
# E-Infrastructures
# H2020-EINFRA-2014-2015

## EINFRA-4-2014: Pan-European High Performance Computing Infrastructure and Services

## PRACE-4IP

## PRACE Fourth Implementation Phase Project

**Grant Agreement Number: EINFRA-653838**

# D7.7
# Performance and energy metrics on PCP systems

***Final***

Version: 1.2
Author(s): Victor Cameo Ponz, CINES
Date: 08.01.2018

## Project and Deliverable Information Sheet

| | | |
|---|---|---|
| **PRACE Project** | **Project Ref. №: EINFRA-653838** | |
| | **Project Title: PRACE Fourth Implementation Phase Project** | |
| | **Project Web Site:** http://www.prace-project.eu | |
| | **Deliverable ID:** < **D7.7**> | |
| | **Deliverable Nature:** <Report> | |
| | **Dissemination Level:** PU* | **Contractual Date of Delivery:** 31 / 12 / 2017 |
| | | **Actual Date of Delivery:** 15 / 01 / 2018 |
| | **EC Project Officer: Leonardo Flores Añover** | |

* - The dissemination level are indicated as follows: **PU** – Public, **CO** – Confidential, only for members of the consortium (including the Commission Services) **CL** – Classified, as referred to in Commission Decision 2991/844/EC.

## Document Control Sheet

| | | |
|---|---|---|
| **Document** | **Title: Performance and energy metrics on PCP systems** | |
| | **ID: D7.7** | |
| | **Version:** <1.2> | **Status:** ***Final*** |
| | **Available at:** http://www.prace-project.eu | |
| | **Software Tool:** Microsoft Word 2010 | |
| | **File(s):** D7.7_v1.2.docx | |
| **Authorship** | **Written by:** | Victor Cameo Ponz, CINES |
| | **Contributors:** | Andrew Emerson (CINECA)<br>Arno Proeme (EPCC)<br>Charles Moulinec (STFC)<br>Dimitris Dellis (GRNET)<br>Emmanuel Agullo (INRIA)<br>Gilles Marait (INRIA)<br>Hayk Shoukourian (LRZ)<br>Jacob Finkenrath (CyI)<br>Luc Giraud (INRIA)<br>Luigi Iapichino (LRZ)<br>Mariusz Uchronski (WCNS/PSNC)<br>Martti Louhivuori (CSC)<br>Raul De La Cruz (BSC)<br>Ricard Borrell (BSC)<br>Stéphane Lanteri (INRIA)<br>Volker Weinberg (LRZ)<br>Valeriu Codreanu (SurfSARA) |
| | **Reviewed by:** | Janez Povh, ULFME<br>Thomas Eickermann, FZJ<br>Alan Simpson, EPCC |
| | **Approved by:** | MB/TB |

## Document Status Sheet

| Version | Date | Status | Comments |
|---|---|---|---|
| 0.1 | 21/12/2017 | Draft | First version |
| 0.2 | 04/01/2018 | Draft | Correct typos |
| 1.0 | 08/01/2018 | Complete version | Improved section 3 and conclusion + final typos |
| 1.1 | 18/01/2018 | Final version | Corrected typo in the wrap-up table |
| 1.2 | 24/04/18 | Final version | Re-wording conclusion post-review |

## Document Keywords

| | |
|---|---|
| **Keywords:** | PRACE, HPC, Research Infrastructure, Intel Xeon Phi, GPU, KNL, Nvidia P100, energy, performance, benchmark, UEABS |

# Table of Contents

## List of Figures



## List of Tables

## References and Applicable Documents

[1] Miletone 33: Work plan definition: https://misterfruits.gitlab.io/ueabs/ms33.html

[2] Stephen Booth et al., Technical lessons learnt from the implementation of the joint PCP for PRACE-3IP, PRACE 3IP deliverable D8.3.4, 2018

[3] Bull website: https://bull.com/

[4] CINES: https://www.cines.fr/

[5] DAVIDE dedicated webpage https://www.e4company.com/en/?id=press&section=1&page=&new=davide_supercomputer

[6] E4 computer engineering: https://www.e4company.com

[7] CINECA: http://hpc.cineca.it/

[8] S.Bernardi, P. Segers et al., Exploitation Plan of the Results Obtained via PCP, deliverable D2.1.5, 2017

[9] V. Cameo Ponz et al., Application performance on Accelerators, PRACE 4IP deliverable D7.5, 2016

[10] The Unified European Application Benchmark Suite – http://www.prace-ri.eu/ueabs/

[11] Mark Bull et al., Unified European Applications Benchmark Suite, PRACE 4IP deliverable D7.4, 2013

[12] Specfem3D_Globe Github repository: https://github.com/geodynamics/specfem3d_globe

[13] A. M. Beck, G. Murante, A. Arth, R.-S. Remus, A. F. Teklu, J. M. F. Donnert, S. Planelles, M. C. Beck, P. Förster, M. Imgrund, K. Dolag, and S. Borgani, “An improved SPH scheme for cosmological simulations,” MNRAS, vol. 455, pp. 2110–2130, 2016.

[14] F. Baruffa, L. Iapichino, V. Karakasis, N.J. Hammer, "Performance optimisation of Smoothed Particle Hydrodynamics algorithms for multi/many-core architectures", proceedings of the 2017 International Conference on High Performance Computing & Simulation (HPCS 2017), 381. DOI: 10.1109/HPCS.2017.64, 2017.

[15] V. Springel, “The cosmological simulation code GADGET-2,” MNRAS, vol. 364, pp. 1105–1134, 2005.

[16] HORSE: http://www-sop.inria.fr/nachos/index.php/Software/HORSE

[17] MaPHyS: https://gitlab.inria.fr/solverstack/maphys

[18] L. Li, S. Lanteri and R. Perrussel. A hybridizable discontinuous Galerkin method combined to a Schwarz algorithm for the solution of the 3D time-harmonic Maxwell’s equations. J. Comp. Phys., Vol. 256, pp. 563-581 (2014)

[19] P.R. Amestoy, I.S. Duff and J.-Y. L'Excellent, Multifrontal parallel distributed symmetric and unsymmetric solvers. Comput. Methods in Appl. Mech. Eng., Vol. 184, pp. 501-520 (2000)

[20] P. Hénon, P. Ramet and J. Roman. PaStiX: A high-performance parallel direct solver for sparse symmetric definite systems. Paral. Comput., Vol. 28, No. 2, pp. 301-321 (2002)

[21] L. Giraud, A. Haidar and L.T. Watson. Parallel scalability study of hybrid preconditioners in three dimensions. Paral. Comput., Vol. 34, pp. 363-379 (2008)

[22] E. Agullo, L. Giraud, A. Guermouche and J. Roman. Parallel hierarchical hybrid linear solvers for emerging computing platforms. Compte Rendu de l'Académie des Sciences – Mécanique, Vol. 339, No. 2-3, pp. 96-105 (2011)

[23] E. Agullo, L. Giraud and L. Poirel, Robust coarse spaces for Abstract Schwarz preconditioners via generalized eigenproblems, Inria Research Report RR-8978 (2016)

## List of Acronyms and Abbreviations

| | |
|---|---|
| aisbl | Association International Sans But Lucratif (legal form of the PRACE-RI) |

| | |
|---|---|
| BCO | Benchmark Code Owner |
| CoE | Center of Excellence |
| CPU | Central Processing Unit |
| CUDA | Compute Unified Device Architecture (NVIDIA) |
| DARPA | Defense Advanced Research Projects Agency |
| DEISA | Distributed European Infrastructure for Supercomputing Applications EU project by leading national HPC centres |
| DoA | Description of Action (formerly known as DoW) |
| EC | European Commission |
| EESI | European Exascale Software Initiative |
| EoI | Expression of Interest |
| ESFRI | European Strategy Forum on Research Infrastructures |
| FPGA | Field-programmable gate array |
| GB | Giga (= $2^{30}$ ~ $10^9$) Bytes (= 8 bits), also GByte |
| Gb/s | Giga (= $10^9$) bits per second, also Gbit/s |
| GB/s | Giga (= $10^9$) Bytes (= 8 bits) per second, also GByte/s |
| GÉANT | Collaboration between National Research and Education Networks to build a multi-gigabit pan-European network. The current EC-funded project as of 2015 is GN4. |
| GFlop/s | Giga (= $10^9$) Floating point operations (usually in 64-bit, i.e. DP) per second, also GF/s |
| GHz | Giga (= $10^9$) Hertz, frequency =$10^9$ periods or clock cycles per second |
| GPU | Graphic Processing Unit |
| HDEEM | High Definition Energy Efficiency Monitoring |
| HDG | Hybridization of DG methods |
| HET | High Performance Computing in Europe Taskforce. Taskforce by representatives from European HPC community to shape the European HPC Research Infrastructure. Produced the scientific case and valuable groundwork for the PRACE project. |
| HMM | Hidden Markov Model |
| HPC | High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing |
| HPL | High Performance LINPACK |
| ISC | International Supercomputing Conference; European equivalent to the US based SCxx conference. Held annually in Germany. |
| KB | Kilo (= $2^{10}$ ~$10^3$) Bytes (= 8 bits), also KByte |
| LINPACK | Software library for Linear Algebra |
| MB | Management Board (highest decision making body of the project) |
| MB | Mega (= $2^{20}$ ~ $10^6$) Bytes (= 8 bits), also MByte |
| MB/s | Mega (= $10^6$) Bytes (= 8 bits) per second, also MByte/s |
| MFlop/s | Mega (= $10^6$) Floating point operations (usually in 64-bit, i.e. DP) per second, also MF/s |
| MooC | Massively open online Course |
| MoU | Memorandum of Understanding. |
| MPI | Message Passing Interface |
| NDA | Non-Disclosure Agreement. Typically signed between vendors and customers working together on products prior to their general availability or announcement. |
| PA | Preparatory Access (to PRACE resources) |
| PATC | PRACE Advanced Training Centres |
| PRACE | Partnership for Advanced Computing in Europe; Project Acronym |

| | |
|---|---|
| PRACE 2 | The upcoming next phase of the PRACE Research Infrastructure following the initial five year period. |
| PRIDE | Project Information and Dissemination Event |
| RI | Research Infrastructure |
| TB | Technical Board (group of Work Package leaders) |
| TB | Tera (= $2^{40}$ ~ $10^{12}$) Bytes (= 8 bits), also TByte |
| TCO | Total Cost of Ownership. Includes recurring costs (e.g. personnel, power, cooling, maintenance) in addition to the purchase cost. |
| TDP | Thermal Design Power |
| TFlop/s | Tera (= $10^{12}$) Floating-point operations (usually in 64-bit, i.e. DP) per second, also TF/s |
| Tier-0 | Denotes the apex of a conceptual pyramid of HPC systems. In this context the Supercomputing Research Infrastructure would host the Tier-0 systems; national or topical HPC centres would constitute Tier-1 |
| UNICORE | Uniform Interface to Computing Resources. Grid software for seamless access to distributed resources. |

