

**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**

***Draft***

Version: 0.1

Author(s): Victor Cameo Ponz, CINES

Date: 21.12.2017

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:** <0.1> | **Status:** |
| **Available at:** <http://www.prace-project.eu> | |
| **Software Tool:** Microsoft Word 2010 | |
| **File(s):** d77\_0.1.docx | |
| **Authorship** | **Written by:** | Victor Cameo Ponz, |
| **Contributors:** | NN, Organisation |
| **Reviewed by:** | NN, Organisation |
| **Approved by:** | MB/TB |

Document Status Sheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Status** | **Comments** |
| 0.1 | 21/12/2017 | Draft |  |
| 0.2 | DD/Month/YYYY |  |  |
| 1.0 | DD/Month/YYYY | Final version |  |

Document Keywords

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

**Disclaimer**

This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement n° EINFRA-653838. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements. Please note that even though all participants to the Project are members of PRACE AISBL, this deliverable has not been approved by the Council of PRACE AISBL and therefore does not emanate from it nor should it be considered to reflect PRACE AISBL’s individual opinion.

**Copyright notices**

© 2017 PRACE Consortium Partners. All rights reserved. This document is a project document of the PRACE project. All contents are reserved by default and may not be disclosed to third parties without the written consent of the PRACE partners, except as mandated by the European Commission contract EINFRA-653838 for reviewing and dissemination purposes. All trademarks and other rights on third party products mentioned in this document are acknowledged as own by the respective holders.

Table of Contents

Project and Deliverable Information Sheet i

Document Control Sheet i

Document Status Sheet i

Document Keywords ii

Table of Contents iii

List of Figures iv

List of Tables iv

References and Applicable Documents v

List of Acronyms and Abbreviations vi

List of Project Partner Acronyms viii

Executive Summary 1

1 Introduction 1

2 Clusters specifications and access 1

2.1 Access to machines 2

2.2 Xeon Phi 2

2.2.1 Compute technology 3

2.2.2 Energy sampling technology 3

2.3 Power8 + GPU 4

2.3.1 Compute technology 4

2.3.2 Energy sampling technology 5

3 Performances and energy metrics of UEABS on PCP systems 6

3.1 ALYA 6

3.1.1 Test case 1 metrics 7

3.1.2 Test case 2 metrics 7

3.2 Code\_Saturne 7

3.2.1 Test case 1 metrics 7

3.2.2 Test case 2 metrics 7

3.3 CP2K 8

3.3.1 Test case 1 metrics 8

3.3.2 Test case 2 metrics 8

3.4 GADGET 9

3.4.1 Test case 1 9

3.5 GPAW 9

3.5.1 Test case 1 metrics 9

3.5.2 Test case 2 metrics 9

3.6 GROMACS 10

3.6.1 Test case 1 metrics 10

3.6.2 Test case 2 metrics 10

3.7 NAMD 11

3.7.1 Test case 1 metrics 11

3.7.2 Test case 2 metrics 11

3.8 NEMO 12

3.9 PFARM 12

3.9.1 Test case 1 metrics 12

3.10 QCD 12

3.11 Quantum Espresso 12

3.11.1 Test case 1 metrics 13

3.11.2 Test case 2 metrics 13

3.12 SHOC 13

3.12.1 Test case GEMM 14

3.12.2 Test case FFT 14

3.12.3 Test case MaxFlops 14

3.12.4 Test case Triad 14

3.12.5 Test case MD5Hash 14

3.12.6 Full SHOC benchmark results 14

3.13 Specfem3D\_Globe 15

3.13.1 Test case 1 15

3.13.2 Test case 2 16

3.14 Wrap-up table 16

4 Energetic Analysis of a Solver Stack for Frequency-Domain Electromagnetics 16

4.1 Numerical approach 16

4.2 Simulation software 17

4.3 MaPHyS algebraic solver 17

4.4 Numerical and performance results 17

4.4.1 MaPHyS used in standalone mode 18

4.4.2 Scattering of a plane wave by a PEC sphere 20

5 Conclusion 21

List of Figures

[Figure 1 4IP-extention project timeline. On top of the figure are printed periods names and on the bottom key dates. Periods in grey stands for task preparation, periods in blue stands for documentation redaction and period in green stand for technical work. 2](#_Toc501642298)

[Figure 2 Example of Grafana HTML output 4](#_Toc501642299)

[Figure 3 Weak scaling of MaPHyS from 1 to 5 nodes, with 64 subdomains per nodes and 1 core per subdomain 19](#_Toc501642300)

[Figure 4 Energy consumption history for the dense preconditioner with hdeeviz (green=CPU, yellow=memory,cyan=total board). 19](#_Toc501642301)

[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). 21](#_Toc501642302)

List of Tables

Table 1 PCP Systems access dates 2

Table 2 Alya test case 1 metrics on DAVIDE 7

Table 3 Alya test case 2 metrics on DAVIDE 7

Table 4 Code Saturn test case 1 metrics on PCP-KNL 7

Table 5 Code Saturn test case 1 metrics on DAVIDE 7

Table 6 CP2K test case 1 metrics on PCP-KNL 8

Table 7 CP2K test case 1 metrics on DAVIDE 8

Table 8 CP2K test case 2 metrics on PCP-KNL 8

Table 9 CP2K test case 2 metrics on DAVIDE 8

Table 10 Gadget test case 1 metrics with 4 MPI task per node and 16 OpenMP thread per task 9

