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

D7.4
Evaluation of Tools and Techniques for Future Exascale Systems
Final

Version: 1.0
Author(s): Buket Benek Gursoy, ICHEC
Michael Browne, ICHEC
Michael Lysaght, ICHEC
Date: 15.02.2017

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: **D7.4**	
	Deliverable Nature: Report	
	Dissemination Level: PU / CO / CL*	**Contractual Date of Delivery:** 28 / 02 / 2017
		Actual Date of Delivery: 28 / 02 / 2017
	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: Evaluation of Tools and Techniques for Future Exascale Systems**	
	ID: **D7.4**	
	Version: 1.0	**Status:** ***Final***
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2010	
	File(s): D7.4.docx	
Authorship	**Written by:**	Buket Benek Gursoy, ICHEC Michael Browne, ICHEC Michael Lysaght, ICHEC
	Contributors:	S Muralidharan, ICHEC V Kannan, ICHEC J C Meyer, NTNU G Goumas, NTUA T Ponweiser, RISC Software/JKU O Robinson, ICHEC L Giraud, INRIA S Lanteri, INRIA O Selvitopi, Bilkent C Aykanat, Bilkent M Uchroński, WCSS J Donners, SURFsara D Sloan-Murphy, EPCC
	Reviewed by:	David Vicente, BSC Florian Berberich, JSC
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	01/02/2017	Draft	First Draft for PMO review
1.0	15/02/2017	Final version	PMO Review comments addressed

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, Exascale, Tools, Techniques, Centres of Excellence, FET-HPC projects

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[1] N. Mc Donnell, PRACE-4IP D7.3 Inventory of Exascale Tools and Techniques, pdf: http://www.prace-ri.eu/IMG/pdf/D7.3_4ip.pdf.

[2] M. Lysaght et al., PRACE-3IP D7.2.1 A Report on the Survey of HPC Tools and Techniques, pdf: http://www.prace-ri.eu/IMG/pdf/d7.2.1.pdf.

[3] To appear at http://www.prace-ri.eu/white-papers/.

[4] T. A. Davis and Y. Hu, The university of florida sparse matrix collection, ACM Transactions on Mathematical Software (TOMS), vol. 38, no. 1, p. 1, 2011.

[5] S. Balay, W. Gropp, L. C. McInnes and B. Smith, PETSc 2.0 User's Manual, Technical Report ANL-95/11, Argonne National Laboratory, November 1995.

[6] http://www.mcs.anl.gov/petsc/petsc.html.

[7] M. Heroux, R. Bartlett, V. H. R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring and A. Williams, An Overview of Trilinos, ACM Transactions on Mathematical Software, vol. 31, pp. 397- 423, 2005.

[8] http://www.kdm.wcss.wroc.pl/wiki/Nova.

[9] H. Hoteit and P. Ackerer, TRACES user's guide, v1.0, 2003.

[10] http://computation.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods.

[11] 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.

[12] 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.

[13] E. Agullo, L. Giraud and L. Poirel, Robust coarse spaces for abstract Schwarz preconditioners via generalized eigenproblems, INRIA Research Report No. RR-8978, December 2016, pdf: https://hal.inria.fr/hal-01399203v1.

[14] https://www.bsc.es/es/computer-applications/alya-system.

[15] 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.

[16] 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.

[17] R. O. Selvitopu and C. Aykanat, Reducing Synchronization Overheads in CG-type Parallel Iterative Solvers by Embedding Point-to-point Communications into Reduction Operation, PRACE Whitepaper, pdf: http://www.prace-ri.eu/IMG/pdf/WP187.pdf.

[18] http://www.nextgenio.eu/.

[19] http://www.archer.ac.uk/.

[20] https://www.dirac.ac.uk/.

[21] http://www.archer.ac.uk/documentation/rdf-guide/cluster.php.

[22] http://www.jasmin.ac.uk/.

List of Acronyms and Abbreviations

aisbl	Association International Sans But Lucratif (legal form of the PRACE-RI)
BioExcel	Centre of Excellence for Biomolecular Research
CoE	Center of Excellence
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture (NVIDIA)
EC	European Commission
E-CAM	An e-Infrastructure for software training and consultancy in simulation and modelling
EoCoE	Energy oriented Centre of Excellence for computer applications
ESiWACE	Excellence in Simulation of Weather and Climate in Europe
Extrae	Package devoted to generate Paraver trace-files for a post-mortem analysis
FET	Future and Emerging Technologies
FET-HPC	The Future and Emerging Technologies (FET) Proactive call for High Performance Computing
GB/s	Giga (= 109) Bytes (= 8 bits) per second, also GByte/s
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
GPFS	General Parallel File System
GPU	Graphic Processing Unit
HDF5	Hierarchical Data Format 5
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
HPCToolkit	An open-source suite of tools for profile-based performance analysis of applications developed at Argonne National Laboratory
MaX	Materials Design@eXascale
MB	Management Board (highest decision making body of the project)
MPI	Message Passing Interface
NetCDF	Network Common Data Form
NoMaD	The Novel Materials Discovery Laboratory
OpenCL	Open Computing Language
Open MP	Open Multi-Processing
POP	Performance Optimisation and Productivity
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
READEX	Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing
RI	Research Infrastructure
SEMA	System for Energy Measurement on Accelerators
TAU	Tuning and Analysis Utilities
TB	Technical Board (group of Work Package leaders)
TDP	Thermal Design Power
TF/s	Tera (= 1012) Floating-point operations (usually in 64-bit, i.e. DP) per second
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

VTune	Intel® VTune Amplifier provides is a tool to provide performance insight into CPU & GPU performance, threading performance & scalability, bandwidth
Xeon Phi	Intel accelerator technology

