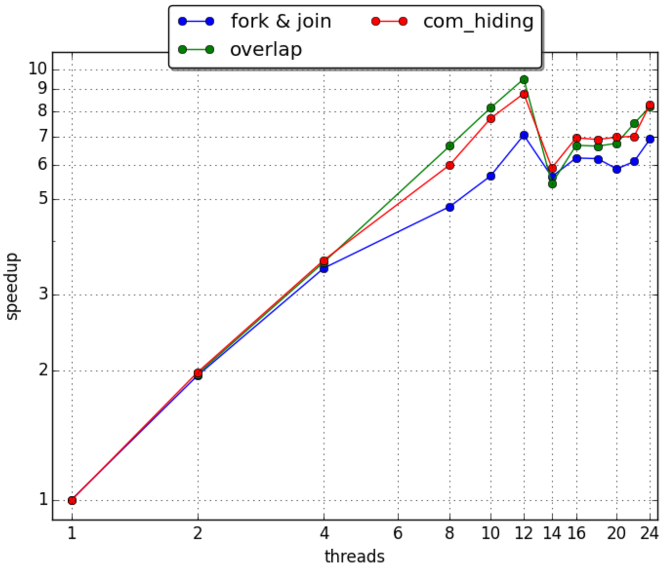
Second a version we called “com_hiding” (communication hiding), this version is similar to the version we started with. In this version we are overlapping communication and computation, also we additionally introduced priorities. This was necessary to be sure that the critical path (outer tasks, MPI task, bnd task) is executed.



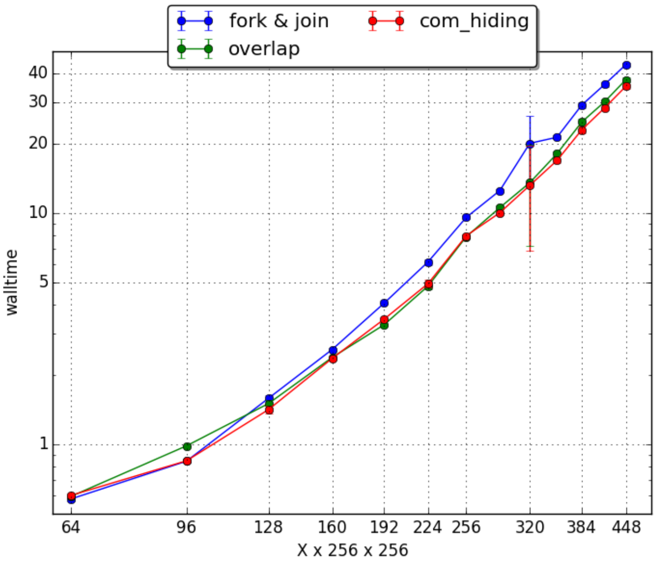
The third version we called “overlap”. In this version we used the native double buffering design of the LBC code. This allows us to overlap two iterations and completely close the gaps (idle time) between the iterations. This allows us to generate an additional buffer, which is useful, if there is a huge delay in the MPI communication.



In the next step we benchmarked the three different versions on our Cray XC40. We used two nodes each of them has 24 cores in 2 sockets. We calculated the speedup to our reference implementation at one thread. On one socket (12 threads) we can get with our overlap implementation a linear speed-up. After 12 threads there is an issue with NUMA effects, which we have to fix in a future version. This effect may only happen on your XC40, because Nanos++ runtime is detecting the node as a 24 core one socket system.



As a second benchmark we increased to domain size from 64x256x256 to 448x256x256. Increasing the domain size effects automatically to ratio between inner and outer tasks. This influences the possibility of overlapping communication and computation. As you can see in the attached plot we can get for bigger domain sizes linear speedup. At smaller domain size the communication takes longer than the calculation and cannot be hidden completely.



Significant Results

First working multi-node hybrid MPI/OmpSs version. At the base of this implementation we did one tuning steps to improve the parallel performance of the application and compared the results with a version without overlapping communication and computation.