Table 11 Gadget test case 1 metrics on 8 PCP-KNL nodes 9

Table 12 GPAW test case 1 metrics on PCP-KNL 9

Table 13 GPAW test case 2 metrics on PCP-KNL 9

Table 14 GROMACS test case 1 metrics on PCP-KNL 10

Table 15 GROMACS test case 1 metrics on DAVIDE 10

Table 16 GROMACS metrics on PCP KNL 10

Table 17 GROMACS test case 2 metrics on PCP-KNL 10

Table 18 GROMACS test case 2 metrics on DAVIDE with SMT off (i.e. SMT=1) 10

Table 19 GROMACS test case 2 metrics on DAVIDE with SMT=8 11

Table 20 NAMD test case 1 metrics on PCP-KNL 11

Table 21 NAMD test case 1 metrics on DAVIDE 11

Table 22 NAMD test case 2 metrics on PCP-KNL 11

Table 23 NAMD test case 2 metrics on DAVIDE 12

Table 24 PFARM test case 1 metrics on PCP-KNL 12

Table 25 PFARM test case 1 metrics on DAVIDE 12

Table 26 PFARM test case 2 metrics on PCP-KNL 13

Table 27 PFARM test case 2 metrics on DAVIDE 13

Table 28 PFARM test case 2 metrics on PCP-KNL 13

Table 29 PFARM test case 2 metrics on DAVIDE 13

Table 30 SHOC metrics test case GEMM on DAVIDE 14

Table 31 SHOC metrics test case FFT on DAVIDE 14

Table 32 SHOC metrics test case MaxFlops on DAVIDE 14

Table 33 SHOC metrics test case Triad on DAVIDE 14

Table 34 SHOC metrics test case MD5Hash on DAVIDE 14

Table 35 SHOC full metrics on DAVIDE 14

Table 36 Specfem3D Globe metrics test case 1 on PCP-KNL 15

Table 37 Specfem3D Globe metrics test case 2 on PCP-KNL 16

Table 38 Size of the global matrix and the global Schur complement matrix solved by MaPHyS in weak scaling. 20

Table 39 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. 21

References and Applicable Documents

1. Bull website: https://bull.com/
2. CINES : <https://www.cines.fr/>
3. DAVIDE dedicated webpage https://www.e4company.com/en/?id=press&section=1&page=&new=davide\_supercomputer
4. E4 computer engineering: https://www.e4company.com
5. CINECA: http://hpc.cineca.it/
6. PRACE 4IP D7.5 Application performance on Accelerators – V. Cameo Ponz et al. – 2016
7. The Unified European Application Benchmark Suite – <http://www.prace-ri.eu/ueabs/>
8. D7.4 Unified European Applications Benchmark Suite – Mark Bull et al. – 2013
9. HORSE: <http://www-sop.inria.fr/nachos/index.php/Software/HORSE>
10. MaPHyS: <https://gitlab.inria.fr/solverstack/maphys>
11. 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)
12. 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)
13. 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)
14. 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)
15. 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)
16. 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

GB Giga (= 230 ~ 109) Bytes (= 8 bits), also GByte

Gb/s Giga (= 109) bits per second, also Gbit/s

GB/s Giga (= 109) 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 (= 109) Floating point operations (usually in 64-bit, i.e. DP) per second, also GF/s

GHz Giga (= 109) Hertz, frequency =109 periods or clock cycles per second

GPU Graphic Processing Unit

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 (= 210 ~103) Bytes (= 8 bits), also KByte

LINPACK Software library for Linear Algebra

MB Management Board (highest decision making body of the project)

MB Mega (= 220 ~ 106) Bytes (= 8 bits), also MByte

MB/s Mega (= 106) Bytes (= 8 bits) per second, also MByte/s

MFlop/s Mega (= 106) 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 (= 240 ~ 1012) 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 (= 1012) 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

RISCSW RISC Software GmbH

RZG Max Planck Gesellschaft zur Förderung der Wissenschaften e.V., Germany (3 rd Party to GCS)

SIGMA2 UNINETT Sigma2 AS, Norway

SNIC Swedish National Infrastructure for Computing (within the Swedish Science Council), Sweden

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 (3 rd 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)

*Depending on the size (number of pages) of the front matter an empty page has to be inserted to force the Executive Summary (Page 1) to the top of a sheet when printed. Simply specifying Section Change to Odd or Even seems not to work reliably.*

Executive Summary

This document describes efforts deployed in order to exploit PRACE Pre-Commercial Procurement (PCP) machines. It aims at giving an overview of what can be done on in term 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 a more detailed case study porting a solver stack using cutting edge tools.

This work has been undertaken by the 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 Fourth Implementation Phase (4IP).

It also heads in the direction of the Task 7.3 in 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 have been selected to be ported on 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.

## Introduction

The work produced within this task is driven by the delivery of PRACE PCP machines. It aims at giving manufacturer-independent performance and energy metrics for future hexa-scale systems. It is also an opportunity to explore and test cutting edge energy hardware stack and tool developed within the scope of PCP.