List of Project Partner Acronyms

BILKENT - UHEM	Bilkent University, Turkey
EPCC	EPCC at The University of Edinburgh, UK
ICHEC	Irish Centre for High-End Computing, Ireland
INRIA - GENCI	Institut National de Recherche en Informatique et Automatique, France
JKU	Johannes Kepler University of Linz, Austria
NTNU - SIGMA	The Norwegian University of Science and Technology, Norway
NTUA - GRNET	National Technical University of Athens, Greece
SURFsara	Dutch national high-performance computing and e-Science support center, part of the SURF cooperative, Netherlands
WCSS - PSNC	Wroclaw Centre of Networking and Supercomputing, Poland

Executive Summary

The objective of the PRACE-4IP Work Package 7 (WP7) 'Application Enabling and Support' is to provide enabling support for High Performance Computing (HPC) applications codes, to ensure that these applications can effectively exploit multi-petaflop systems and future PRACE Exascale systems. The Task 7.2a, within WP7, 'Preparing for Future PRACE Exascale Systems' aims to investigate the various programming tools, languages, libraries and algorithms needed for future Exascale systems through an analysis and exploitation phase.

In this deliverable, we report on the exploitation of state-of-the-art HPC tools and techniques on different codes that are of interest to the European scientific and engineering research community, and where possible, with a focus on European Centres of Excellence (CoEs) interests and requirements. In this sense, the report here follows on naturally from deliverable D7.3, 'Inventory of Exascale Tools and Techniques', which represented the first phase of activity in T7.2a. Much of the exploitation work reported on here, was inspired by the comprehensive and in-depth analysis of European CoE requirements and state-of-the-art HPC tools and techniques as reported in D7.3 (as well as D7.2.1 in PRACE-3IP).

We focus on five separate topics that we have identified as being important to enable European applications on the road to exascale, and which mirror four of the topics reported on in the survey of D7.3. These fours topics are: Programming Models, Scalable Libraries and Algorithms, Debuggers and Profilers and I/O Management Techniques. The fifth topic we have chosen to focus on is Energy Efficiency, which is becoming an increasingly relevant consideration on deep-petascale systems, and will become an even greater challenge to contend with on future European extreme-scale systems. In particular, we see the need for an increased focus on energy efficient computing within WP7 in PRACE-5IP in order to take full advantage of the solutions being delivered as part of the final phase of the PRACE Pre-commercial Procurement (PCP) and beyond.

While some of the tools and techiques we have exploited are well established and standardized, such as MPI and OpenMP, we have also driven effort into exploring newer, pathbreaking tools, such as the newly developed (and implemented) cache-aware roofline model, novel adaptive sparse solvers, high-resolution energy measurement systems and a prototype runtime-tuning tool for increased energy efficiency at exteme-scale, the latter emerging from one of the FET-HPC projects. We feel strongly that the engagement between PRACE and the FET-HPC projects (as well as other Exascale research and development projects) reflected by such WP7 projects should be further expanded and strengthened during PRACE-5IP.

Below, we provide a flavour of some of the investigative work carried out as part of the T7.2a exploitation phase. For a more detailed description of each of the projects summarised in this document, we refer the reader to the PRACE-4IP whitepapers associated with each of the 15 projects.

Energy Efficient Computing

Throughout the lifetime of PRACE, the energy consumption of the large-scale European HPC systems has continued to increase, which poses a significant challenge in meeting the power constraints placed on any future European Exascale system by the early 2020s time-frame. While there are many ongoing initiatives to confront the challenge of energy consumption on the hardware level, it is now increasingly appreciated that a more holistic approach is required that includes all layers of the "HPC stack", including the application layer. Several of the CoEs are keenly focused on improving the energy efficiency of their applications running on extreme-scale systems, including MaX, ESiWACE and PoP (as well as the ESCAPE FET-

HPC project) and it is with their interests in mind, that we investigate a prototype tool suite, known as READEX, being developed by the H2020 FET-HPC project (also named READEX). We find that a pre-alpha version of the READEX tool suite can already identify dynamism in highly scalable European applications that regularly use large-scale PRACE resources, which allows for the determination of their tuning potential for improving energy efficiency on such systems. We also apply the prototype tool to a proxy version of the well-known molecular dynamics code, NAMD, which is of interest to the ECAM and MaX CoEs.

Programming Interfaces and Standards

We note from the survey of CoE requirements carried during the first phase of T7.2a, that many of the CoEs have not yet shown evidence of exploring emerging or novel programming models, beyond the now well-established models, such as MPI, OpenMP, CUDA and OpenCL. As a result, we have focused mainly on exploiting these well known models on emerging many-core systems and, where appropriate have focused on investigating typically underutilized modern features of these standards (e.g., MPI 3.0). We see the lack of awareness of emerging and novel programming models, as well as a possible aversion to risk in utilizing these, as a worthwhile challenge for WP7 to confront more aggressively beyond PRACE 4IP. With a view to future exploitation of programming models (e.g. in PRACE 5IP), we feel that there is an important role for WP7 to play in carrying out exploratory work on more novel prototype-level programming models and tools, as well as to increase awareness of such programming models within the CoEs.

Debuggers and Profilers

Improving energy-to-solution and time-to-solution in European weather forecasting codes is a key objective of the ESiWACE CoE and even more so for the H2020 FET-HPC ESCAPE project. Such focus includes targeting the IFS weather code by a ‘divide-and-conquer’ approach in optimizing its most heavily used subroutines through the development of so-called “Weather Dwarfs”. This goal is predicated on a move to European Exascale computing, which is expected to exploit massively parallel node-level chip architectures, a “stepping-stone” example of which is the recently released Intel Xeon Phi Knights Landing (KNL) processor. With these challenges in mind, we investigate the powerful potential of the recently developed cache-aware roofline model (CARM) profiling technique to better understand where bottlenecks exist and improve performance for so-called “European Weather Dwarfs” on the Intel Xeon Phi platform.