## List of Project Partner Acronyms

| | |
|---|---|
| BADW-LRZ | Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften, Germany (3rd Party to GCS) |
| BILKENT | Bilkent University, Turkey (3rd Party to UYBHM) |
| BSC | Barcelona Supercomputing Center - Centro Nacional de Supercomputacion, Spain |
| CaSToRC | Computation-based Science and Technology Research Center, Cyprus |
| CCSAS | Computing Centre of the Slovak Academy of Sciences, Slovakia |
| CEA | Commissariat à l'Energie Atomique et aux Energies Alternatives, France (3 rd Party to GENCI) |
| CESGA | Fundacion Publica Gallega Centro Tecnológico de Supercomputación de Galicia, Spain, (3rd Party to BSC) |
| CINECA | CINECA Consorzio Interuniversitario, Italy |
| CINES | Centre Informatique National de l'Enseignement Supérieur, France (3 rd Party to GENCI) |
| CNRS | Centre National de la Recherche Scientifique, France (3 rd Party to GENCI) |
| CSC | CSC Scientific Computing Ltd., Finland |
| CSIC | Spanish Council for Scientific Research (3rd Party to BSC) |
| CYFRONET | Academic Computing Centre CYFRONET AGH, Poland (3rd party to PNSC) |
| EPCC | EPCC at The University of Edinburgh, UK |
| ETHZurich (CSCS) | Eidgenössische Technische Hochschule Zürich – CSCS, Switzerland |
| FIS | FACULTY OF INFORMATION STUDIES, Slovenia (3rd Party to ULFME) |
| GCS | Gauss Centre for Supercomputing e.V. |
| GENCI | Grand Equipement National de Calcul Intensiv, France |
| GRNET | Greek Research and Technology Network, Greece |
| INRIA | Institut National de Recherche en Informatique et Automatique, France (3 rd Party to GENCI) |
| IST | Instituto Superior Técnico, Portugal (3rd Party to UC-LCA) |
| IUCC | INTER UNIVERSITY COMPUTATION CENTRE, Israel |
| JKU | Institut fuer Graphische und Parallele Datenverarbeitung der Johannes Kepler Universitaet Linz, Austria |
| JUELICH | Forschungszentrum Juelich GmbH, Germany |
| KTH | Royal Institute of Technology, Sweden (3 rd Party to SNIC) |
| LiU | Linkoping University, Sweden (3 rd Party to SNIC) |
| NCSA | NATIONAL CENTRE FOR SUPERCOMPUTING APPLICATIONS, Bulgaria |
| NIIF | National Information Infrastructure Development Institute, Hungary |
| NTNU | The Norwegian University of Science and Technology, Norway (3rd Party to SIGMA) |
| NUI-Galway | National University of Ireland Galway, Ireland |
| PRACE | Partnership for Advanced Computing in Europe aisbl, Belgium |
| PSNC | Poznan Supercomputing and Networking Center, Poland |
| RCS | Radar Cross Section |
| RISCSW | RISC Software GmbH |

| | |
|---|---|
| RZG | Max Planck Gesellschaft zur Förderung der Wissenschaften e.V., Germany (3rd Party to GCS) |
| SIGMA2 | UNINETT Sigma2 AS, Norway |
| SMT | Simultaneous Multi-Threading, name given by IBM for the hyper-threading feature of Power processors |
| SNIC | Swedish National Infrastructure for Computing (within the Swedish Science Council), Sweden |
| SoC | System on a Chip |
| STFC | Science and Technology Facilities Council, UK (3rd Party to EPSRC) |
| SURFsara | Dutch national high-performance computing and e-Science support center, part of the SURF cooperative, Netherlands |
| UC-LCA | Universidade de Coimbra, Labotatório de Computação Avançada, Portugal |
| UCPH | Københavns Universitet, Denmark |
| UHEM | Istanbul Technical University, Ayazaga Campus, Turkey |
| UiO | University of Oslo, Norway (3rd Party to SIGMA) |
| ULFME | UNIVERZA V LJUBLJANI, Slovenia |
| UmU | Umea University, Sweden (3rd Party to SNIC) |
| UnivEvora | Universidade de Évora, Portugal (3rd Party to UC-LCA) |
| UPC | Universitat Politècnica de Catalunya, Spain (3rd Party to BSC) |
| UPM/CeSViMa | Madrid Supercomputing and Visualization Center, Spain (3rd Party to BSC) |
| USTUTT-HLRS | Universitaet Stuttgart – HLRS, Germany (3rd Party to GCS) |
| VSB-TUO | VYSOKA SKOLA BANSKA - TECHNICKA UNIVERZITA OSTRAVA, Czech Republic |
| WCNS | Politechnika Wroclawska, Poland (3rd party to PNSC) |

# Executive Summary

This document describes efforts undertaken in order to exploit PRACE Pre-Commercial Procurement (PCP) machines. It aims at giving an overview of what can be done in terms of performances and energy analysis on these prototypes. The key focus has been given to a general study using the PRACE Unified European Application Benchmark Suite (UEABS) and to a more detailed case study porting a solver stack using cutting edge tools.

This work has been undertaken by the PRACE Fourth Implementation Phase (PRACE-4IP) extension task "Performance and energy metrics on PCP systems" which is a follow-up of the Task 7.2B "Accelerators benchmarks" in the PRACE-4IP.

It also heads in the direction of the Task 7.3 in PRACE-5IP meaning to merge PRACE accelerated and standard benchmark suites, as codes of the latter have been run on accelerators in this task.

As a result, ALYA, Code_Saturne, CP2K, GPAW, GROMACS, NAMD, PFARM, QCD, Quantum Espresso, SHOC and Specfem3D_Globe (already ported to accelerator) and GADGET and NEMO (newly ported) have been selected to run on Intel KNL and NVDIA GPU to give an overview of performances and energy measurement.

Also, the HORSE+MaPHyS+PaStiX solver stack has been selected to be ported to Intel KNL. Focus here has been given to performing an energetic profiling of theses codes and studying the influence of several parameters driving the accuracy and numerical efficiency of the underlying simulations.

## 1 Introduction

The work conducted within this task is driven by the delivery of PCP machines. It is a separate project from PRACE-3IP PCP and stands as an extension of the completed WP7 PRACE-4IP project. It aims at giving manufacturer-independent performance and energy metrics for future exascale systems. It is also an opportunity to explore and test the cutting edge energy hardware stack and tools developed within the scope of PCP.

As stated in the Milestone 33 of PRACE-4IP - Workplan definition (MS33)[1], this document will present metrics for selected codes among the UEABS. It shows results concerning many scientific fields used among European scientific communities. It will also go deeper in the porting and energetic profiling activities using the HORSE+MaPHyS+PaStiX solver stack as an example.

Section 2 details hardware and software specifications of Frioul-PCP and DAVIDE where metrics have been carried out. In Section 3 the metrics for UEABS are brought together. The work on porting and energy profiling is presented in Section 4. Section 5 concludes and outlines further work on PCP prototypes and energy related work.

## 2 Clusters specifications and access

The PRACE PCP project includes three different prototypes, Frioul-PCP, DAVIDE and Jumax using respectively Xeon Phi, GPU and FPGA. First two machines become more and more common in HPC infrastructures, making the energy stack being the innovation. On the opposite, the last architecture is brand new in this field, leading to a substantially higher effort to get familiar with it.

As demonstrated in section 2.1 tight deadlines did not let the time to produce relevant metrics on the FPGA cluster. This risk has been risen from the beginning. Therefore, only GPU, DAVIDE and KNL, Frioul-PCP prototypes are presented here.

## 2.1 Access to machines

Working with prototypes can be painful in terms of project management and meeting deadlines. This section is dedicated to give a feedback on accessing the hardware and software stack.

Figure 1 outlines the initial tight deadlines for this project. Also, it shows that access to machines for running codes has been possible quite late.

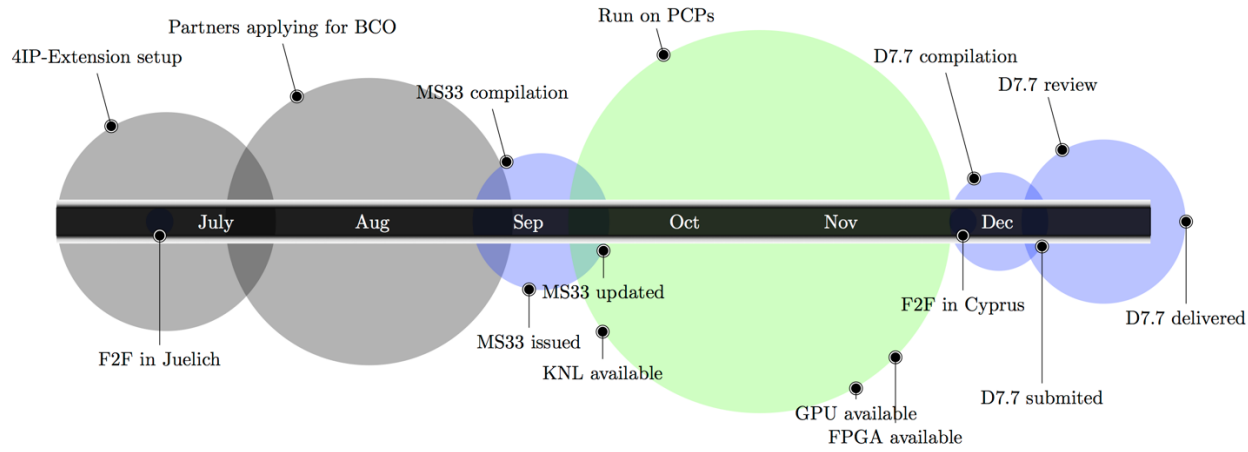


**Figure 1: PRACE-4IP-extension project timeline. On top of the figure are printed periods names and on the bottom key milestones. Periods in grey stand for task preparation, periods in blue stand for documentation redaction and period in green stand for technical work.**

Table 1 shows the precise timeline. Some technical interruptions occurred right at the end of the running phase:

*Frioul-PCP:*
- closed from 22th November to December the 4th
- login node has been down form the 5th to the 7th of December.
- energy metrics tools down from 5th to the 12th of December

*DAVIDE:*
- slurm not working from 6th to the 11th of December
- energy metrics tools *randomly* not working during beginning of December

Therefore, the planned work could not be fully completed within the timeframe of PRACE-4IP extention. However, the prototypes will be made available to PRACE partners for further testing under the new “Preparatory Access Type P”, created for this purpose (see PRACE-3IP Deliverable D2.1.5 Exploitation Plan of the Results Obtained via PCP[8]).

**Table 1: PCP Systems access dates**

| | Frioul-PCP | DAVIDE | Jumax |
|---|---|---|---|
| Envisioned | Jun-17 | Jul-17 | Aug-17 |
| Actual access | 01-Sep-17 | 16-Oct-17 | 02-Nov-17 |
| Access to energy stack | 06-Oct-17 | 08-Nov-17 | / |

## 2.2 Frioul-PCP

This machine[2] has been designed by Atos/Bull[1] and is hosted at CINES[4] in Montpellier, France. It is made of 56 Bull Sequana X1210 blades, each including 3 Xeon Phi KNL nodes. It total, it has a theoretical peak performance of 465 Tflop/s with an estimated consumption of 82kW1.