As stated in the Milestone 33, this document will present metrics for selected code among the UEABS. It allows to show results concerning many fields used among European scientific communities. As well as it will go deeper in the porting and energetic profiling activities using the HORSE+MaPHyS+PaStiX solver stack as example.

Section 2 will details hardware and software specifications where metrics have been carried out. On 3 the metrics for UEABS will be bring together. The work on porting and energy profiling will be presented in section 4. Section 5 will conclude and outline further work on PCP prototypes.

## Clusters specifications and access

PRACE PCP project include tree different prototypes 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 making it harder get familiar with.

As demonstrated in section 2.1 tight deadlines didn't let the time to produce relevant metrics on the FPGA cluster. Therefore, only GPU and KNL prototype are presented here.

### Access to machines

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

The Figure 1 outlines the initial tight deadlines for this project. Also, showing that access to machines have been possible quite late during the phase for running codes.

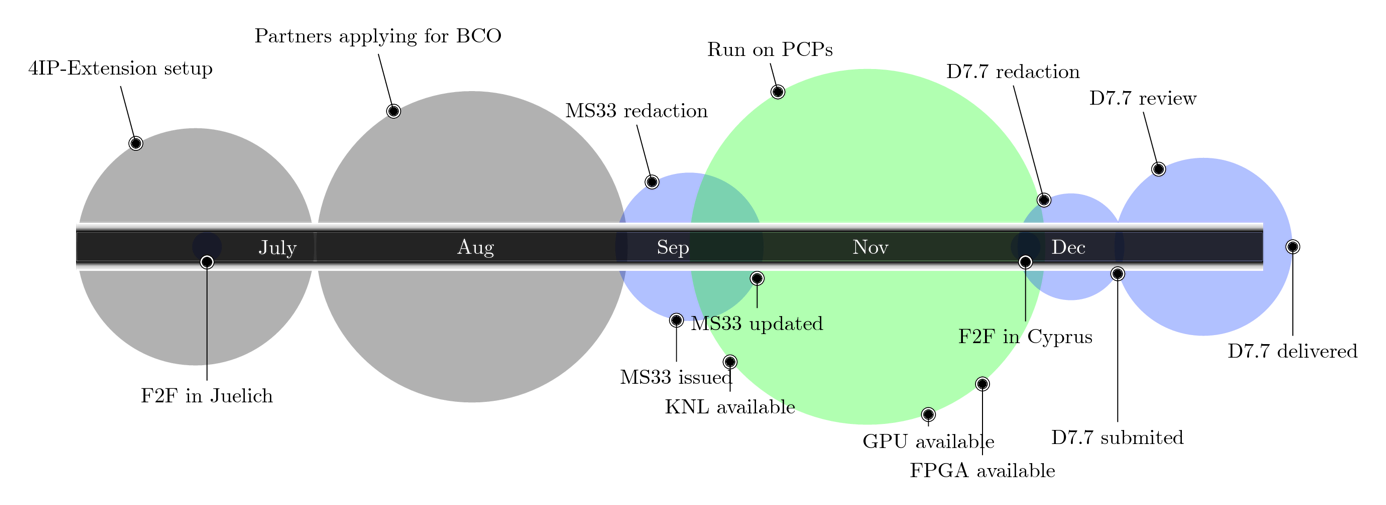


Figure 1 4IP-extention project timeline. On top of the figure are printed periods names and on the bottom key dates. Periods in grey stands for task preparation, periods in blue stands for documentation redaction and period in green stand for technical work.

The Table 1 shows the precise timeline. To this delays some technical interruptions occurred right at the end of the running phase, not helping with the redaction of this document:

*PCP-KNL:*

* 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-GPU:*

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

Table 1 PCP Systems access dates

|  |  |  |  |
| --- | --- | --- | --- |
|  | KNL | GPU | FPGA |
| 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 | / |

### Xeon Phi

This machine has been designed by Atos/Bull[1] and is hosted at CINES[2] in Montpellier, France. It is made of 76 Bull Sequana X1210 blades, each including 3t Xeon Phi KNL nodes. It totals a theoretical peak performance of 465 Tflop/s with an estimated consumption of 82kW[[1]](#footnote-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

#### 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 will be made through the Grafana web interface as show in Figure 2.



Figure 2 Example of Grafana HTML output

**BEO**

BEO is a system administrator oriented tools that allow to get energy metrics at switch and node level. At user level the main interesting feature is the 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 € |

### Power8 + GPU

D.A.V.I.D.E has been designed by E4 computer engineering[4] and is hosted at CINECA[5] in Bologna, Italy. It totals a theoretical peak performance of 990 TFlop/s (double precision). A more detailed description can be found on the E4 dedicated webpage[3].

#### Compute technology

Hardware features fat-nodes with the following design:

* 45 nodes with
  + x2 IBM POWER8+ processors, i.e. 8x2 cores with Simultaneous Multi-Threading (SMT) 8
  + x4 NVIDIA P100 GPU with 16GB High Bandwidth Memory 2 (HBM2)
* intranode communications integrated using NVLink
* extranode communications integrated using Infiniband ERD interconnect in fat-tree with no oversubscription topology
* CPU and GPU direct hot water (~27°C) cooling, removing 75-80% of the total heat
* 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]](#footnote-2).