Future Plans

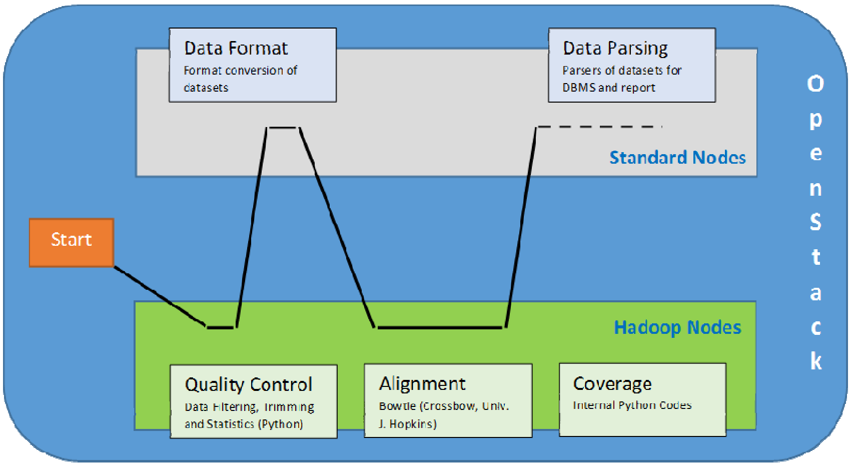
Fixing issue with NUMA effects shown and discussed before. Also clarify if this is only an effect on the Cray XC40(former XC30) (at HLRS). Start to port the application to the target platform and tune, benchmark it there. Here we can benefit from the tuning and benchmarking work we did on the XC40.

9. HTC for Bioinformatics

Partner: CINECA
ARM Cluster: N/A
Kernel:
Domain: New Mont-Blanc2 HTC Application

Performed Activities

At CINECA we designed the initial setup of a bioinformatics HTC application able to run on some of the prototypes made available to the WP3 during the project timeframe.
As reported in the previous QR, we decided to first implement the model of the application(s) workflow on a virtualized environment (OpenStack) using Hadoop/MapReduce (H/MR) and after its assessment to select the most suitable prototype(s) on which test the HT performance. During the present period of activity we selected as a bioinformatics "application" a full NGS pipeline to analyse a *Whole Exome* lab experiment and to alternatively rely on a simpler *ChipSec* pipeline on the most critical cases to implement on the physical hardware of the MB2 prototypes.The model we finally selected to implement following primarily well-known best-practice in this area and then taking into account internal experience at CINECA, is outlined in figure below:


Data Format
Format conversion of datasets
Data Parsing
Parsers of datasets for DBMS and report
Standard Nodes
Start
Hadoop Nodes
Quality Control
Data Filtering, Trimming and Statistics (Python)
Alignment
Bowtie (Crossbow, Univ. J. Hopkins)
Coverage
Internal Python Codes
OpenStack

The Whole Exome (as well as ChipSec) NGS workflow we started from is a well-defined and standardized (ISO9001) pipeline CINECA has deployed to its end-users through the PLX system during the last three years or so. After a preliminary installation on the NUBES virtualized system we have now defined the VM architectures' configuration parameters for the new big-data machine (PICO). Four nodes will be dedicated to this activity, each with 20 physical cores, 128 GB RAM, Fibre Channel storage and Infiniband FDR communication subsystem.

Significant Results

The model of the H/MR bioinformatics application has been selected (Whole Exome pipeline) and machine hardware and software for the virtualized computing environment assigned.

Future Plans

In the next time period of four months we are going to implement the designed model on the new big-data machine (PICO) and start its initial test.

10. Cannon benchmark

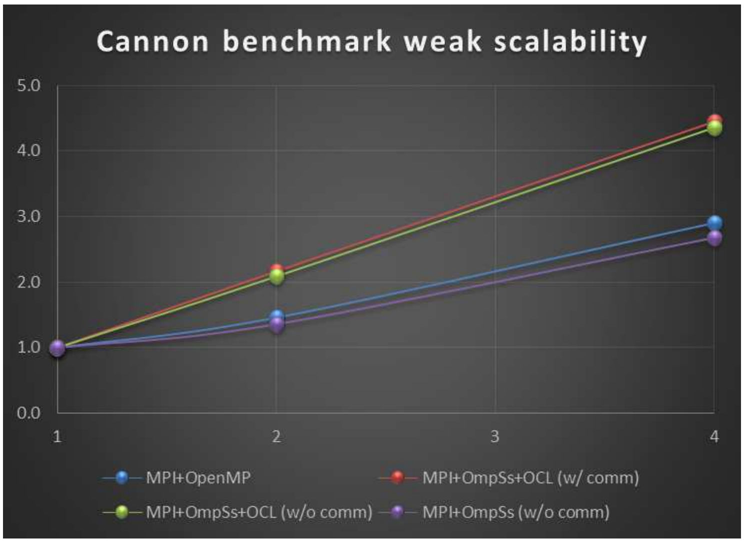
Partner: BSC/CINECA
ARM Cluster: full Mont-Blanc prototype
Kernel: N/A
Domain: Scalability tests