### 2.2.1 *Compute technology*

Hardware features the following nodes:

- 168 nodes with
  - 1x Intel Xeon Phi 7250 processor (KNL), 68 cores cadenced to 1.4 GHz with SMT 4.
  - 96GB memory, 16GBx6 DDR4 DIMMs
- intranode communications integrated using InfiniBand EDR
- 100% Hot water cooled nodes
- Half of the configuration feature liquid cooled Power Supply Unit (PSU) make this part of the machine 100% liquid cooled.
- MooseFS I/O

### 2.2.2 *Energy sampling technology*

Power measurements at node level occurs at the sampling rate of 1 kHz at converters and 100 Hz at CPU/DRAM. It is provided through a HDEEM FPGA on each node.

Atos/Bull[1] allow energy access through two frameworks, namely HDEEM VIZualization (HDEEVIZ) and Bull Energy Optimizer (BEO).

**HDEEVIZ:**
Components:

- SLURM synchronisation + initialisation
- HDEEM writing results to local storage
- Grafana: Graphical HTML user interface

Here's an example of usage in a submission script:

```
#SBATCH -N 2
#SBATCH -time 00:30:00
#SBATCH -J Specfem3D_Globe
#SBATCH -n 89

module load intel/17.2 intelmpi/2018.0.061
module load hdeeviz/hdeeviz_intelmpi_2018.0.061

hdeeviz mpirun -n 89 $PWD/bin/xspecfem3D
```

Access to generated data is provided through the Grafana web interface as shown in Figure 2.

[1] 1080W measured at blade power supply


**Figure 2: Example of Grafana HTML output**

### BEO

BEO is a set of system administrator oriented tools that allows to get energy metrics at switch and node level. At user level the main interesting feature is `get_job_energy slurm<job_id<optionnal:.jobstep>>`. It produces the following output:

```
$ beo report energy slurm8170
| job  | nodes.energy | switches.energy | job.energy | job.cost |
=====================================================================
| 8170 |     618.4 kJ |         56.3 kJ |   674.7 kJ | 0.0219 € |
```

## 2.3 DAVIDE

DAVIDE[2] has been designed by E4 computer engineering[6] and is hosted at CINECA[7] in Bologna, Italy. It has a total theoretical peak performance of 990 TFlop/s (double precision). A more detailed description can be found on the E4 dedicated webpage[5].

### 2.3.1 *Compute technology*

Hardware features fat-nodes with the following design:

- 45 nodes with
  - 2 IBM POWER8+ processors, i.e. 8x2 cores with Simultaneous Multi-Threading (SMT) 8
  - 4 NVIDIA P100 GPU with 16GB High Bandwidth Memory 2 (HBM2)
- intranode communications integrated using NVLink
- internode communications integrated using Infiniband EDR interconnect in a fat-tree with no oversubscription topology
- CPU and GPU direct hot water (~27°C) cooling, removing 75-80% of the total heat
- the remaining 20-25% heat is air-cooled

Each compute node has a theoretical peak performance of 22 Tflop/s (double precision) and a power consumption of less than 2kW[2].

### 2.3.2 *Energy sampling technology*

Information is collected from processors, memory, GPUs and fans exploiting Analog-to-Digital Converters in the embedded SoC. It provides sampling up to 800 kHz lowered to 50kHz on power measuring sensor outputs.

The technology has been developed in collaboration with the University of Bologna, which developed the `get_job_energy <job_id>` program. Usage is straightforward and has a very verbose output (truncated here for lisibility):

```
$ get_job_energy 12389
Job 12389
         - Duration (seconds): 421.0
         - Used Node(s): davide20
         - Requested CPUs: 16
         […]
<============================================================>
        Total nodes power consumption "at the plug". Integral of the
          power consumed by each node sampled at 800KHz. BBB Measures
        Cumulative (all nodes)
         - Mean power (W): 536.402900943
         - Total energy (J): 225825.621297
<------------------------------------------------------------>
        Node Average
         - Mean node power (W): 536.402900943
         - Total node energy (J): 225825.621297
<============================================================>
        AMESTER Power Measures of main components. Integral of the
          power consumed by each component sampled at 4KHz :
        Cumulative (all nodes)
         - Mean power (W): 513.785714286
         - Total energy (J): 216303.785714
         - Mean FANs power (W): 27.0
         - Total FANs energy (J): 11367.0
         - Mean GPUs power (W): 107.047619048
         - Total GPUs energy (J): 45067.0476192
         […]
<------------------------------------------------------------>
        Node Average
         - Mean node power (W): 513.785714286
         - Total node energy (J): 216303.785714
         - Mean FAN power (W): 27.0
         - Total FAN energy (J): 11367.0
         - Mean GPU power (W): 107.047619048
         - Total GPU energy (J): 45067.0476192
         […]
```

# 3 Performances and energy metrics of UEABS on PCP systems

This section presents results of UEABS on both DAVIDE and Frioul-PCP systems. There is two version of this suite. The first is used to be run on standard CPU and the latter has been ported to accelerators. The accelerated suite is described in D7.5[9] and the standard suite is

[2] Including Power8+ and 4 Pascal GPU consumption only

described on the PRACE UEABS official webpage[10] and D7.4[11]. These documents also describe test cases specific to this suite and where to find corresponding datasets.

Metrics exhibited systematically are time to solution and energy to solution. This choice allows measuring the exact same computation for both figures. Indeed, some codes feature specific performance metrics, e.g. not considering warm up and teardown phases. This metrics are thus not biased and small benchmark test cases can then give more information about hypothetic production runs. Unfortunately, such a system is not available yet for energy, therefore performances metrics will be shown as *side metrics*.

To be comparable between machines, the `Cumulative (all nodes) Total energy (J)` has been selected for DAVIDE. And the `nodes.energy` has been selected for Frioul-PCP prototype. Both measure full nodes consumption in Joule.

Each code will be presented along with a short description and the full set of metrics. The set of metrics is obtained by benchmark owners after multiple calibration runs until an optimal compilation setup have been found. The section ends with a recap chart with a line of metric picked up for its relevance.

## 3.1 ALYA

Alya[9][10][11] is a high performance computational mechanics code that can solve different coupled mechanics problems.

Some specific developments have been carried out in Alya to take advantage of the PCP systems. Vectorisation has been extended to the explicit part of the code. Vectorisation strategy allows to adapt data structures depending on the desired vector size. The same code is used for the KNL and GPU versions, only the vector size changes at runtime, on DAVIDE it is $10^4$ while for Frioul-PCP it is 16. Finally, on the KNL OpenMP is used for the multi-threaded execution while almost equivalent OpenACC pragmas are used to offload work to the GPUs.

### 3.1.1 *Test case 1 metrics*

**Table 2: Alya test case 1 metrics on Frioul-PCP**

| Number of full PCP-KNL nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 88,96 | 22,99 |
| 2 | 53,63 | 15,55 |
| 4 | 26,50 | 50,64 |
| 8 | 13,38 | 47,85 |

**Table 3: Alya test case 1 metrics on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 49,22 | 105,14 |
| 2 | 27,26 | 199,28 |
| 4 | 14,82 | 196,56 |
| 8 | 8,35 | 278,62 |

### 3.1.2 *Test case 2 metrics*

**Table 4: Alya test case 2 metrics on Frioul-PCP**

| Number of full PCP-KNL nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 4 | 220,07 | 332,13 |
| 8 | 108,18 | 389,95 |
| 16 | 55,81 | 380,00 |
| 32 | 28,68 | 470,00 |

**Table 5: Alya test case 2 metrics on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 4 | 92,80 | 898,22 |
| 8 | 47,83 | 1 441,30 |
| 16 | 26,65 | 1 722,93 |
| 32 | 14,77 | 1 821,98 |

Two tests cases for tetrahedral meshes have been considered. Mesh of case 1 is composed of 8,6M elements and mesh of case 2 of 68,8 M elements (8 times larger). As expected, the strong scalability is better for the larger case in both PCP systems. This is because its initial load per node doubles the one of the small case and, therefore, the ratio between communications and computations becomes more favourable. The parallel efficiency obtained is good in both systems, but significantly better for the Frioul-PCP KNL (10% for the largest tests). This can be explained by the opposite effect of the workload reduction, inherent of the strong speedup tests. For DAVIDE, which is a throughput oriented device, the decrease of the occupancy reduces performances because it becomes more difficult two hide latencies. Frioul-PCP, which is latency oriented device, decrease of the local problem results in a better exploitation of the cache memory and benefits the performance.

Comparing executions with 1, 2 and 4 nodes for the small case respectively with 8, 16 and 32 nodes for the big one gives three weak speedup tests. The average slowdown is 1.1 on the Frioul-PCP and 1.0 on the DAVIDE system. Both cases are excellent, but DIVIDE is slightly better in this situation, not harmed by the occupancy reduction.

Finally, Frioul-PCP is two times slower than the DAVIDE in a node to node comparison of absolute times. However note that the last ones are composed of 4 GPUs and 1 Power8+ CPU. Roughly speaking, we could say that currently for Alya the execution in an Intel Xeon Phi 7250 is as fast as two NVIDIA P100 GPUs.

For the strong speedup test, an ideal acceleration and a linear increase of energy cost would result in a constant energy cost per job. Both conditions are not true in practice on the PCP systems. Energy consumption grows between 1.4 and 2.6 times, this increase being more notorious for the Davide system. However, an important dispersion on the energy results can be observed, specially for the small test (case 1). Considering the larger case (case 2), the executions on Frioul-PCP are 3.7 times more energy efficient than the ones on DAVIDE system. However, note that on the energy measurements for DAVIDE are also considered the Power8+ hosts that are not in Alya's calculations, but only to carry out intra-node communications.

## 3.2 Code_Saturne

Code_Saturne[9][10][11] is a multi-purpose CFD software package developed by EDF R&D since 1997 and open-source since 2007. Parallelism is handled by distributing the domain over the processors. Communications between subdomains are handled by MPI. Hybrid

parallelism using MPI/OpenMP has recently been optimised for improved multicore performance. The code has also been linked to PETSc to offer alternatives to the internal solvers to compute the pressure. Note that PETSc developer's version supports CUDA.

### 3.2.1 *Test case 1 metrics*

The lid-driven cavity is computed for a cubic box meshed by 13 million tetrahedral cells. PETSc developer's library Krylov solvers are used to compute the pressure in order to benefit from CUDA support. The last column of Table 6 and Table 7 shows the computing time per time step on PCP-KNL and DAVIDE (1, 2 and 4 GPUs per node). A nearly ideal speed-up is observed for all 3 configurations on DAVIDE, where the energy consumption is reduced, when increasing the number of nodes. While the energy consumption is roughly constant on DAVIDE, it drastically increases on PCP-KNL for 8 and 16 nodes.