#### Energy sampling technology

Information is collected from processors, memory, GPUs and fans exploiting Analog-to-Digital Converter 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 straight forward and has the following verbose output:

$ get\_job\_energy 12389

Job 12389

- Duration (seconds): 421.0

- Used Node(s): davide20

- Requested CPUs: 16

- Start time: 2017-12-05 17:33:47; End time: 2017-12-05 17:40:48

(Negative values indicate problems in the job info collection - check back in half an hour)

<===============================================================>

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

- Mean CPU\_0 processors power (W): 78.9761904762

- Total CPU\_0 processors energy (J): 33248.9761905

- Mean CPU\_1 processors power (W): 118.023809524

- Total CPU\_1 processors energy (J): 49688.0238096

- Mean CPU\_0 memories power (W): 137.0

- Total CPU\_0 memories energy (J): 57677.0

- Mean CPU\_1 memories power (W): 137.023809524

- Total CPU\_1 memories energy (J): 57687.0238096

- Mean CPU\_0 VCS0s VR power (W): 65.2380952381

- Total CPU\_0 VCS0s VR energy (J): 27465.2380952

- Mean CPU\_1 VCS0s VR power (W): 62.6666666667

- Total CPU\_1 VCS0s VR energy (J): 26382.6666667

- Mean CPU\_0 VDD0s VR power (W): 13.5952380952

- Total CPU\_0 VDD0s VR energy (J): 5723.59523808

- Mean CPU\_1 VDD0s VR power (W): 55.3333333333

- Total CPU\_1 VDD0s VR energy (J): 23295.3333333

<--------------------------------------------------------------->

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

- Mean CPU\_0 processors power (W): 78.9761904762

- Total CPU\_0 processors energy (J): 33248.9761905

- Mean CPU\_1 processors power (W): 118.023809524

- Total CPU\_1 processors energy (J): 49688.0238096

- Mean CPU\_0 memories power (W): 137.0

- Total CPU\_0 memories energy (J): 57677.0

- Mean CPU\_1 memories power (W): 137.023809524

- Total CPU\_1 memories energy (J): 57687.0238096

- Mean CPU\_0 VCS0 VR power (W): 65.2380952381

- Total CPU\_0 VCS0 VR energy (J): 27465.2380952

- Mean CPU\_1 VCS0 VR power (W): 62.6666666667

- Total CPU\_1 VCS0 VR energy (J): 26382.6666667

- Mean CPU\_0 VDD0 VR power (W): 13.5952380952

- Total CPU\_0 VDD0 VR energy (J): 5723.59523808

- Mean CPU\_1 VDD0 VR power (W): 55.3333333333

- Total CPU\_1 VDD0 VR energy (J): 23295.3333333

## Performances and energy metrics of UEABS on PCP systems

This section will present results of UEABS on both GPU and KNL systems. This benchmark suite is made of two set of codes that covers each other’s. The former is used to be run on standard CPU and de latest have been ported to accelerators. The accelerated suite is described in D7.5[6] and the standard suite is described on the PRACE UEABS official webpage[7] and D7.4[8]. In these documents are also described test cases specific to this suite and where to find corresponding datasets.

Metrics exhibited systematically will be time to solution and energy to solution. This choice allows to measure the exact same computation. Indeed, some code features 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 a hypothetic production runs. Unfortunately, such a system is not available yet for energy, and this metrics will be shown as side metrics.

To be comparable between machines, the Cumulative (all nodes) Total energy (J) has been selected for the GPU machine. And the nodes.energy has been selected for the KNL prototype. Both measure full nodes consumption in Joules.

Each code will be presented along with a short description and the full set of metrics. The section ends with a recap chart with a line of metric picked up for its relevance.

### ALYA

Alya is a high performance computational mechanics code that can solve different coupled mechanics problems.

The code is parallelised with MPI and OpenMP. Two OpenMP strategies are available, without and with a colouring strategy to avoid ATOMICs during the assembly step. A CUDA version is also available for the different solvers.

#### Test case 1 metrics

Table 2 Alya test case 1 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (J)** |
| 1 | 51,81 | 99487,2 |
| 2 | 29,63 | 184923,6 |
| 4 | 16,12 | 172866,7 |
| 8 | 9,18 | 219305,8 |

#### Test case 2 metrics

Table 3 Alya test case 2 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (J)** |
| 4 | 97,68 | 831423,3 |
| 8 | 50,88 | 1356584,3 |
| 16 | 28,97 | 1591488,7 |
| 32 | 16,41 | 1553542,1 |

### Code\_Saturne

Code\_Saturne is a 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. PETSc has recently been linked to the code to offer alternatives to the internal solvers to compute the pressure and supports CUDA.

#### Test case 1 metrics

Table 4 Code Saturn test case 1 metrics on PCP-KNL

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (J)** | **Time/timestep** |
| 1 Node, 68 MPI tasks | 1421,715 | 369 400 | 54,11 |
| 2 Nodes, 136 MPI tasks | 894,453 | 469 600 | 28,98 |
| 4 Nodes, 272 MPI tasks | 596,74 | 607 000 | 15,01 |
| 8 Nodes, 544 MPI tasks | 442,332 | 889 900 | 8,02 |
| 16 Nodes, 1088 MPI tasks | 408,858 | 1 600 000 | 5,06 |

Table 5 Code Saturn test case 1 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (J)** |
| 1 Node, 16 MPI tasks | 479 | 396776,8266 |
| 2 Nodes, 32 MPI tasks | 118 | 182036,1518 |
| 4 Nodes, 64 MPI tasks | 153 | 490682,283 |

#### Test case 2 metrics

### CP2K

CP2K 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.