Scalable Libraries and Algorithms

Several of the 23 application codes within the EoCoE CoE are concerned with the solution of large sparse linear systems, where the performance of such solvers greatly influences the overall scalability of the applications. We have focused on improving the scalability of two heavily used application codes within EoCoE (TRACES and HORSE), by investigating the potential of new algebraic hybrid iterative/direct solvers within these codes as an alternative to algebraic multigrid preconditioned Krylov iterative solvers and preconditioners, which are known to scale poorly at large scale. We also describe two applications for illustrating the modeling capabilities of the simulation software supported by EoCoE and assessing their parallel performance on the road to Exascale: the scattering of a plane wave by an aircraft, and the interaction of an electromagnetic wave with a heterogeneous model of head tissues.

I/O Management Techniques

Parallel I/O performance plays a key role in many high-performance computing (HPC) applications. I/O bottlenecks are an important challenge to understand and, where possible, eliminate on both current petascale systems and looking forward to exascale computing. It is therefore necessary for research communities with high I/O requirements to understand the

parallel I/O performance of existing HPC systems and applications to be suitably equipped to make informed plans for future procurements and software development projects. With this challenge in mind, in this deliverable we report on how we have worked closely with the ESiWACE CoE to better understand its I/O requirements and have focused on developing and running I/O benchmarks to provide deeper insight into performance across a range of large-scale European HPC systems. Given the ubiquity of I/O challenges across HPC domains, the findings should be of interest to the majority of the other CoEs.

1 Introduction

1.1 The Purpose of the document

The objective of the PRACE-4IP Work Package 7 (WP7) 'Application Enabling and Support' is to provide enabling support for High Performance Computing (HPC) applications codes, to ensure that these applications can effectively exploit multi-petaflop systems and future PRACE Exascale systems. The Task 7.2a, within WP7, 'Preparing for Future PRACE Exascale Systems' aims to investigate the various programming tools, languages, libraries and algorithms needed for future Exascale systems through an analysis and exploitation phase.

In this deliverable, we report on the exploitation of state-of-the-art HPC tools and techniques on different codes that are of interest to the European scientific and engineering research community, and where possible, with a focus on European Centres of Excellence (CoEs) interests and requirements. In this sense, the report here follows on naturally from deliverable D7.3, 'Inventory of Exascale Tools and Techniques', which represented the first phase of activity in T7.2a [1]. Much of the exploitation work reported on here, was inspired by the comprehensive and in-depth analysis of European CoE requirements and state-of-the-art HPC tools and techniques as reported in D7.3 (as well as D7.2.1 in PRACE 3IP [2]) [1].

We focus on five separate topics that we have identified as being important to enable European applications on the road to exascale, and which mirror four of the topics reported on in the survey of D7.3 [1]. These fours topics are: Programming Models, Scalable Libraries and Algorithms, Debuggers and Profilers and I/O Management Techniques. The fifth topic we have chosen to focus on is Energy Efficiency, which is becoming an increasingly relevant consideration on deep-petascale systems, and will become an even greater challenge to contend with on future European extreme-scale systems. In particular, we see the need for an increased focus on energy efficient computing within WP7 in PRACE 5IP in order to take full advantage of the solutions being delivered as part of the final phase of the PRACE Pre-commercial Procurement (PCP) and beyond.

1.2 Structure of the Document

The document presents five subsections: Energy Efficiency, Programming Interfaces and Standards, Debuggers and Profilers, Scalable Libraries and Algorithms and I/O Management Techniques. Within each section, a short introduction is provided which further details the structure of the individual section, which is then followed by a collection of reports summarising the work and findings of each of the T7.2a projects during the exploitation phase.

1.3 Organization of Work

PRACE-4IP T7.2a 'HPC Tools & Techniques' fully started in May 2015 with an initial Telco and then a subsequent Face-to-Face meeting in Amsterdam in June 2015. The objective was

to discuss how to develop links with the CoEs. The outcomes of the meeting included a list of people to contact on a technical level to understand more of the requirements for the CoEs. The team drafted a questionnaire to be submitted to each CoEs through a designated Point of Contact (PoC). The questionnaire was completed by mid September 2015. The PoCs were identified by late November and the CoEs had received the questionnaire by early December 2015. The completed questionnaires were returned in January 2016 and the team analysed the results. The findings were documented in deliverable D7.3, 'Inventory of Exascale Tools and Techniques' that was submitted to the EC (European Commission) in March 2016 [1]. The participating CoEs were:

- BioExcel, Centre of Excellence for Biomolecular Research;
- CoeGSS, Center of Excellence for Global Systems Science;
- E-CAM, An e-Infrastructure for software training and consultancy in simulation and modelling;
- EoCoE, Energy oriented Centre of Excellence for computer applications;
- ESiWACE, Excellence in Simulation of Weather and Climate in Europe;
- MaX Materials, Design@eXascale;
- NoMaD, The Novel Materials Discovery Laboratory; and
- POP, Performance Optimisation and Productivity.

The subsequent exploitation phase within T7.2a was inspired by D7.3. During the PRACE-4IP WP7 Face-to-Face Meeting in Norway in June 2016 the targeting of appropriate state of the art HPC tools and techniques along with CoE-relevant applications [where possible] was discussed between partners and where each of the exploitation projects were defined. In Q3 and Q4 of 2016 the bulk of the work on these projects (15 in total) was completed at the technical level and some were increasingly aligned with FET-HPC projects and CoEs to better share knowledge and expertise. Most of the technical work was completed early December resulting first drafts of whitepapers which present the work of the exploitation phase. At the PRACE-4IP WP7 Face-to-Face Meeting in INRIA in December 2016 the partners presented their progress reports and conclusions in the whitepapers. 7 of these whitepapers were completed and submitted for internal T7.2a and PRACE-4IP review in early February 2017. The rest of the whitepapers will be submitted for review in early March 2017.