**Table 6: Code Saturn test case 1 metrics on Frioul-PCP**

| Number of full PCP-KNL nodes | Time to solution (s) | Energy to solution (kJ) | Time/time-step (s) |
|---|---|---|---|
| 1 | 2 029 | 602,2 | 400,59 |
| 2 | 1 120 | 618,3 | 209,33 |
| 4 | 651 | 778 | 109,36 |
| 8 | 470 | 970,3 | 59,47 |
| 16 | 353 | 1 500,000 | 33,93 |

**Table 7: Code Saturn test case 1 metrics on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Energy to solution (kJ) | Time/time-step (s) |
|---|---|---|---|
| 1 Node, 16 MPI tasks, 1 GPU | 640 | 505,57 | 119,91 |
| 2 Nodes, 32 MPI tasks, 1 GPU | 342 | 533,05 | 58,79 |
| 4 Nodes, 64 MPI tasks, 1 GPU | 206 | 629,25 | 29,79 |
| 1 Node, 16 MPI tasks, 2 GPUs | 530 | 426,01 | 98,05 |
| 2 Nodes, 32 MPI tasks, 2 GPUs | 274 | 435,98 | 47,18 |
| 4 Nodes, 64 MPI tasks, 2 GPUs | 166 | 524,88 | 23,89 |
| 1 Node, 16 MPI tasks, 4 GPUs | 479 | 396,77 | 87,36 |
| 2 Nodes, 32 MPI tasks, 4 GPUs | 118 | 182,03 | 42,65 |
| 4 Nodes, 64 MPI tasks, 4 GPUs | 153 | 490,68 | 21,15 |

### 3.2.2 *Test case 2 metrics*

This test case deals with the classical Taylor-Green vortex test case traditionally used to assess numerical schemes accuracy. The cells are hexahedral and the mesh 2563 large. The native algebraic multigrid solver (which does not support CUDA) is used as a preconditioner and the conjugate gradient algorithm as a solver. Table 8 shows for Frioul-PCP that the energy to solution increases as a function of the number of nodes, and that the compute time per time-step is nearly halved up to 8 nodes (the case might be too small for 16 nodes, as about 15 000 cells only are available per task). However the full time to solution does not scale that well, due to I/O consumption. This effect is magnified because of the low number of time steps used for this case, e.g. 100 vs up to 10 000 for production runs.

**Table 8: Code Saturn test case 2 metrics on Frioul-PCP**

| Number of full PCP-KNL nodes | Time to solution (s) | Energy to solution (kJ) | Time / time-step (s) |
|---|---|---|---|
| 1 | 1 421,71 | 369,40 | 54,11 |

| Number of full PCP-KNL nodes | Time to solution (s) | Energy to solution (kJ) | Time / time-step (s) |
|---|---|---|---|
| 2 | 894,45 | 469,60 | 28,98 |
| 4 | 596,7 | 607,00 | 15,01 |
| 8 | 442,33 | 889,90 | 8,02 |
| 16 | 408,85 | 1,600,00 | 5,06 |

## 3.3 CP2K

CP2K[9][10][11] is a quantum chemistry and solid state physics software package.

Parallelisation is achieved using a combination of OpenMP-based multi-threading and MPI. Offloading for accelerators is implemented through CUDA.

For both test cases on DAVIDE system CP2K was run on Power8 CPU only (no GPU) using the pure MPI build with 16 processes per node and with SMT turned off. Few results with have been added for test case 1 but for particular numbers of nodes the linear algebra consistently broke down for no clear reason. Test case 2 give an unexpected error using GPU. These problems have been followed up to developers.

### 3.3.1 *Test case 1 metrics*

**Table 9: CP2K test case 1 metrics on Frioul-PCP**

| Number of full PCP-KNL nodes | Time to solution (s) | Speedup | Energy to solution (kJ) | Energy scaling |
|---|---|---|---|---|
| 1 | 5 917,00 | 1,00 | 1 417,40 | 1,00 |
| 2 | 3 737,00 | 1,58 | 1 631,30 | 1,15 |
| 4 | 1 922,00 | 3,08 | 1 596,20 | 1,13 |
| 8 | 794,00 | 7,45 | 1 520,20 | 1,07 |
| 16 | 424,00 | 13,96 | 1 603,60 | 1,13 |
| 32 | 231,00 | 25,61 | 1 795,50 | 1,27 |
| 64 | 147,00 | 40,25 | 2 343,40 | 1,65 |

**Table 10: CP2K test case 1 metrics on DAVIDE without GPU**

| Number of full Davide nodes | Time to solution (s) | Speedup | Energy to solution (kJ) | Energy scaling | Energy to solution minus GPU energy (kJ) |
|---|---|---|---|---|---|
| 1 | 4 686,00 | 1,00 | 3 365,00 | 1,00 | 2 825,00 |
| 2 | 2 344,00 | 2,00 | 3 351,00 | 1,00 | 2 833,00 |
| 4 | 1 194,00 | 3,92 | 3 459,00 | 1,03 | 2 926,00 |
| 8 | 612,00 | 7,66 | 3 528,00 | 1,05 | 2 978,00 |
| 16 | 323,00 | 14,51 | 3 745,00 | 1,11 | 3 166,00 |

**Table 11: CP2K test case 1 metrics on DAVIDE with GPU**

| Number of full Davide nodes | Time to solution (s) | Speedup | Energy to solution (kJ) | Energy scaling |
|---|---|---|---|---|
| 1 | 4 657,00 | 1,00 | 3 458,00 | 1,00 |
| 2 | 2 337,00 | 1,99 | 3 484,00 | 1,01 |
| 16 | 320,00 | 14,55 | 3 963,00 | 1,15 |

### 3.3.2 *Test case 2 metrics*

**Table 12: CP2K test case 2 metrics on Frioul-PCP**

| Number of full PCP-KNL nodes | Time to solution (s) | Speedup[3] | Energy to solution (kJ) | Energy scaling |
|---|---|---|---|---|
| 2 | 2963,00 | 2,00 | 1410,20 | 1,00 |
| 4 | 1210,00 | 4,90 | 1396,00 | 0,99 |
| 8 | 729,00 | 8,13 | 1531,00 | 1,09 |
| 16 | 383,00 | 15,47 | 1616,00 | 1,15 |
| 32 | 226,00 | 26,22 | 1857,00 | 1,32 |
| 64 | 139,00 | 42,63 | 2427,00 | 1,72 |

**Table 13: CP2K test case 2 metrics on DAVIDE without GPUs**

| Number of full Davide nodes | Time to solution (s) | Speedup | Energy to solution (kJ) | Energy scaling | Energy to solution minus GPU energy (kJ) |
|---|---|---|---|---|---|
| 1 | 24 573,00 | 1,00 | 18 302,00 | 1,00 | 15 504,00 |
| 2 | 12 502,00 | 1,97 | 18 444,00 | 1,01 | 15 684,00 |
| 4 | 6 380,00 | 3,85 | 19 118,00 | 1,04 | 16 217,00 |
| 8 | 3 295,00 | 7,46 | 19 737,00 | 1,08 | 16 777,00 |
| 16 | 1 695,00 | 14,50 | 20 378,00 | 1,11 | 17 314,00 |

Performance is scaling similarly on Frioul-PCP and the DAVIDE, at least up until the largest common node count examined on both platforms (16 nodes), obtaining similar speedup values on a per-node basis, namely 14 to 15 times speedup on 16 nodes. On Frioul-PCP, on which larger node counts were examined, speedup for both test cases is just over 40 times on 64 nodes, being over 60% parallel efficiency. In terms of absolute performance, times to solution for the first test case (LiH-HFX) is broadly similar on the two platforms on a per-node basis. For the second test case (H2O-DFT-LS), which on DAVIDE could only be run on the CPU, i.e. excluding the GPU, Frioul PCP have 4-5 shorter times to solution on equal numbers of nodes. The difference in absolute performance for this test case would be expected to change if the GPU could have been used too.

Total energy to solution is also scaling similarly on both platforms up until the largest common nodecount examined on both platforms (16 nodes), remaining for 16 nodes within a factor of 1.13 of the energy to solution on a single node. On Frioul-PCP energies to solution come to 1.6 to 1.7 times their single-node value on 64 nodes. For absolute energy to solution for the LiH-HFX test case is 2 to 2.3 times higher on DAVIDE compared to on an equal number of nodes on Frioul-PCP, regardless of whether the test case is run on CPU only on DAVIDE or on CPU+GPU. For the H2O-DFT-LS test case absolute energy to solution is around 13 times higher on DAVIDE compared to on an equal number of nodes on Frioul-PCP, though here the comparison could only be made with CPU-only runs on DAVIDE and higher energy efficiency would be expected if the GPU could be used also.

[3] speedup for testcase 2 on Frioul-PCP is relative to their single-node values, which are estimated as being equal to twice their value on 2 nodes

## 3.4 GADGET

P-Gadget3[13] is a cosmological, fully hybrid MPI + OpenMP parallelised Smoothed Particle Hydrodynamics code based on Gadget-2[15]. This was not ported during the PRACE-4IP project on accelerators, and effort spend within this task have been focused on producing results on Frioul-PCP. This effort should be maintained to in PRACE-5IP so UEABS can run on both architechtures.

This version has further undergone some of the node-level optimisation, described in Baruffa et al. 2017[14]. On the Frioul-PCP cluster, instead of using the UEABS test cases, we defined a more suitable single test problem consisting of a cosmological simulation, including cooling and star formation routines, evolving 256^3 N-Body particles and the same number of gas particles. The needed memory per node does not fit into MCDRAM, therefore the code was run with the KNL nodes set in quadrant / flat mode and with memory allocation on DDR. At the time of writing, a full GPU version of the code is not yet available, but under development, thus no measurements could be obtained on the DAVIDE.

The baseline version of the test has been run on 8 KNL nodes. Per node, 4 MPI tasks are run, each of them with 16 OpenMP threads. Please note that this has to be considered as a first reasonable guess, guided by user experience on KNL, and therefore this configuration was not tuned for optimal performance. Also, with this choice not all KNL cores of a single node are in use (the 7250 KNL model has 68 cores). SMT has not being used because it has been verified that it does not bring any performance benefit. Addressing these performance issues is beyond the scope of the current study.

We highlight the results of the last test of Table 14. This has been run with a configuration of MPI tasks and OpenMP threads which has been proved as optimal in previous tests, with respect to the baseline. Consequently, both time and energy to solution are positively impacted by this simple optimisation step, which seems to be crucial on KNL.

**Table 14: Gadget test case metrics with 4 MPI task per node and 16 OpenMP thread per task**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (MJ) |
|---|---|---|
| 4 | 2 082,97 | 1,7 |
| 8 | 1 332,86 | 2,2 |
| 16 | 9 65,82 | 3,1 |

**Table 15: Gadget test case metrics on 8 Frioul-PCP nodes**

| MPI task/node | OpenMP threads/task | Time to solution (s) | Energy to solution (MJ) |
|---|---|---|---|
| 4 | 16 | 1 332,86 | 2,2 |
| 4 | 32 | 1 514,17 | 2,6 |
| 32 | 4 | 897,90 | 1,7 |

## 3.5 GPAW

GPAW[9][10][11] is a DFT program for ab-initio electronic structure calculations using the projector augmented wave method.

GPAW is written mostly in Python, but includes also computational kernels written in C as well as leveraging external libraries such as NumPy, BLAS and ScaLAPACK. There is support for offloading to accelerators using either CUDA or pyMIC, respectively.

The GPU branch of GPAW happens to be very old and not functional so it has been agreed not to run GPAW on DAVIDE[1]. However as in PRACE 4IP some effort has been spent to make it run on GPU unsuccessfully.