#### Test case 1 metrics

Table 6 CP2K test case 1 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (kJ)** |
| 1 | 5917 | 1417,4 |
| 2 | 3737 | 1631,3 |
| 4 | 1922 | 1596,2 |
| 8 | 794 | 1520,2 |
| 16 | 424 | 1603,6 |
| 32 | 231 | 1795,5 |
| 64 | 147 | 2343,4 |

Table 7 CP2K test case 1 metrics on DAVIDE

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (kJ)** | **Energy to solution minus GPU energy (kJ)** |
| 1 | 4686 | 3365,175 | 2825,306 |
| 2 | 2344 | 3351,042 | 2833,297 |
| 4 | 1194 | 3459,177 | 2926,947 |
| 8 | 612 | 3528,345 | 2978,465 |
| 16 | 323 | 3745,263 | 3166,731 |

#### Test case 2 metrics

Table 8 CP2K test case 2 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (kJ)** |
|  |  |  |
| 2 | 2963 | 1410,2 |
| 4 | 1210 | 1396 |
| 8 | 729 | 1531 |
| 16 | 383 | 1616 |
| 32 | 226 | 1857 |
| 64 | 139 | 2427 |

Table 9 CP2K test case 2 metrics on DAVIDE

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (kJ)** | **Energy to solution minus GPU energy (kJ)** |
| 1 | 24573 | 18302,468 | 15504,87 |
| 2 | 12502 | 18444,03 | 15684,86 |
| 4 | 6380 | 19118,29 | 16217,08 |
| 8 | 3295 | 19737,644 | 16777,44 |
| 16 | 1695 | 20378,766 | 17314,52 |

### GADGET

GADGET is a freely available code for cosmological N-body/SPH simulations.

It is written in C and uses an explicit communication model that is implemented with the standardized MPI communication interface.

#### Test case 1

Table 10 Gadget test case 1 metrics with 4 MPI task per node and 16 OpenMP thread per task

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (MJ)** |
| 4 | 2082.97 | 1.7 |
| 8 | 1332.86 | 2.2 |
| 16 | 965.82 | 3.1 |

Table 11 Gadget test case 1 metrics on 8 PCP-KNL nodes

|  |  |  |  |
| --- | --- | --- | --- |
| **MPI task/node** | **OpenMP threads/task** | **Time to solution (s)** | **Energy to solution (MJ)** |
| 4 | 16 | 1332.86 | 2.2 |
| 4 | 32 | 1514.17 | 2.6 |
| 32 | 4 | 897.9 | 1.7 |

### GPAW

GPAW 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. Support for offloading to accelerators using either CUDA or pyMIC, respectively.

#### Test case 1 metrics

Table 12 GPAW test case 1 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL 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 |

#### Test case 2 metrics

Table 13 GPAW test case 2 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL 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 |

### GROMACS

GROMACS 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).

#### Test case 1 metrics

Table 14 GROMACS test case 1 metrics on PCP-KNL

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Optional metric (ns/day)** | **Energy to solution (kJ)** |
| 1 | 672,316 | 16,065 | 232.8 |
| 2 | 403,7 | 26,748 | 261.2 |
| 4 | 278,13 | 38,832 | 287.1 |

Table 15 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 | 31,133 | 317,71 |
| 2 | 226,28 | 49,949 | 390,03 |
| 4 | 201,32 | 53,646 | 702,5 |
| 8 | 132,82 | 81,316 | 938,48 |

#### Test case 2 metrics

Table 17 GROMACS test case 2 metrics on PCP-KNL

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Optional metric (ns/day)** | **Energy to solution (kJ)** |
| 1 | 1166,932 | 1,481 | 529,7 |
| 4 | 353,336 | 4,891 | 533,9 |
| 8 | 183,348 | 9,426 | 603,5 |
| 16 | 121,896 | 14,177 | 817,4 |
| 32 | 77,334 | 22,347 | 1200 |
| 48 | 59,08 | 29,251 | 1700 |

Table 18 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 | 2,364 | 641,6 |
| 4 | 195,64 | 9,247 | 682,9 |
| 8 | 122,2 | 14,132 | 900,4 |
| 16 | 64,58 | 21,46 | 1264,1 |
| 32 | 44,84 | 38,542 | 1723 |
| 40 | 43,458 | 39,77 | 2186,5 |

Table 19 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 | 4,134 | 436,03 |
| 4 | 120,38 | 14,357 | 508,9 |
| 8 | 77,308 | 22,357 | 620,9 |
| 16 | 50,85 | 33,989 | 859,18 |
| 32 | 30,81 | 56,097 | 1180,04 |

### NAMD

NAMD 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.