1.4 Intended Audience

Our objective in preparing this report is to exploit the most promising HPC tools and techniques that may have applicability for preparing applications for near-term European deep petascale and future Exascale systems. Targeted primarily at the European HPC community, including the European CoEs, it provides an overview of how a selection of state-of-the-art HPC tools and techniques fared when enabling real applications targeting petascale systems/future exascale systems. We also hope that the report here will be of interest to the European scientific computing community more generally.

2 Energy Efficient Computing

In this section, we report on two projects that have each focused on exploiting state-of-the- art energy-focused tools, in order to enable applications for multi-petaflop/future exascale systems. Each subsection provides a summary of the project along with a reference to the PRACE-4IP whitepaper associated with the project, as well as the CoEs that we expect will find the work of interest. We recommend that the reader also refers to the associated whitepaper for each project, which provides a more detailed report on the projects than is

provided here. The list of energy-focused tools, applications targeted, as well as CoEs focused on can be seen in Table 1.

HPC Tool/Technique	Application	CoEs/Scientific Communities
SEMA	ViennaCL	MaX, POP
READEX	PERMON, ESPRESO, BLAS, AMG2013, miniMD	ECAM, MaX, EoCoE, NoMaD, POP

Table 1: Tools/Techniques exploited along with corresponding applications with in the Energy Efficient Computing

2.1 Investigating and Exploiting Application Dynamism For Energy-Efficient Exascale Computing

WP242: Investigating and Exploiting Application Dynamism For Energy-Efficient Exascale Computing

Authors: Venkatesh Kannan (ICHEC), Lubomír Říha (VŠB-TUO), Michael Gerndt (TUM), Anamika Chowdhury (TUM), Ondrej Vysocky (VŠB-TUO), Martin Beseda (VŠB-TUO), Horák David (VŠB-TUO), Radim Sojka (VŠB-TUO), Jakub Kruzik (VŠB-TUO), Michael Lysaght (ICHEC)

HPC Tool/Technique: READEX

Application: PERMON, ESPRESO, BLAS, AMG2013, miniMD

CoEs/Scientific Communities: Energy efficient Exascale computing

READEX is a EU Horizon 2020 FET-HPC project whose objective is to exploit the dynamism found in high-performance computing applications at runtime to achieve efficient computation on Exascale systems. HPC is a major driving force for European research and innovation in many scientific and industrial domains. The applications in these areas are highly complex and demand high performance. As a result of this growing need for computational performance, the energy consumption of the HPC systems has continued to increase. Dynamic resource requirements in an application presents opportunity to tailor the utilisation of resources in the HPC system based on their requirements by the application at runtime. The READEX (Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing) project leverages this approach to deliver improved performance and energy-efficiency when executing applications on current and future extreme-scale systems. The goal of the READEX project is to create a tool suite that:

- Identifies existing dynamism in an application to determine its tuning potential.
- Determines the configurations for different hardware-, system software- and application-level tuning parameters that suit different scenarios that may arise during the application's execution.
- Applies the best configuration for the tuning parameters during the application's runtime.

This work investigates a pre-alpha prototype of the READEX tool suite to identify existing dynamism in proxy applications as well as highly scalable European applications that regularly use large-scale PRACE resources to determine their tuning potential. We also apply the prototype tool to a proxy version of the well-known molecular dynamics code, NAMD, which is of interest to the ECAM and MaX CoEs.

When analysing a given application, the READEX tool suite quantifies the application dynamism using dynamism metrics such as compute intensity and execution time of the application. The objective of the tool suite is to switch values for the different hardware-,

system- and application-level tuning parameters to optimal configurations at runtime based on the observed application dynamism. We also present the preliminary results from our tuning experiments to demonstrate the potential savings that may be achieved using the READEX approach.

Firstly, we demonstrate the identification of application dynamism by the READEX tool suite using the miniMD application from the Mantevo benchmark suite, which is a light-weight version of NAMD, a widely-used molecular dynamics application in the European HPC community. An illustration of the dynamism in execution time and computational intensity observed in miniMD is illustrated in Figure 1. Such dynamism present in a given application is identified by the readex-dyn-detect tool that is developed as a part of the READEX tool suite.

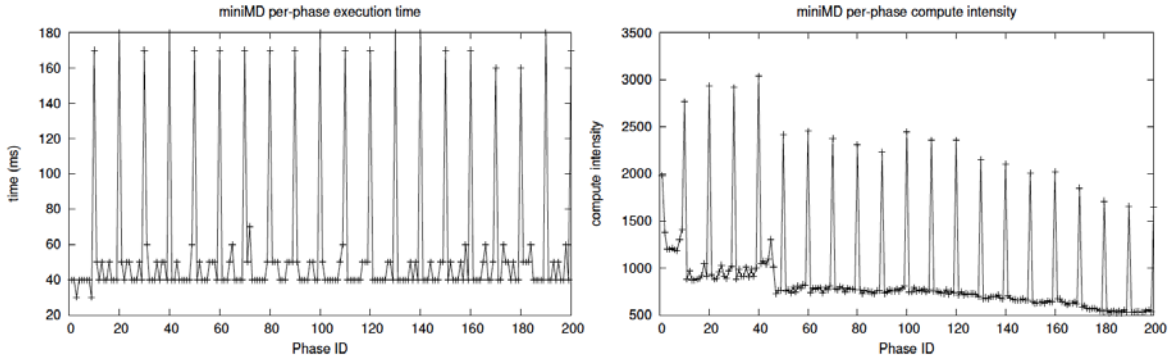

Figure 1: Inter-phase dynamism observed in miniMD

Secondly, we present the potential execution time and energy savings that are achieved from tuning actions that are performed by a tool called MERIC which is developed to perform the tuning experiments and measure the savings for different configurations. For this, we use total FETI solvers (PERMON and ESPRESO), BLAS routines and AMG2013 proxy applications. Table 2 presents a sample of the overall energy savings measured by MERIC.