### 3.5.1 *Test case 1 metrics*

**Table 16: GPAW test case 1 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 527 | 139 |
| 2 | 307 | 225 |
| 4 | 187 | 277 |
| 8 | 141 | 442 |
| 16 | 115 | 774 |
| 32 | 118 | 1700 |

### 3.5.2 *Test case 2 metrics*

**Table 17: GPAW test case 2 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 457 | 144 |
| 2 | 215 | 188 |
| 4 | 129 | 231 |
| 8 | 72 | 327 |
| 16 | 50 | 577 |
| 32 | 36 | 1100 |

## 3.6 GROMACS

GROMACS[9][10][11] is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles.

Parallelisation is achieved using combined OpenMP and MPI. Offloading for accelerators is implemented through CUDA for GPU and through OpenMP for MIC (Intel Xeon Phi).

### 3.6.1 *Test case 1 metrics*

**Table 18: GROMACS test case 1 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Optional metric (ns/day) | Energy to solution (kJ) |
|---|---|---|---|
| 1 | 672,316 | 16,06 | 232,8 |
| 2 | 403,7 | 26,74 | 261,2 |
| 4 | 278,13 | 38,83 | 287,1 |

**Table 19: GROMACS test case 1 metrics on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (ns/day) | Energy to solution (kJ) |
|---|---|---|---|
| 1 | 346,91 | 31,13 | 317,71 |

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (ns/day) | Energy to solution (kJ) |
|---|---|---|---|
| 2 | 226,28 | 49,94 | 390,03 |
| 4 | 201,32 | 53,64 | 702,50 |
| 8 | 132,82 | 81,31 | 938,48 |

### 3.6.2 *Test case 2 metrics*

**Table 20: GROMACS test case 2 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Optional metric (ns/day) | Energy to solution (kJ) |
|---|---|---|---|
| 1 | 1 166,93 | 1,48 | 529,7 |
| 4 | 353,33 | 4,89 | 533,9 |
| 8 | 183,34 | 9,42 | 603,5 |
| 16 | 121,89 | 14,17 | 817,4 |
| 32 | 77,33 | 22,34 | 1200 |
| 48 | 59,0 | 29,25 | 1700 |

**Table 21: GROMACS test case 2 metrics on DAVIDE with SMT off (i.e. SMT=1)**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (ns/day) | Energy to solution (kJ) |
|---|---|---|---|
| 1 | 731 | 2,36 | 641,6 |
| 4 | 195,64 | 9,24 | 682,9 |
| 8 | 122,2 | 14,13 | 900,4 |
| 16 | 64,58 | 21,4 | 1264,1 |
| 32 | 44,84 | 38,54 | 1723 |
| 40 | 43,45 | 39,77 | 2186,5 |

**Table 22: GROMACS test case 2 metrics on DAVIDE with SMT=8**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (ns/day) | Energy to solution (kJ) |
|---|---|---|---|
| 1 | 418,04 | 4,13 | 436,03 |
| 4 | 120,38 | 14,35 | 508,9 |
| 8 | 77,308 | 22,35 | 620,9 |
| 16 | 50,85 | 33,98 | 859,18 |
| 32 | 30,81 | 56,09 | 1180,04 |

The performance and energy results on both PCP systems show the expected behaviour. Increasing the number of nodes, speed up diverges from linear. As a result, the energy to solution increases.

Comparing Frioul-PCP with DAVIDE, it seems that KNL is more efficient for case A (small), both KNL and Power8+P100 with SMT off are comparable in performance and energy. When Power8+ SMT is turned on, we get a speed up ~2 while energy consumption is almost the same with SMT off.

## 3.7 NAMD

NAMD[9][10][11] is a widely used molecular dynamics application designed to simulate bio-molecular systems on a wide variety of compute platforms.

It is written in C++ and parallelised using Charm++ parallel objects, which are implemented on top of MPI.

### 3.7.1 *Test case 1 metrics*

**Table 23: NAMD test case 1 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 3 955,17 | 1 300 |
| 2 | 2 085,82 | 1 400 |
| 4 | 1 181,52 | 1 500 |
| 8 | 695,57 | 1 600 |
| 16 | 464,85 | 2 300 |

**Table 24: NAMD test case 1 metrics on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 3 616,50 | 3 575,67 |
| 2 | 2 609,08 | 4 999,39 |
| 4 | 1 503,56 | 5 627,77 |
| 8 | 721,72 | 5 407,02 |
| 16 | 470,97 | 7 037,86 |

### 3.7.2 *Test case 2 metrics*

**Table 25: NAMD test case 2 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 16 | 11 280,23 | 48 200 |
| 32 | 6 624,53 | 72 000 |
| 64 | 5 280,57 | 91 900 |

**Table 26: NAMD test case 2 metrics on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 8 | 1846,99 | NC[4] |
| 16 | 1078,34 | NC[5] |
| 32 | 608,43 | 20 224,81 |
| 40 | 529,71 | 22 896,61 |

The behaviour on DAVIDE is as expected: increasing performance with small increase in energy to solution. On the other hand, on Frioul-PCP, the performance is quite lower than that

[4] No results available du to energy software stack errors
[5] No results available du to energy software stack errors

of DAVIDE, while the required energy to solution is higher. Frioul-PCP was configured with Flat Memory. NAMD cases use large amount of memory. In previous Prace-4IP accelerated Benchmarks there was a clear increase of performance with Cache mode.

## 3.8 PFARM

PFARM is part of a suite of programs based on the 'R-matrix' ab-initio approach to the varitional solution of the many-electron Schrödinger equation for electron-atom and electron-ion scattering.

It is parallelised using hybrid MPI / OpenMP and CUDA offloading to GPU.

### 3.8.1 *Test case 1 metrics*

**Table 27: PFARM test case 1 metrics on Frioul-PCP**

| Number of full PCP-KNL nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 1 702 | 420,5 |
| 2 | 900 | 432,5 |
| 4 | 555 | 504,1 |
| 8 | 695 | 1 100,0 |
| 16 | 487 | 1 400,0 |

**Table 28: PFARM test case 1 metrics on DAVIDE**

| Number of full Davide nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 441,45 | 256,96 |
| 2 | 266,29 | 315,61 |
| 4 | 199,44 | 583,13 |
| 8 | 165,36 | 922,05 |
| 16 | 167,61 | 3 073,01 |

Time to solution is decreasing for both DAVIDE and Frioul-PCP when using more nodes. Regarding performance on DAVIDE speedup is between 1,6 and 2,6 when compared with 1 node and on Frioul-PCP speedup is between 1.8 and 3.4 when compared with 1 node. PFARM code for DAVIDE is 2,7 to 4,2 times faster in comparison with Frioul-PCP. For both systems energy consumption is increasing when more nodes are used. For 1, 2 and 8 nodes Frioul-PCP consumes more energy than DAVIDE – respectively 63%, 37% and 19%. And for 4 and 16 nodes DAVIDE consumes more energy than PCP-KNL – respectively 13% and 54%.

## 3.9 QCD

The theory of how quarks and gluons interact to form nucleons and other elementary particles is called Quantum Chromo Dynamics (QCD).

The QCD benchmark benefits of two different implementations:

- One[9][10][11] benchmark used here is derived from the MILC code (v6), and consists of a full conjugate gradient solution using Wilson fermions. The benchmark is consistent with "QCD kernel E" in the full UAEBS.
- The second[9][10][11] consists of two kernels, the QUDA and the QPhix library. The library QUDA is based on CUDA and optimized for running on NVIDIA GPUs.

### 3.9.1 *First implementation metrics*

**Table 29: QCD part 1 test case 1 metrics on Frioul-PCP 68 OpenMP thread per node**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 151 | 48,7 |
| 2 | 86,9 | 55,8 |
| 4 | 52,7 | 66,8 |
| 8 | 36,5 | 89,8 |
| 16 | 27,8 | 124,4 |
| 32 | 15,6 | 162,4 |
| 64 | 11,7 | 268,1 |

**Table 30: QCD part 1 test case 1 metrics on Frioul-PCP 68 MPI tasks per node**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 110,0 | 41,6 |
| 2 | 62,7 | 47,6 |
| 4 | 39 | 61,2 |
| 8 | 29,3 | 87,7 |
| 16 | 38,3 | 201,6 |
| 32 | 61,0 | 569,1 |
| 64 | 150,0 | 2 600,00 |

**Table 31: QCD part 1 test case 1 metrics on DAVIDE**

| Number of full Davide nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 21,4 | 21,6 |
| 2 | 14,8 | 28,1 |
| 4 | 10,1 | 39,5 |
| 8 | 6,94 | 46,42 |
| 16 | 4,88 | 73,26 |
| 32 | 3,92 | 122,36 |

**Table 32: part 1 test case 2 metrics on PCP-KNL 68 OpenMP thread per node**

| Number of full PCP-KNL nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 4 | 330 | 369,3 |
| 8 | 183 | 412,1 |
| 16 | 102 | 471,2 |
| 32 | 63.6 | 577,6 |
| 64 | 43.1 | 801,5 |

**Table 33: QCD part 1 test case 2 metrics on DAVIDE**

| Number of full Davide nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 84,2 | 85,04 |
| 2 | 53,6 | 105,06 |
| 4 | 33,9 | 123,71 |
| 8 | 22,4 | 142,69 |
| 16 | 15,1 | 196,00 |
| 32 | 9,4 | 256,85 |

Test case 1 consists of a 64x64x64x8 lattice with a constant number of conjugate gradient iterations given by 1000. In case of test case 2 the lattice size is increased to 64x64x64x32.

The shorter run time of the application on the GPU nodes results into a better energy to solution ratio. The benchmark-kernel shows on both architectures a good scaling; however, the kernel is not fully optimized to the specific architectures.

### 3.9.2 *Second implementation metrics*

**Table 34: QCD part 2 test case 1 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Optional metrics (GFLOP/S) | Energy to solution (kJ) |
|---|---|---|---|
| 1 | 81,9 | 184,729 | 34,1 |
| 2 | 56,1 | 269,705 | 39,9 |
| 4 | 34,3 | 441,534 | 49,8 |
| 8 | 24,6 | 614,466 | 65,8 |
| 16 | 23,5 | 644,303 | 117,0 |
| 32 | 16,1 | 937,755 | 171,2 |
| 64 | 18,9 | 800,514 | 375,0 |

**Table 35: QCD part 2 test case 1 metrics on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metrics (GFLOP/S) | Energy to solution (kJ) |
|---|---|---|---|
| 1 | 3,76 | 1 533,13 | 14,90 |
| 2 | 4,88 | 3 005,07 | 19,81 |
| 4 | 3,72 | 5 409,18 | 26,46 |
| 8 | 4,04 | 7 248,57 | 43,07 |
| 16 | 4,86 | 3 490,27 | 88,14 |
| 32 | 4,86 | 4 570,13 | 288,51 |

**Table 36: QCD part 2 test case 2 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution | Optional metrics (GFLOP/S) | Energy to solution (kJ) |
|---|---|---|---|
| 8 | 194,6 | 828,94 | 522.8 |
| 16 | 126,8 | 1 272,43 | 611.3 |
| 32 | 78,2 | 2 063,40 | 755.2 |
| 64 | 57,2 | 2 819,23 | 1100.0 |

Test case 1 consists of a 96x32x32x32 lattice, while test case 2 is given by a 128x64x64x64 one. The benchmark kernels are optimized for the specific architectures which results into different implementation of the underlying conjugate gradient solver. Due to that only relative numbers can be compared. Benchmark kernels show for the test case 1 a good scaling up to 8 nodes however for larger number of nodes the time to solution stagnate. Note that the benchmark-kernel optimized for NVIDIA GPUs, QUDA, is tuned such that it gains optimal performance for the specific configuration. This results into a better scaling of GFLOP/S compared to the time to solution.