#### Test case 1 metrics

Table 20 NAMD test case 1 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (kJ)** |
| 1 | 3955,177246 | 1300 |
| 2 | 2085,820312 | 1400 |
| 4 | 1181,529297 | 1500 |
| 8 | 695,572998 | 1600 |
| 16 | 464,854797 | 2300 |

Table 21 NAMD test case 1 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (kJ)** |
| 1 | 3616.5 | 3575.671 |
| 2 | 2609.08 | 4999.399 |
| 4 | 1503.56 | 5627.773 |
| 8 | 721.72 | 5407.021 |
| 16 | 470.97 | 7037.861 |

#### Test case 2 metrics

Table 22 NAMD test case 2 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (MJ)** |
| 16 | 11280,23438 | 48.2 |
| 32 | 6624,53125 | 72.0 |
| 64 | 5280,578125 | 91.9 |

Table 23 NAMD test case 2 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (kJ)** |
| 8 | 1846,995239 |  |
| 16 | 1078,346558 |  |
| 32 | 608,431824 | 20224,81 |
| 40 | 529,711365 | 22896,61 |

### NEMO

### 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.

#### Test case 1 metrics

Table 24 PFARM test case 1 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (HH:mm:ss)** | **Energy to solution (kJ)** |
| 1 | 0:28:22 | 420.5 |
| 2 | 0:15:00 | 432.5 |
| 4 | 0:09:15 | 504.1 |
| 8 | 0:11:35 | 1100 |
| 16 | 0:08:07 | 1400 |

Table 25 PFARM test case 1 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution (s)** | **Energy to solution (J)** |
| 1 | 441.4520 | 256966.011616 |
| 2 | 266.2920 | 315615.673125 |
| 4 | 199.4450 | 583133.088165 |
| 8 | 165.3610 | 922055.245831 |
| 16 | 167.6120 | 3073015.5563 |

### 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 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 consists of two kernels, the QUDA and the QPhix library. The library QUDA is based on CUDA and optimize for running on NVIDIA GPU

#### First implementation metrics

Table 26 QCD part 1 test case 1 metrics on PCP-KNL 68 OpenMP thread per node

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution** | **Energy to solution** |
| 1 | 151 sec | 48.7 kJ |
| 2 | 86.9 sec | 55.8 kJ |
| 4 | 52.7 sec | 66.8 kJ |
| 8 | 36.5 sec | 89.8 kJ |
| 16 | 27.8 sec | 124.4 kJ |
| 32 | 15.6 sec | 162.4 kJ |
| 64 | 11.7 sec | 268.1 kJ |

Table 27 QCD part 1 test case 1 metrics on PCP-KNL 68 MPI tasks per node

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution** | **Energy to solution** |
| 1 | 110.0 sec | 41.6 kJ |
| 2 | 62.7 sec | 47.6 kJ |
| 4 | 39 sec | 61.2 kJ |
| 8 | 29.3 sec | 87.7 kJ |
| 16 | 38.3 sec | 201.6 kJ |
| 32 | 61.0 sec | 569.1 kJ |
| 64 | 150.0 sec | 2600 kJ |

Table 28 QCD part 1 test case 1 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution** | **Energy to solution (J)** |
| 1 | 21.4 sec | 21604,417 |
| 2 | 14.8 sec | 28058,656 |
| 4 | 10.1 sec | 39536,779 |

#### Second implementation metrics

Table 29 QCD part 2 test case 1 metrics on PCP-KNL

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution** | **Optional metrics** | **Energy to solution** |
| 1 | 81.9 sec | 184.729 Gflops | 34.1 kJ |
| 2 | 56.1 sec | 269.705 Gflops | 39.9 kJ |
| 4 | 34.3 sec | 441.534 Gflops | 49.8 kJ |
| 8 | 24.6 sec | 614.466 Gflops | 65.8 kJ |
| 16 | 23.5 sec | 644.303 Gflops | 117.0 kJ |
| 32 | 16.1 sec | 937.755 Gflops | 171.2 kJ |
| 64 | 18.9 sec | 800.514 Gflops | 375.0 kJ |

Table 30 QCD part 2 test case 1 metrics on DAVIDE

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full Davide nodes** | **Time to solution** | **Optional metrics** | **Energy to solution** |
| 1 | 3.76 sec | 1533.13 Gflops | 14.901 kJ |
| 2 | 4.88 sec | 3005.07 Gflops | 19.813 kJ |
| 4 | 3.72 sec | 5409.18 Gflops | 26.466 kJ |
| 8 | 4.04 sec | 7248.57 Gflops | 43.078 kJ |
| 16 | 4.86 sec | 3490.27 Gflops | 88.145 kJ |
| 32 | 4.86 sec | 4570.13 Gflops | 288.513 kJ |

Table 31 QCD part 2 test case 2 metrics on PCP-KNL

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution** | **Optional metrics** | **Energy to solution** |
| 8 | 194.6 sec | 828.941 Gflops | 522.8 kJ |
| 16 | 126.8 sec | 1272.43 Gflops | 611.3 kJ |
| 32 | 78.2 sec | 2063.4 Gflops | 755.2 kJ |
| 64 | 57.2 sec | 2819.23 Gflops | 1100 kJ |

### Quantum Espresso

QUANTUM ESPRESSO is an integrated suite of computer codes for electronic-structure calculations and materials modelling, based on density-functional theory, plane waves, and pseudopotentials.

It is implemented using MPI and CUDA offloading to GPU.