Application	Static savings [%]	Dynam. savings [%]	Total Savings [%]
PERMON TFETI	11.84	2.68	14.52
ESPRESO TFETI	7.24	5.44	12.68
BLAS ROUTINES	5-23		5-23
ProxyApps 1 - AMG2013	11.42	1.43	12.85

Table 2: Static and dynamic energy savings measured by the MERIC Tool

For PERMON the number of processor cores used is statically tuned through the number of MPI processes per node and the processor core frequency is dynamically tuned. For ESPRESO the processor core frequency and number of OpenMP threads are both dynamically tuned. All these tests were performed on a single socket of a dual socket compute node. Consequently, this increases the baseline consumption of energy consumption. Thus, the dynamism can be potentially higher if both sockets are used. In future, these tests will be performed to perform the measurements on both sockets. In the case of the Algebraic Multigrid (AMG2013) solver ProxyApp we were able to achieve static/dynamic savings 11.42/1.43% for AMG2013 by setting the optimal core frequency and statically tuning the number of MPI processes.

Finally, the measurements collected by the application dynamism analysis experiments performed by the READEX tool suite are logged into a READEX Applicaton Dynamism Analysis Report (RADAR). The RADAR presents the values set for the tuning parameters in

the experiments and measurements of the dynamism metrics and the objective values (execution time and energy consumed).

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP242: Investigating and Exploiting Application Dynamism For Energy-Efficient Exascale Computing [3].

2.2 A System for Energy Measurement on Accelerators (SEMA)

WP240: A System for Energy Measurement on Accelerators (SEMA)

Authors: Servesh Muralidharan (ICHEC), Michael Lysaght (ICHEC)

HPC Tool/Technique: SEMA

Application: ViennaCL

CoEs/Scientific Communities: MaX, POP, ESiWACE

The increasing dominance of accelerators for energy efficient computing in current and future extreme scale European HPC systems demands a deeper understanding of the relationship between power consumption, performance and application code in order to exploit many-core platforms in the most energy efficient manner. The measurement of power consumption is supported at different levels of accuracy by different vendors through their platforms. However, such proprietary methods lack any standardized metrics, making it difficult to make reliable comparisons of power and energy consumption on many-core accelerators. Furthermore, the typical accuracy such methods support is often at a coarse level making it difficult to obtain and sufficiently accurage energy profile of a given application running on such systems.

This work describes the development of an open low-cost System for Energy Measurement on Accelerators (SEMA), which allows the measurement of energy consumption on accelerators with a PCIe form factor to a milli-watt level accuracy at milli-second time resolution. SEMA works based on the standard current shunt based power measurement techniques. To meet the goal of easy use, we have come up with set of novel ideas that abstract the technical details and provide the users with a very simple interface to measure energy and power. This interface can be used to profile very short regions of code within the application and allows for the extraction of insights into the performance and power consumption of different segments of code at a level not possible before, thereby leading to improvements in understanding code behavior and optimizations. These can also feed back to the design of better energy efficient accelerator architectures in the future.

We have setup a few initial experiments to evaluate our system. This mainly consists of a few native codes running on the two accelerators and a benchmark that can run on either of these devices. In this evaluation our objective is not to optimize the code for a particular device but more to present results that can be used as a starting point for further analysis. The two accelerators chosen for our system are the NVIDIA Tesla K40 and Intel Xeon Phi 7120. The Tesla K40 comes with 6GB RAM and 2880 threading cores. It can run at a maximum frequency of 875 MHz and is designed for a TDP of 235 Watts. Its capable of ~4 TF/s in single precision, ~1.5 TF/s in double precision operations and has a maximum memory bandwidth of ~288 GB/s. The Intel Xeon Phi 7120 has 61 cores and 16GB of memory and can run at a maximum frequency of 1.33 GHz in turbo mode with a designed TDP of 300 Watts. Its capable of ~2.4 TF/s in single precision, ~1.2 TF/s in double precision operations and ~352 GB/s of peak memory bandwidth. These accelerators are used heavily in large supercomputing clusters part of the PRACE group to achieve several Petaflops of performance. Optimizations on performance and energy efficiency improvements of these

devices could greatly impact European scientific communities to help effectively use these resources.

The initial results consist of measurements on the two devices with different benchmarks. The X-axis consists of different types of measurement performed on the same program marked by ‘xN’ where ‘N’ is the factor multiplied with the value in Y-axis. Each item in the X-axis also contains variations such as the type of precision used or the kind of operation performed. We have two metrics for performance derived from execution time of the code within the benchmark and from the counter within SEMA interface. We find that there is a slight overhead in using the SEMA library in comparison to directly measuring it from within the code. We can also see the energy efficiency result represented by GFlops/J and the peak power represented in ‘W’ or Watts.

In Figure 2 we present results from optimized version of ViennaCL for GEMM operation on the Tesla K40. It compares different GEMM operations. The prefix correlates to the precision used ‘s’ for single and ‘d’ for double. The suffix corresponds to whether the matrix was transposed before the GEMM operation. ‘NN’ corresponds to normal GEMM, ‘NT’ corresponds to the second matrix being transposed, ‘TN’ corresponds to first matrix being transposed and ‘TT’ corresponds to both matrix being transposed. Two interesting results comparing performance and energy efficiency can be made. While the “sGEMM-NT” and “sGEMM-TT” provides similar performance, it can be observed that “sGEMM-NT” is in fact slightly more energy efficiency in comparison to “sGEMM-TT”. In the case of double precision, the “dGEMM-TN” and “dGEMM-TT” are similar in energy efficiency the performance between them significantly differs.