## 3.10 Quantum Espresso

QUANTUM ESPRESSO[9][10][11] (or QE) is an integrated suite of computer codes for electronic-structure calculations and materials modelling, based on density-functional theory, plane waves, and pseudopotentials. Since only one of these codes, the PWSCF program, has been ported to the GPU on DAVIDE, we restricted our benchmarks to this application.

It is implemented using MPI and CUDA, offloading to GPU. For DAVIDE 4 MPI tasks per node were used (i.e 1 task per GPU), while on the Frioul-PCP jobs allocated 68 tasks per node. The hybrid MPI/OpenMP version was not used in these tests.

For the test cases, we chose the PRACE UEABS Small Test Case (called AUSURF) for Test Case 1, while for the second Test Case 2 we used an input called TA2O5, available from the program developers. This was chosen because it has lower memory footprint than the PRACE UEABS Large Test Case (CNT) which makes running on DAVIDE easier since the CUDA port of QE runs almost entirely on the GPU and so can access at most up to 64 Gb/node.

For the Test Case 1, where the energy of an atomic system is optimized until convergence with a specified tolerance, we ran the input without modification. For Test Case 2 a similar calculation was performed but the default tolerance was lowered to allow convergence to be obtained within a wall time of no more than a few hours in the worst cases (i.e. with few nodes).

### 3.10.1 *Test case 1 metrics*

**Table 37: Quantum Espresso test case 1 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 2 062,0 | 682,0 |
| 2 | 1 442,0 | 620,4 |
| 4 | 1 063,0 | 676,1 |
| 8 | 659,0 | 1024,0 |
| 16 | 728,0 | 1400,0 |

**Table 38: Quantum Espresso test case 1 metrics on DAVIDE**

| Number of full DAVIDE GPU nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 1 | 312 | 266,99 |
| 2 | 248 | 379,49 |
| 3 | 200 | 432,58 |
| 4 | 197 | 591,36 |

### 3.10.2 *Test case 2 metrics*

**Table 39: Quantum Espresso test case 2 metrics on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 10 | 5 916 | 16 000 |
| 15 | 3 549 | 14 900 |
| 20 | 3 886 | 20 000 |
| 30 | 3 539 | 29 200 |

**Table 40: Quantum Espresso test case 2 metrics on DAVIDE**

| Number of full DAVIDE GPU nodes | Time to solution (s) | Energy to solution (MJ) |
|---|---|---|
| 2 | 2 337 | 3 920,86 |
| 4 | 1 511 | 4 842,34 |
| 5 | 1 470 | 5 835,58 |
| 6 | 1 324 | 6 126,43 |

| Number of full DAVIDE GPU nodes | Time to solution (s) | Energy to solution (MJ) |
|---|---|---|
| 8 | 995 | 5 982,44 |
| 10 | 1 041 | 8 005,33 |
| 20 | 1 189 | 16 107,56 |

The times to solution required for both inputs on the Frioul-PCP scale as expected for KNL even though the absolute values are higher when compared to similar architectures, such as, for example, the KNL partition of Marconi (CINECA). One reason may be due to the fact that the default mode on Marconi is cache mode, while on the Frioul-PCP it is flat. Unfortunately, it was not possible to try cache mode on the Frioul-PCP to test this.

The DAVIDE GPU times to solution are much lower, exhibiting speedups of between 3 and 6 times when compared to the KNLs. The energy to solution results exhibit reasonable trends, the increase due to increasing number of nodes attenuated by decreasing wall times until the scaling limit. The GPU energies are much smaller than those of the KNLs, although we should emphasise that due to time constraints it was not possible to optimise the calculations for KNL using, for example, cache mode, OpenMP threads or QE runtime options.

## 3.11 SHOC

The Accelerator Benchmark Suite[9] will also include a series of synthetic benchmarks.

SHOC is written in C++ and is MPI-based. Offloading for accelerators is implemented through CUDA and OpenCL for GPUs.

Being a synthetic benchmark, SHOC does not really fit the time and energy to solution paradigm as the other scientific benchmarks. However, it has been included for completeness (although "solution" does not represent much in this case) on some representative benchmarks.

As an interesting note, all compute-bound workloads draw around 1200W on average, whereas the memory-bound ones only around 750W.

The three first test cases shows GFLOPS as optional metrics. These tests have been carried in Single Precision (SP) and Double Precision (DP).

### 3.11.1 *Test case 1, GEMM*

**Table 41: SHOC metrics test case GEMM on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (GFLOPS SP/DP) | Energy to solution (kJ) |
|---|---|---|---|
| 1 node - 1 GPU | 193 | 8901/4202 | 140 |
| 1 node - 4 GPUs | 226 | 35320/17276 | 289 |

### 3.11.2 *Test case 2, FFT*

**Table 42: SHOC metrics test case FFT on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (GFLOPS SP/DP) | Energy to solution (kJ) |
|---|---|---|---|
| 1 node - 1 GPU | 54 | 1467/734 | 34,7 |
| 1 node - 4 GPUs | 166 | 5900/2940 | 126 |

### 3.11.3 *Test case 3, MaxFlops*

**Table 43: SHOC metrics test case MaxFlops on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (GFLOPS SP/DP) | Energy to solution (kJ) |
|---|---|---|---|
| 1 node - 1 GPU | 43 | 10475/5318 | 37,2 |
| 1 node - 4 GPUs | 22 | 41904/21276 | 51,6 |

### 3.11.4 *Test case 4, Triad*

**Table 44: SHOC metrics test case Triad on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric (GB/s) | Energy to solution (kJ) |
|---|---|---|---|
| 1 node - 1 GPU | 37 | 41,3 | 24 |
| 1 node - 4 GPUs | 38 | 142,8 | 28,8 |

### 3.11.5 *Test case 5, MD5Hash*

**Table 45: SHOC metrics test case MD5Hash on DAVIDE**

| Number of full DAVIDE nodes | Time to solution (s) | Optional metric GH/s | Energy to solution (kJ) |
|---|---|---|---|
| 1 node - 1 GPU | 104 | 15,87 | 70,7 |
| 1 node - 4 GPUs | 106 | 60,3 | 125 |

### 3.11.6 *Full SHOC benchmark results*

Table 46 shows SHOC wrap-up table. The first column indicate the micro benchmark while the second and the third indicate performances for respectively one and four GPUs. The latter have been normed so that both column can be comparable.

**Table 46: SHOC full metrics on DAVIDE**

| Device/Bench | Power 8 + P100 CUDA (DAVIDE 1GPU) | Power 8 + P100 CUDA (DAVIDE 4GPU) – res * 4 |
|---|---|---|
| BusSpeedDownload | 32.90 GB/s | 30.67 GB/s |
| BusSpeedReadback | 34.00 GB/s | 27.76 GB/s |
| maxspflops | 10475 GFLOPS | 10476 GFLOPS |
| maxdpflops | 5318 GFLOPS | 5319 GFLOPS |
| gmem_readbw | 574.53 GB/s | 544.37 GB/s |
| gmem_readbw_strided | 98.65 GB/s | 98.63 GB/s |
| gmem_writebw | 436 GB/s | 436.9 GB/s |
| gmem_writebw_strided | 26.15 GB/s | 26.2 GB/s |
| lmem_readbw | 4245 GB/s | 4256 GB/s |
| lmem_writebw | 5485 GB/s | 5500 GB/s |
| BFS | **64,5 MEdges/s** | N/A |

| Device/Bench | Power 8 + P100 CUDA (DAVIDE 1GPU) | Power 8 + P100 CUDA (DAVIDE 4GPU) – res * 4 |
|---|---|---|
| FFT_sp | 1467 GFLOPS | 1475 GFLOPS |
| FFT_dp | 734 GFLOPS | 735 GFLOPS |
| SGEMM | 8732-8901 GFLOPS | 8830 GFLOPS |
| DGEMM | 3654-4202 GFLOPS | 4319 GFLOPS |
| MD (SP) | 522 GFLOPS | 479 GFLOPS |
| MD5Hash | 15.87 GH/s | 15.09 GH/s |
| Reduction | 270 GB/s | 270 GB/s |
| Scan | 98.5 GB/s | 98.5 GB/s |
| Sort | 12.52 GB/s | 12.53 GB/s |
| Spmv | 23-65 GFLOPS | 23-57 GFLOPS |
| Stencil2D | 470 GFLOPS | 414 GFLOPS |
| Stencil2D_dp | 258 GFLOPS | 214 GFLOPS |
| Triad | 41.3 GB/s | 35.7 GB/s |
| S3D (level2) | 292 GFLOPS | 291 GFLOPS |

## 3.12 Specfem3D_Globe

The software package SPECFEM3D_Globe[9][10][11] simulates three-dimensional global and regional seismic wave propagation based upon the spectral-element method.

It is written in Fortran and uses MPI combined with OpenMP to achieve parallelisation.

Test cases used here from the accelerated benchmark suite[9] and simulate simple earth model response. It has been setup by application developers.

Comparing to PRACE 4IP runs[9], KNL performance are 5 time slower. This is due to the fact that a code modified by Intel were used but as Intel didn't release the code publicly yet and doesn't seems to plan to do so, the current public code [12] have been used to carry these performances.

### 3.12.1 *Test case 1*

**Table 47: Specfem3D Globe metrics test case 1 on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 4 | 261 | 221,5 |

**Table 48: Specfem3D Globe metrics test case 1 on DAVIDE**

| Number of full DAVIDEP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 2 | 67 | 106,5 |

### 3.12.2 *Test case 2*

**Table 49: Specfem3D Globe metrics test case 2 on Frioul-PCP**

| Number of full Frioul-PCP nodes | Time to solution (s) | Energy to solution (kJ) |
|---|---|---|
| 5 | 352 | 363,5 |
| 10 | 272 | 501,0 |

## 3.13 Wrap-up table

Table 50 gathers scientific application performances and energy metric. On metric by test case by machine by code have been picked at best scalability parameters. This table shows that both architectures can produce similar results in case the code has comparable performances, while some code performs better in one or the other architecture. In any case, it is recommended to refer to detailed code section to understand better performance and energy interaction.