#### Test case 1 metrics

Table 32 Quantum Espresso test case 1 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (MJ)** |
| 1 | 2062,0 | 0,6820 |
| 2 | 1442,0 | 0,6204 |
| 4 | 1063,0 | 0,6761 |
| 8 | 659,0 | 1,0240 |
| 16 | 728,0 | 1,4000 |

Table 33 Quantum Espresso test case 1 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide GPU nodes** | **Time to solution (s)** | **Energy to solution (MJ)** |
| 1 | 312 | 0,26699 |
| 2 | 248 | 0,379492 |
| 3 | 200 | 0,43258 |
| 4 | 197 | 0,591365 |

#### Test case 2 metrics

Table 34 Quantum Espresso test case 2 metrics on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (MJ)** |
| 10 | 5916 | 16,00 |
| 15 | 3549 | 14,90 |
| 20 | 3886 | 20,00 |
| 30 | 3539 | 29,20 |

Table 35 Quantum Espresso test case 2 metrics on DAVIDE

|  |  |  |
| --- | --- | --- |
| **Number of full Davide GPU nodes** | **Time to solution (s)** | **Energy to solution (MJ)** |
| 2 | 2337 | 3,92086 |
| 4 | 1511 | 4,842343 |
| 5 | 1470 | 5,835589 |
| 6 | 1324 | 6,126432 |
| 8 | 995 | 5,982444 |
| 10 | 1041 | 8,005337 |
| 20 | 1189 | 16,107562 |

### SHOC

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

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

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 completion (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.

#### Test case 1, GEMM

Table 36 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 |

#### Test case 2, FFT

Table 37 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 |

#### Test case 3, MaxFlops

Table 38 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 |

#### Test case 4, Triad

Table 39 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 |

#### Test case 5, MD5Hash

Table 40 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 GH/s | 70.7 |
| 1 node - 4 GPUs | 106 | 60.3 GH/s | 125 |

#### Full SHOC benchmark results

Table 41 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 |
| 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 |

### Specfem3D\_Globe

The software package SPECFEM3D\_Globe 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 case 1

Table 42 Specfem3D Globe metrics test case 1 on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Time to solution (s)** | **Energy to solution (kJ)** |
| 4 | 261 | 221.5 kJ |

#### Test case 2

Table 43 Specfem3D Globe metrics test case 2 on PCP-KNL

|  |  |  |
| --- | --- | --- |
| **Number of full PCP-KNL nodes** | **Testcase 2 time to solution** | **Testcase 2 energy to solution** |
| 5 | 352 | 363.5 kJ |
| 10 | 272 | 501.0 kJ |

### Wrap-up table



## 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[9] (High Order solver for Radar cross Section Evaluation) simulation software for 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[10] hybrid iterative-direct sparse system solver, which is based on domain decomposition principles.

### 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[11].

### 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 [11]. 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[12] or PaStiX[13]. However, sparse direct solvers are in general poorly scalable when it comes to solve very large linear system 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.

### 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) [14][15] 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 coarse grid mechanism that enables one to keep constant the number of iterations 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 nodes, 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).

### Numerical and performance results

For the numerical simulations reported below we have used the PRACE-PCP Intel “Manycore” Knights Landing (KNL) cluster presented in Section 2.2.

#### MaPHyS used in standalone mode

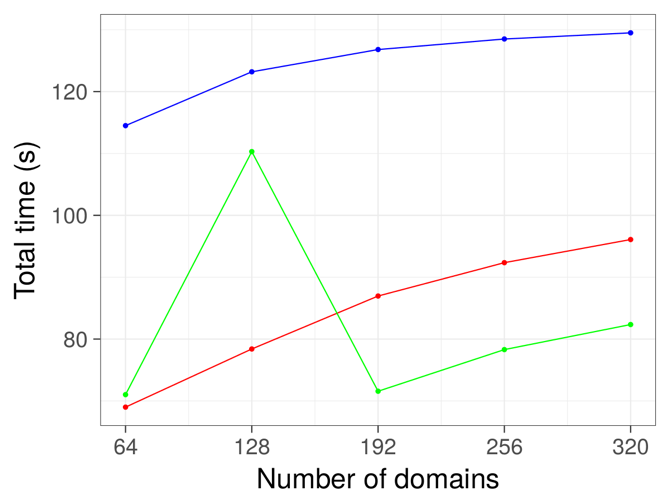
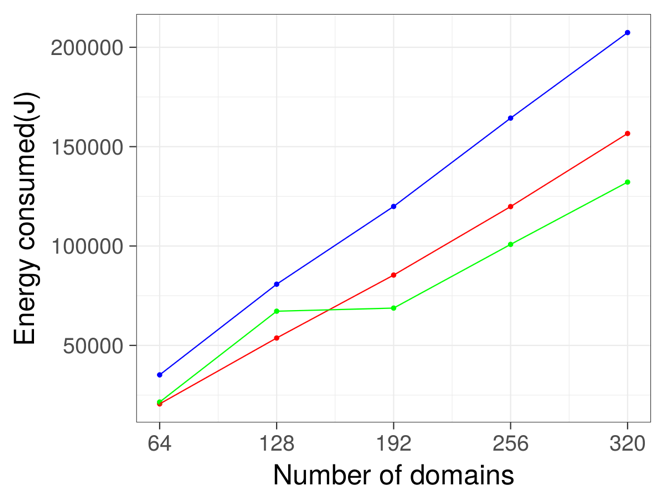
Weak scalability performance of the MaPHyS solver has been investigated in standalone mode. For these experiments, we solve a 3D Poisson problem on a 2.5D domain that corresponds to a beam and a 1D decomposition. Each subdomain has at most two neighbors and is essentially a regular cube of size 403 (i.e., 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: we consider the fully assembled local Schur complements to build the additive Schwarz preconditioner;
* sparse: 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 addition to the previously described dense preconditioner a coarse grid correction [16] 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, that 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.