Performed Activities
Based on the first preliminary benchmark analysis of the Cannon algorithm reported on P1-Q3 QR, a more detailed analysis has been carried out on up to 64 SDBs.
The results obtained (in GFLOPS) are briefly outlined in the following table:

N. of nodes/MPI procs	**MPI** + OpenMP (2 threads x node)	**MPI+OmpSs+OpenCL** (w/ comm task)	**MPI+OmpSs+OpenCL** (w/o comm task)	**MPI+OmpSs** (w/o comm task)
4/4	0.51	4.78	**5.35**	**0.36**
16/16	1.49	20.69	**22.34**	**0.98**
64/64	5.93	84.98	**93.17**	**3.86**

We used for tests the *MPI+OpenMP/OmpSs/OpenCL* Cannon benchmark developed in this project. Data in table refer to weak scaling benchmark with a matrix size containing 1536/3072/6144 double precision elements at increasing number of MPI processors and/or number of nodes. In this benchmark we used the Nanos **NX_THREADS** variable equal to **1** or **2** and we reported the best value obtained. Furthermore, we compared the performance of the MPI+OmpSs+OpenCL version where the communication stream was an OmpSs SMP task or not.

Significant Results
Weak scaling behaviour of the algorithm is very well reproduced by the present version of the MB prototype when using MPI+OmpSs+OpenCL configuration even if the full node performance are still far from be reached (ca. 4.5% of 6.8+25.5 GFLOPS given by the 2x ARM Cortex-A15 + Mali GPU of a single SDB). The positive impact of OpenCL kernels on performances it is also highlighted by these preliminary results.


Cannon benchmark weak scalability
5.0
4.0
3.0
2.0
1.0
0.0
1
2
3
4
MPI+OpenMP
MPI+OmpSs+OCL (w/ comm)
MPI+OmpSs+OCL (w/o comm)
MPI+OmpSs (w/o comm)

11. Conclusions and next steps

For the next half-period of the project we will continue the test COSMO and QuantumESPRESSO on the Mont-Blanc2 miniclusters that will be make available to the project partners (e.g cluster of Odroid XU3 Octa from BSC recently installed).

The new Mont-Blanc2 applications (BUDE, ROTORSIM and LBC) will also be tested in terms of T3.1 and T3.2 activities on those new prototypes and we also plan to have an updated version of the Whole Exome HPC pipeline with the first timings of a partial workflow running on the PICO machine at CINECA.

Starting from M25 a comprehensive benchmarking activity on the selected applications/kernels plus the HTC workflow will be carried out on the available mini-clusters and T3.1-T3.3 results will be outlined on *D3.3 Applications porting and tuning reports* and *D3.4 Final benchmark report* documents.

12. Acronyms and Abbreviations

ARM	Advanced RISC (Reduced Instruction Set Computer) Machine
ATLAS	Automatically Tuned Linear Algebra Software
BLAS	Basic Linear Algebra Subprograms
clBLAS	OpenCL BLAS library from AMD
clFFT	OpenCL FFT library from AMD
CPU	Central Processing Unit
CUBLAS	CUDA BLAS library
CUDA	Compute Unified Device Architecture
cuFFT	CUDA FFT library
DARPA	Defense Advanced Research Project Agency
DEISA	Distributed European Infrastructure for Supercomputing Application
FFT	Fast Fourier Transform
GEMM	General Matrix-Matrix Multiply
GNU	Gnu’s Not Unix
GPU	Graphic Processing Unit
HDF	Hierarchical Data Format
HPC	High Performance Computing
MPI	Message Passing Interface
NetCDF	Network Common Data Format
OmpSs	BSCs’ OpenMP SuperScalar programming model
OpenCL	Open Computing Language
PRACE	Partnership for Advanced Computing in Europe
QCD	Quantum Chromo-Dynamics
SDB	System Daughter Board
SDK	Software Development Kit
SoC	System on Chip
ZGEMM	Double precision complex General Matrix-Matrix Multiply