**Table 50: Wrap-up table gathering main results for scientific codes**

| Code | Test case # | Power8 + GPU | | | KNL | | |
|---|---|---|---|---|---|---|---|
| | | Node # | Time (s) | Energy (kJ) | Node # | Time (s) | Energy (kJ) |
| ALYA | 1 | 4 | 14,82 | 196,56 | 8 | 13,38 | 47,85 |
| | 2 | 32 | 14,77 | 1 821,99 | 32 | 28,68 | 470,00 |
| Code Saturne | 1 | 2 | 118,00 | 182,04 | 4 | 651,00 | 778,00 |
| | 2 | *NC* | *NC* | *NC* | 8 | 442,33 | 889 900 |
| CP2K* | 1 | 16 | 323,00 | 3 166,73 | 32 | 231,00 | 1 795,50 |
| | 2 | 16 | 1 695,00 | 17 314,52 | 32 | 226,00 | 1 857,00 |
| GADGET | 1 | *NC* | *NC* | *NC* | 8 | 897.9 | 1 700,00 |
| GPAW | 1 | *NC* | *NC* | *NC* | 4 | 187,00 | 277,00 |
| | 2 | *NC* | *NC* | *NC* | 4 | 129,00 | 231,00 |
| GROMACS | 1 | 2 | 226,28 | 390,03 | 4 | 278,13 | 287,10 |
| | 2 | 4 | 120,38 | 508,90 | 8 | 183,35 | 603,50 |
| NAMD | 1 | 8 | 721,72 | 5 407,02 | 8 | 695,57 | 1 600,00 |
| | 2 | 40 | 529,71 | 22 896,61 | 23 | 6 624,53 | 72 000,00 |
| PFARM | 1 | 1 | 441,45 | 256,96 | 4 | 555,00 | 504,10 |
| QCD part 1 | 1 | 1 | 21.4 | 21,60 | 2 | 62,70 | 47,60 |
| | 2 | 2 | 53,60 | 105,06 | 32 | 63,60 | 577,60 |
| QCD part 2 | 1 | 1 | 3.76 | 14,90 | 4 | 34.3 | 49,80 |
| | 2 | *NC* | *NC* | *NC* | 32 | 78.2 | 755,20 |
| Quantum Espresso | 1 | 1 | 312,00 | 266,99 | 8 | 659,00 | 1 024,00 |
| | 2 | 4 | 1 511,00 | 4 842,34 | 15 | 3 549,00 | 14 900,00 |
| Specfem3D Globe | 1 | 2 | 67,00 | 106,50 | 4 | 261,00 | 221,50 |
| | 2 | *NC* | *NC* | *NC* | 5 | 352,00 | 363,50 |

# 4 Energetic Analysis of a Solver Stack for Frequency-Domain Electromagnetics

This work is concerned with the energetic analysis of the combined HORSE/MaPHyS numerical tool developed at Inria. The HORSE[16] (High Order solver for Radar cross Section Evaluation) simulation software implements an innovative high order finite element type method for solving the system of three-dimensional frequency-domain Maxwell equations. From the computational point of view, the central operation of a HORSE simulation is the solution of a large sparse and indefinite linear system of equations. High order approximation is particularly interesting for solving high frequency electromagnetic wave problems and, in that case, the size of this linear system can easily exceed several million unknowns. In this study, we adopt the MaPHyS[17] hybrid iterative-direct sparse system solver, which is based on domain decomposition principles.

## 4.1 Numerical approach

During the last 10 years, Discontinuous Galerkin (DG) methods have been extensively considered for obtaining an approximate solution of Maxwell's equations. Thanks to the discontinuity of the approximation, this kind of methods has many advantages, such as adaptivity to complex geometries using unstructured possibly non-conforming meshes, easily obtained high order accuracy, hp-adaptivity and natural parallelism. However, despite these advantages, DG methods have one main drawback particularly sensitive for stationary problems: the number of globally coupled degrees of freedom (DoF) is much greater than the number of DoF required by conforming finite element methods for the same accuracy. Consequently, DG methods are expensive in terms of both CPU time and memory consumption, especially for time-harmonic problems. Hybridization of DG methods is devoted to address this issue while keeping all the advantages of DG methods. HDG methods introduce an additional hybrid variable on the faces of the elements, on which the definition of the local (element-wise) solutions is based. A so-called conservativity condition is imposed on the numerical trace, whose definition involved the hybrid variable, at the interface between neighbouring elements. As a result, HDG methods produce a linear system in terms of the DoF of the additional hybrid variable only. In this way, the number of globally coupled DoF is reduced. The local values of the electromagnetic fields can be obtained by solving local problems element-by-element. We have recently designed such a high order HDG method for the system of 3D time-harmonic Maxwell's equations [18].

## 4.2 Simulation software

HORSE is a computational electromagnetic simulation software for the evaluation of radar cross section of complex structures. This software aims at solving the full set of 3D time-harmonic Maxwell equations modelling the propagation of a high frequency electromagnetic wave in interaction with irregularly shaped structures and complex media. It relies on an arbitrary high order HDG method that is an extension of the method proposed in [18]. This HDG method designed on an unstructured possibly non-conforming tetrahedral mesh, leads to the formulation of an unstructured complex coefficients sparse linear system of equations for the DoF of the hybrid variable, while the DoF of the components of the electric and magnetic fields are computed element-wise from those of the hybrid variable. This software is written in Fortran 95. It is parallelized for distributed memory architectures using a classical SPMD strategy combining a partitioning of the underlying mesh with a message-passing programming model using the MPI standard. One important computational kernel of this software is the solution of a large sparse linear system of complex coefficients equations. In a

preliminary version of this software, this system was solved using parallel sparse direct solvers such as MUMPS[19] or PaStiX[20]. However, sparse direct solvers are in general poorly scalable when it comes to solve very large linear systems arising from the discretization of 3D problems. In this project, we study the possibility of improving the scalability of HORSE by considering the use of hybrid iterative/direct solvers whose design is based on domain decomposition principles and its impact on the energy consumption.

## 4.3 MaPHyS algebraic solver

The solution of large sparse linear systems is a critical operation for many numerical simulations. To cope with the hierarchical design of modern supercomputers, hybrid solvers based on algebraic domain decomposition methods have been proposed. Among them, approaches consisting of solving the problem on the interior of the domains with a sparse direct method and the problem on their interface with a preconditioned iterative method applied to the related Schur Complement have shown an attractive potential as they can combine the robustness of direct methods and the low memory footprint of iterative methods. MaPHyS (Massively Parallel Hybrid Solver) [21][22] is a parallel linear solver, which implements this idea. The underlying idea is to apply to general unstructured linear systems domain decomposition ideas developed for the solution of linear systems arising from PDEs. The interface problem, associated with the so-called Schur complement system, is solved using a block preconditioner with overlap between the blocks that is referred to as Algebraic Additive Schwarz. To cope with the possible lack of a coarse grid mechanism that enables one to keep the number of iterations constant when the number of blocks is increased, the solver exploits two levels of parallelism (between the blocks using MPI and within the treatment of the blocks using threads). This allows exploiting a large number of cores with a moderate number of MPI tasks, which ensures a reasonable convergence behavior. MaPHyS makes use of a sparse direct solver as a subdomain solver such as PaStiX (Parallel Sparse matriX package) or MUMPS. The parallelization of the direct solver relies on a specific partitioning of the matrix blocks; the core operations are multithreaded allowing a second level of parallelization. PaStiX and MUMPS make extensive use of highly optimized dense linear algebra kernels (e.g. BLAS kernels).

## 4.4 Numerical and performance results

For the numerical simulations reported below we have used Frioul-PCP cluster presented in Section 2.2.

### 4.4.1 *MaPHyS used in standalone mode*

Weak scalability performance of the MaPHyS solver has been investigated in standalone mode. For these experiments, we solve a 3D Poisson problem on a 2.5D domain that corresponds to a beam and a 1D decomposition. Each subdomain has at most two neighbours and is essentially a regular cube of size 403 (i.e., each subdomain has around 64,000 unknowns). The energy performance has been measured with Bull Energy Optimizer (BEO) as the total energy consumed by the job. We also had the opportunity to test Bull's graphic tool HDEEVIZ which shows detailed energy consumption over time (Figure 4). The additional metrics relevant for the performance of MaPHyS are the time for the factorization of interior subdomain unknowns, the time spent in the iterative solver, the number of iterations performed, and the total time spent in the solver. The local matrices are read from files, which is both time and energy consuming but not relevant to MaPHyS performance

since the matrices are usually computed locally and directly provided to the solver by the user.

For our experiments, we consider three numerical configurations of the solver. In Figure 3, they are referred to as:

- dense (in red): we consider the fully assembled local Schur complements to build the additive Schwarz preconditioner;
- sparse (in green): the entries of the local dense Schur complements that are smaller than a given relative threshold (10-5) are discarded, the resulting sparse matrices are used to build the additive Schwarz preconditioner;
- dense+CGC (in blue): in addition to the previously described dense preconditioner a coarse grid correction [23] is applied to ensure that the convergence will be independent from the number of subdomains. In this experiment, we compute five vectors per subdomain to create the coarse grid. The coarse grid being relatively small compared to the global problem, computations are centralized on one process and solved by the direct solver (MUMPS here).

Because the dense and sparse preconditioner do not implement any global coupling numerical mechanisms, the number of iterations is expected to grow as the number of subdomains for the 1D decomposition of the domain and our elliptic test example. This poor numerical behaviour can be observed in Figure 3-Number of iterations-, while the coarse space correction plays its role and ensures several iterations independent from the number of domains. This nice numerical behaviour translates in term of solution time for the iterative part where the variant with the coarse space correction outperforms the other two. However, the overhead of the setup phase for the construction of the coarse grid, which requires the solution of generalized eigen problems, is very high and cannot be amortized if a single right-hand sides has to be solved (which is not the case for, e.g., radar cross section evaluation where many right-hand sides must be solved). The relative ranking of the variants with respect to the time to solution remains the same when we consider the energy criterion. However, the power requirements are different; using simple linear regression the power requirement for the dense preconditioner is around 5 kW, 8kW for the sparse and 10 kW for the two-level preconditioner. The high energy required by the two-level preconditioner is mainly due to the setup of the coarse space correction that is memory and CPU consuming. The fact that the sparse preconditioner is more demanding than the dense might be due to the more irregular memory pattern associated with it, which requires more memory traffic. As can be seen in Figure 4, the memory energy consumption represents a significant part of the total.

Figure 4 shows the detailed energy consumption over time for the case on one node with the dense preconditioner. One can see the setup and analysis parts of the run with low energy consumption. Then looking at the memory curve, one can identify the three steps of the MaPHyS solver. The iterative solver appears quite clearly as a large plateau where the energy cost is high for memory and low for CPU. It is consistent with the fact that this step is memory bound with many communications and relatively few computations. The total energy consumed by the node is 5.6 Wh = 20,160 J, which corresponds to the results given by BEO for this case.