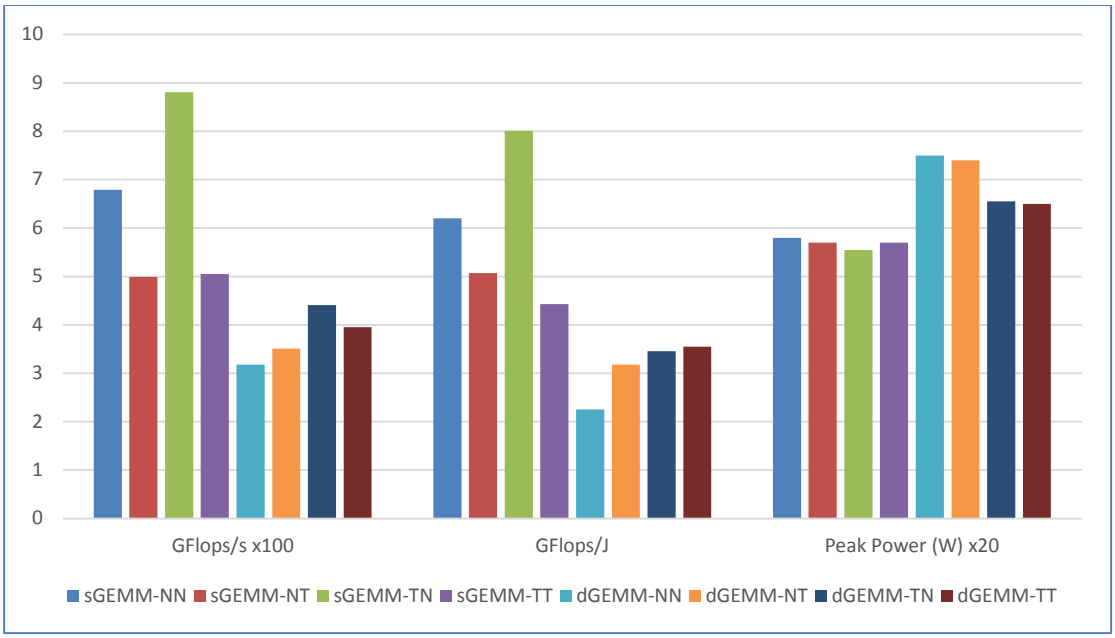

Figure 2: ViennaCL - 1.7.1 - Optimized for Tesla K40

For more details on this project, including a detailed description of the SEMA architecture, methodology and results, we refer the reader to PRACE Whitepaper WP240: A System for Energy Measurement on Accelerators (SEMA) [3].

3 Programming Interfaces and Standards

In this section, we report on seven projects that have each focused on exploiting state-of-the-art programming interfaces and standards (hereafter referred to as programming models), in order to enable applications for European multi-petaflop/future exascale systems. Each subsection provides a summary of the project along with a reference to the PRACE-4IP whitepaper associated with the project, as well as the CoEs that we expect will find the work of interest. We recommend that the reader also refers to the associated whitepaper for each project, which provides a more detailed report on the projects than is provided here. The list of programming models, applications targeted, as well as CoEs focused on, can be seen in Table 3.

HPC Tool/Technique	Application	Exascale/CoEs/Scientific Communities
OpenCL, OpenMP	Kinetic Ising model	Neuroscience community
CUDA, TAU	Smoothed Particle Hydrodynamics (SPH) method	SPHERIC, SPHysics, CNRS, CNR-INSEAN and MARSTRUCT Network of Excellence
MPI, OpenMP, Intel VTune	Bifrost	Theoretical Astrophysics community
OpenMP	CoMD	Computer architecture and molecular dynamics research communities
OpenMP	Sparse matrix-vector multiplication (SpMV)	MAX, BioExcel, ESiWACE, EoCoE
CUDA	Configuration interaction method	MaX, NoMaD
MPI 3, GABRIEL	LES-COAST, NEMO	ESiWACE

Table 3: Programming Interfaces and Standards exploited along with corresponding applications

3.1 Performance portability of OpenCL with application to Neural Networks

WP231: Performance portability of OpenCL with application to Neural Networks

Authors: Jan Christian Meyer (NTNU), Benjamin Adric Dunn (NTNU)

HPC Tool/Technique: OpenCL, OpenMP

Application: Kinetic Ising model

CoEs/Scientific Communities: Neuroscience community

This work investigates the application of the OpenCL programming model to an application prototype which models maximum likelihood estimation of a hidden neural network within the framework of the kinetic Ising model. The enabling tools are the OpenCL programming model as implemented in the NVIDIA and Intel compiler software suites, as well as the OpenMP programming model. On the path toward applying the method to exascale problems, a key component of exploiting the inherent parallelism in the computation is to develop methods to evaluate candidate node architectures in terms of their application-specific performance. The OpenCL programming model is suitable for this purpose, as it offers portability across a range of accelerator architectures, thereby reducing the need to produce highly customized tests for each candidate system. This work presents experiments from three

different heterogeneous node architectures with variable fitness for the task, and evaluate their applicability.

Work is focused on a computation which infers the structure of an underlying, hidden neural network based on a time series of states recorded from an observable surface. The method is composed of two distinct stages; an iteration to convergence which can be fully hosted within accelerator memory, and a less frequent global update of the system state. The former is almost fully parallelizable as a set of independent, element-wise matrix operations, making it highly amenable to acceleration on throughput-oriented architectures, such as GPUs. The resulting computational kernels are, however, limited in numerical intensity, which limits the attainable speedup over comparable multi-core processors with fewer parallel units, but more elaborate memory hierarchies. Three generations of hybrid node architectures are tested, as summarized in Table 4.

Type	CPU	CPU core/socket	Accelerator
A	Intel E5-4627	8	NVIDIA K6000
B	Intel i7-3770	8	AMD Radeon HD 7979
C	Intel i7-4930K	6	AMD Radeon HD 4650

Table 4: Test Architectures

Comparisons between the OpenCL and OpenMP versions of the computation reflect a development in the memory systems of the tested GPUs. The older system (type C) shows that the OpenMP version vastly outperforms the OpenCL version regardless of the problem instance dimensions (N,M), as shown in Figure 3.