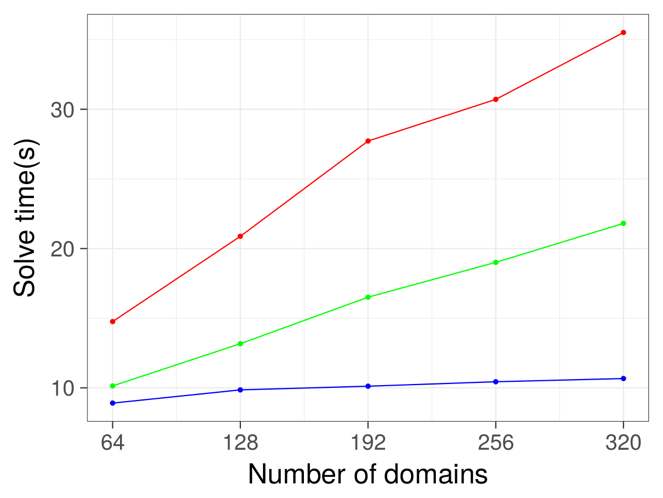
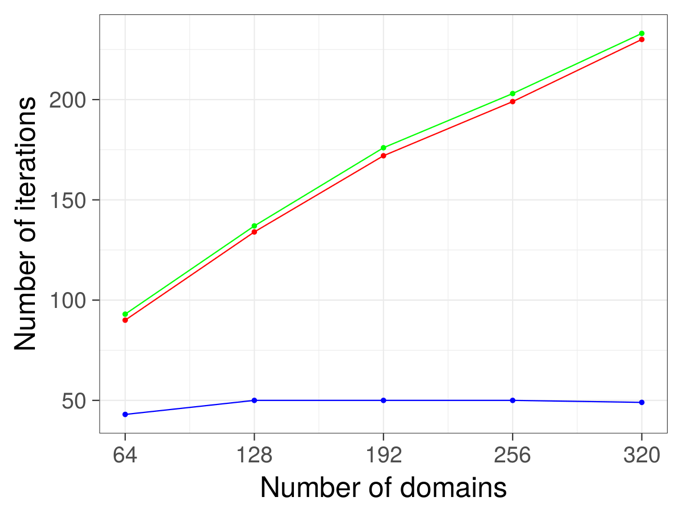


Figure 3 Weak scaling of MaPHyS from 1 to 5 nodes, with 64 subdomains per nodes and 1 core per subdomain



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

Table 44 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 |

#### 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 2. 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.

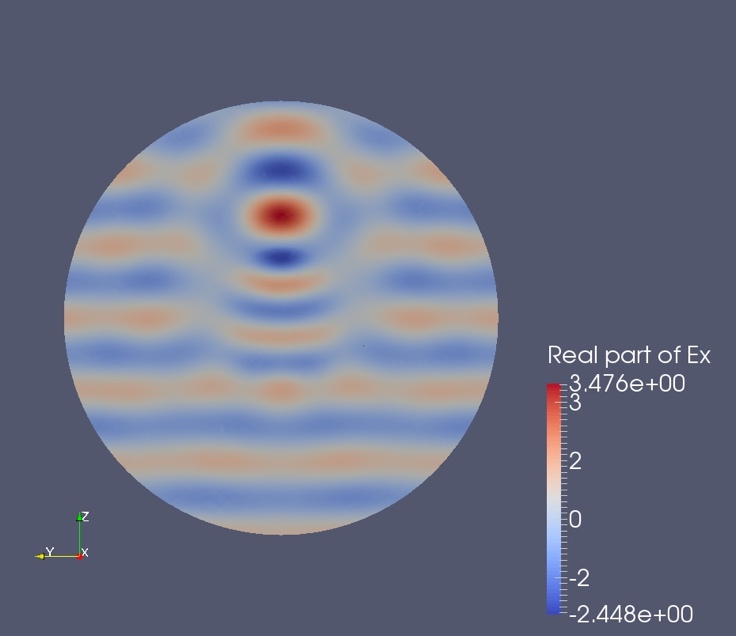


Figure 5 Scattering of a plane wave by a perfectly electric conducting sphere: contour lines of the x-component of the electric field (left) and RCS (right).

Table 45 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 |

## Conclusion

Work presented here stands as the first combined performances and energy results for UEABS. Also, it presents a detailed energy study has been conducted on KNL. It started to explore possibly available with new energy software and hardware.

Most of code of the PRACE benchmark suite have been run but there is still code, test cases, platform combinations missing. Such work suites well the PRACE WP7 5IP task on application performance that aims at merging standard and accelerated UEABS.

1. 1080W measured at blade power suply [↑](#footnote-ref-1)
2. Including Power8 and 4 Pascal GPU consumption only [↑](#footnote-ref-2)