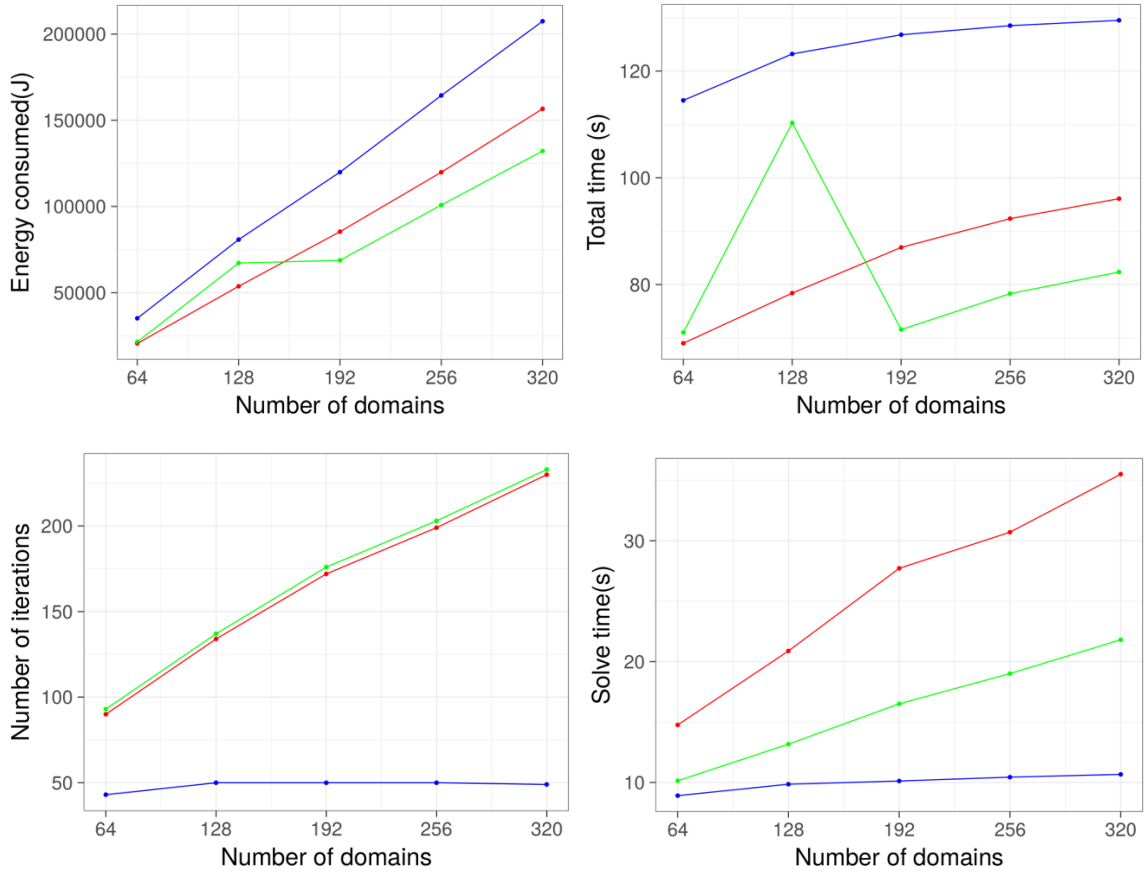


**Figure 3: Weak scaling of MaPHyS from 1 to 5 nodes, with 64 subdomains per nodes and 1 core per subdomain**


**Figure 4: Energy consumption history for the dense preconditioner with hdeeviz (green=CPU, yellow=memory,cyan=total board).**

**Table 51: Size of the global matrix and the global Schur complement matrix solved by MaPHyS in weak scaling.**

| Number of nodes | Number of domains | Global matrix size | Global Schur size |
|---|---|---|---|
| 1 | 64 | 4,305,041 | 211,806 |
| 2 | 128 | 9,033,444 | 426,974 |
| 3 | 192 | 14,202,169 | 642,142 |
| 4 | 256 | 19,826,576 | 857,31 |
| 5 | 320 | 25,922,025 | 1,072,478 |

### 4.4.2 *Scattering of a plane wave by a PEC sphere*

We now consider a more realistic problem that consists in the scattering of plane wave with frequency F=600 MHz by a perfectly electric conducting (PEC) sphere. The contour lines of the x-component of the electric field are visualized in Figure 5 left, and the obtained RCS is plotted in Figure 5 right together with a comparison with a reference RCS obtained from a BEM (Boundary Element Method) calculation. This problem is simulated using the coupled HORSE/MaPHyS numerical tool. The underlying tetrahedral mesh contains 37,198 vertices and 119,244 elements. We have realized a series of calculations for which the number of iterations of the MaPHyS interface solver has been fixed to 100. Simulations are performed using a flat MPI mode. We consider two mains situations: (a) the interpolation order in the HDG discretization method is uniform across the cells of the mesh; (b) the interpolation order is adapted locally to the size of the cell based on goal-oriented criterion. In the latter situation, we distribute the interpolation order such that there are at list 9 integration points (degrees of freedom of the Lagrange basis functions) per local wavelength. For the tetrahedral mesh used in this study, we obtain the following distribution of mesh elements: 12,920 (P1), 70,023 (P2), 31,943 (P3) and 4358 (P4). For a given mesh, a uniform interpolation order is not necessarily the best choice in terms of computational cost versus accuracy, especially if the mesh is unstructured as it is the case here. Increasing the interpolation order allows for a better accuracy at the expense of a larger sparse linear system to be solved by MaPHyS. By distributing the interpolation order according to the size of mesh cells allows for a good compromise between time to solution and accuracy.

Performance and energy consumption figures are reported in Table 52. In this table, the number of subdomains also corresponds to the total number of core or MPI processes. The number of MPI processes per node can be deduced from the number of nodes. First, in most of the tested configurations, we observe a super-linear speedup, as a result of the reduction of the size of the local factors within each subdomain, which is not evolving linearly with the number of subdomains. We first note, as expected, that the energy consumption with higher values of the interpolation order since the size of the problem, i.e. of the HDG sparse linear system, increases drastically. A second noticeable remark is that the energy consumption decreases when the number of MPI processes per node increases for a given number of subdomains, for instance, for the HDG-P1 method, using a decomposition of the tetrahedral mesh in 64 subdomains, the energy consumption is equal to 38,198 J on 4 nodes (i.e. with 16 MPI processes per node) and 68,045 J on 8 nodes (i.e. with 8 MPI processes per node). A similar behaviour is observed for the HDG-P2 method and a 64 subdomains decomposition. A final comment is that the use of a locally adapted distribution of the interpolation order allows a substantial reduction of the energy consumption for a target accuracy. This is in fact the result of lower time to solution because of the reduction of the size of the problem, as can be seen by comparing the figures for the HDG-P4 and HDG-Pk methods with a 256 subdomain decomposition.

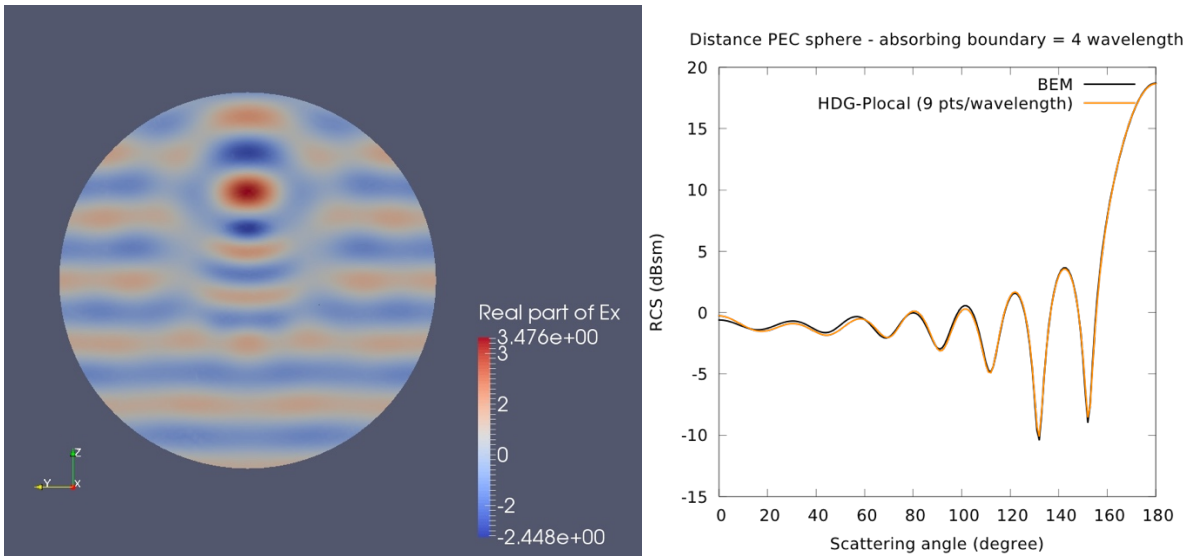


**Figure 5: Scattering of a plane wave by a perfectly electric conducting sphere: contour lines of the x-component of the electric field (left) and RCS (right).**

**Table 52: Performance figures of the coupled HORSE/MaPHyS numerical tool. Scattering of a plane wave by a PEC sphere. Timings for 100 iterations of the interface solver of MaPHyS.**

| Method | Number of subdomains | Number of nodes | Wall time | Energy consumption |
|---|---|---|---|---|
| **HDG-P1** | 16 | 1 | 143.0 sec | 40,507 J |
| | 32 | 2 | 52.4 sec | 35,450 J |
| | 64 | 4 | 21.0 sec | 38,198 J |
| | 64 | 8 | 20.2 sec | 68,045 J |
| | 128 | 16 | 9.5 sec | 98,050 J |
| **HDG-P2** | 64 | 4 | 104.7 sec | 114,302 J |
| | 64 | 8 | 102.6 sec | 198,889 J |
| | 128 | 16 | 38.3 sec | 187,500 J |
| | 256 | 16 | 15.8 sec | 124,516 J |
| **HDG-P3** | 64 | 8 | 415.7 sec | 723,900 J |
| | 128 | 16 | 130.5 sec | 479,855 J |
| | 256 | 16 | 48.7 sec | 239,774 J |
| **HDG-P4** | 128 | 16 | 383.4 sec | 1,286,780 J |
| | 256 | 16 | 132.5 sec | 537,802 J |
| **HDG-Pk, k=1,4** | 128 | 4 | 96.4 sec | 123,084 J |
| | 128 | 8 | 89.5 sec | 186,570 J |
| | 256 | 4 | 35.2 sec | 95,600 J |
| | 256 | 8 | 35.2 sec | 113,699 J |
| | 256 | 16 | 31.1 sec | 179,165 J |

# 5 Conclusion and Outlook

The work performed during the extension represents the first combined performances and energy results for UEABS on KNL and GPU. This deliverable also presents a detailed energy study conducted on Frioul-PCP that starts to explore possibilities available with new energy software and hardware.

These results allow application users and system administrators to get a clearer view of the performances and energy consumption of a wide range of scientific applications. Both architectures have pros and cons depending on the envisioned set of applications. Regarding energy efficiency, a general trend can be observed: non-linear speedup leads to higher energy consumption.

One main feature that is still missing is to assess energy consumption excluding the bias of benchmarks (e.g. not including pre/post processing): as for performance, it would allow energy prediction for a given production run from a benchmark run.

Most of the codes of the PRACE benchmark suite have been run but there are still some combinations of code, test case and platform missing. The PCP systems are invaluable for carrying out such investigations, but more and more machine now come with at least a basic energy measurement system. Adding such metrics where possible would give a greater view of energy consumption of various architectures available. On top of that, some test cases should be redesigned to fit the biggest machine size. Such work could be incorporated into the PRACE-5IP WP7 task on benchmarking that aims at merging standard and accelerated UEABS.

The detailed study in Section 4 shows that there is a lot of room for improvement in the energy-related field in tuning the input parameters of codes as well as in numerical methodology. The PCP systems can clearly help in the area of improving energy-efficiency of codes. This study was conducted on KNL and it would be interesting to perform a similar study on GPU.

Ultimately, the lack of time to investigate the FPGA prototype is a significant disappointment of this work. While this system looks very hard to take advantage from, figures revealed by Maxeler in term of energy efficiency looks very promising. More effort should be spent on attempting to use this machine in the future.