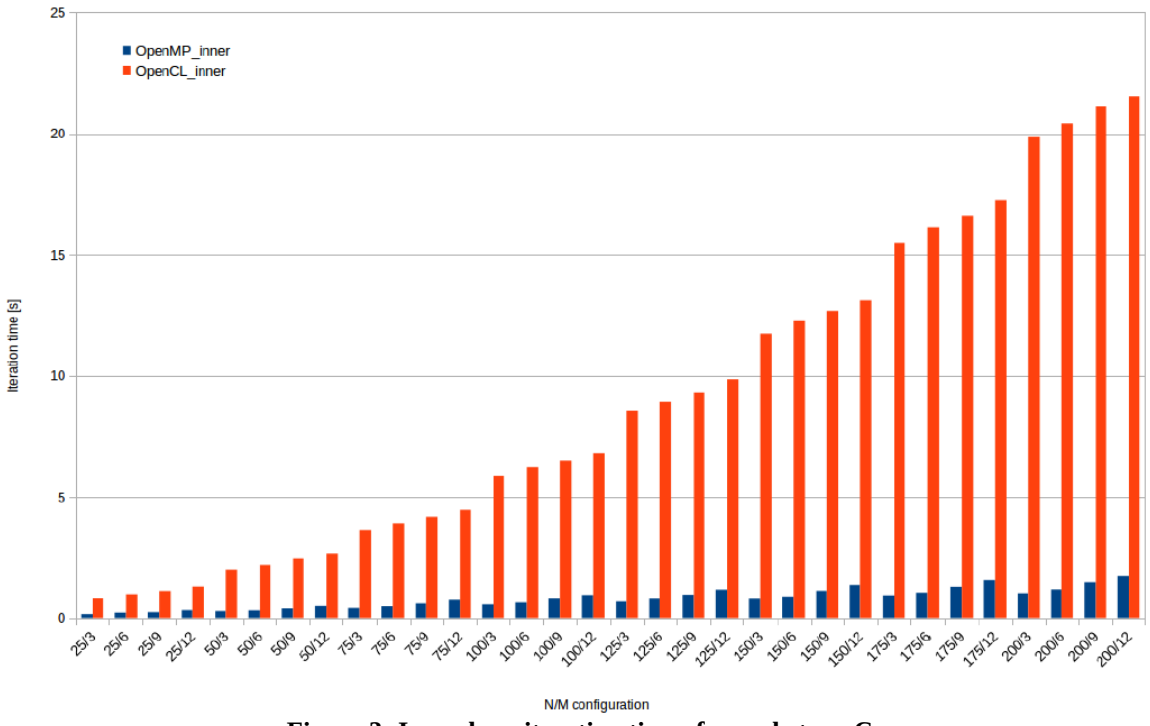

Figure 3: Inner loop iteration times for node type C

For the newer systems, we observe that the OpenCL implementation can provide superior performance for particular problem instance dimensions, but the advantage is small, and occurs at irregular points in the spectrum of configurations, demonstrated most clearly in the measurements from node type A. See Figure 4.

The results indicate that the accelerator architectures in this study are not generally suited for upscaling the application, but that the performance portability of the OpenCL implementation allows it to function as a tool to identify emerging architectures, which may prove to become

better candidates for exascale system components if the trend we observe in their architectural development continues.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP231: Performance portability of OpenCL with application to Neural Networks [3].

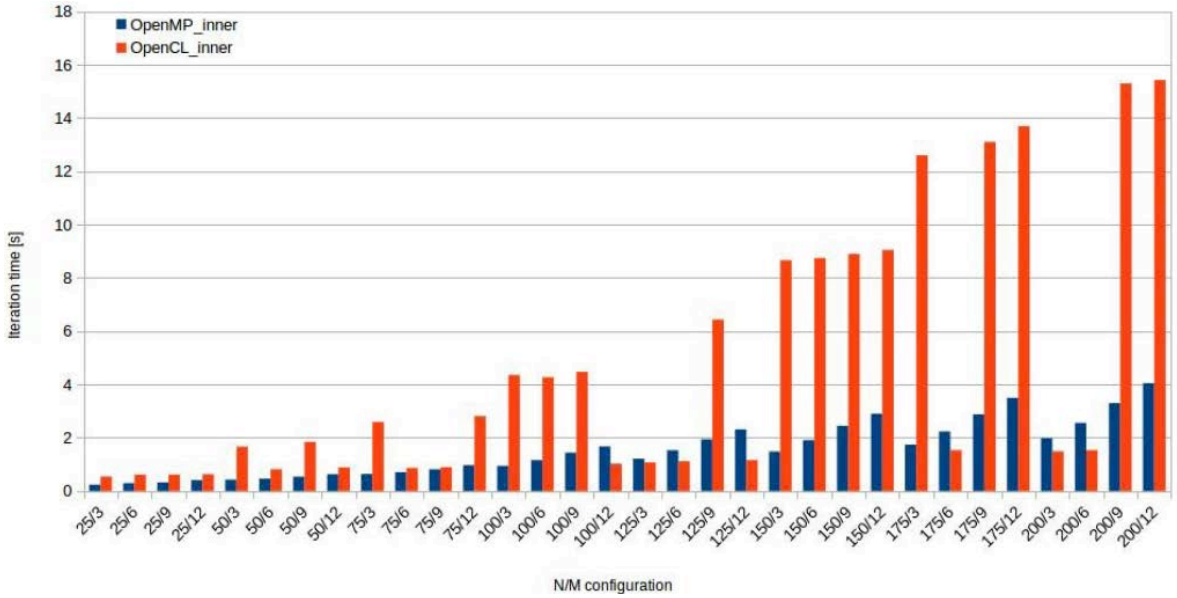

Figure 4: Inner loop iteration times for node type A

3.2 GPU Simulations of Violent Flows with Smooth Particle Hydrodynamics (SPH) Method

WP232: GPU Simulations of Violent Flows with Smooth Particle Hydrodynamics (SPH) Method

Authors: Tufan Arslan (NTNU), Murat Özbulut (Sabanci)

HPC Tool/Technique: CUDA, TAU

Application: Smoothed Particle Hydrodynamics (SPH) method

CoEs/Scientific Communities: SPHERIC, SPHysics, CNRS, CNR-INSEAN and MARSTRUCT Network of Excellence

This work investigates the potential for using GPUs to simulate a violent free-surface flow problem by a Smoothed Particle Hydrodynamics (SPH) method. The enabling tools are the state-of-the-art open source TAU profiling software, and the PGI CUDA Fortran compiler. The simulated problem models sway sloshing in a tank, which is computationally demanding due to the large number of particles required. The problem is relevant to applications in marine, oil/gas, and aeronautics, due to the effects of sloshing in gas, fuel and water tanks in ships and aircraft.

The results from this work may be interesting for CoEs and research communities such as SPHERIC, SPHysics, CNRS, CNR-INSEAN and MARSTRUCT Network of Excellence. SPH methods make promising candidates for GPU computation, as the explicit approach of encoding physical properties in independent particles is highly parallel, and increasing their number admits improvements in either the size of the simulated system, or its resolution. As GPU architectures contribute significantly to the peak performance of several petascale systems, investigating parallelization techniques to efficiently exploit them is an important step towards exascale simulations.

TAU is used to identify particle neighborhood and weight functions at computational hotspots, accounting for 70% of the computation time. In isolated testing with a 3000 particle simulation, the speed of the CUDA accelerated particle distance subroutine improves 15x over a CPU implementation, and the weight function computation exhibits a 6x speedup. The net effect of these improvements on total simulation time is a speedup factor of 2.2x. A summary of these figures is shown in Figure 5.

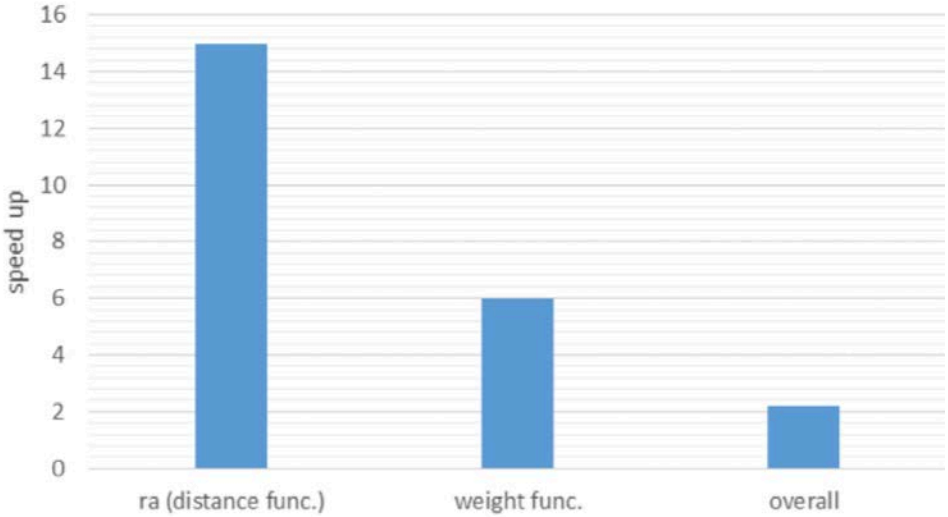

Figure 5: Performance over CPU

A comparison of simulation results with experimental and numerically obtained results presented in Figure 6 is showing that the period and amplitude of the simulated harmonic fluid motion are in agreement with results from the literature.

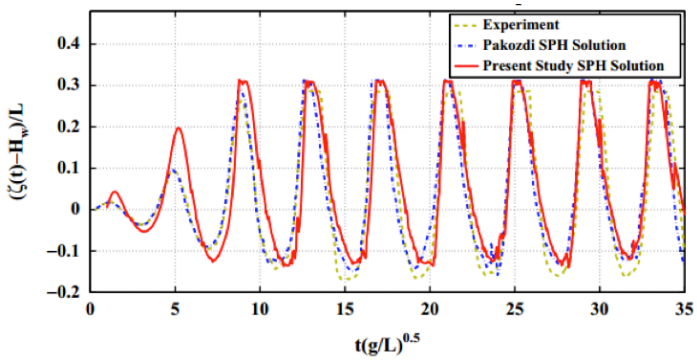

Figure 6: Comparison of results

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP232: GPU Simulations of Violent Flows with Smooth Particle Hydrodynamics (SPH) Method [3].

3.3 Stellar Atmosphere Simulation code Bifrost on Intel Xeon Phi Knights Landing

WP233: Stellar Atmosphere Simulation code Bifrost on Intel Xeon Phi Knights Landing

Authors: Mikolaj Szydlarski (UIO), Vegard Eide (NTNU)

HPC Tool/Technique: Intel VTune, MPI, OpenMP

Application: Bifrost

CoEs/Scientific Communities: Theoretical Astrophysics community

This work investigates the effectiveness of porting the stellar atmospheric simulation software Bifrost to the Intel Knight's Landing (KNL) architecture. The enabling tools are the Intel compiler software suite and state-of-the-art VTune profiling tool, as well as the MPI programming model.

Bifrost is a solver program for the 3D radiation magnetohydrodynamic equations on a staggered grid, using a high order, compact finite difference scheme. The application is developed at the Institute of Theoretical Astrophysics at the University of Oslo, and is of direct interest to the astrophysics community. Beyond this, finite difference methods combined with domain decomposition create computational and communication requirements which are similar in any application of related numerical approaches, specifically, numerical kernels with dense data access patterns and neighbor-node border exchange communication in a regular mesh. The scalability characteristics of this class of applications is well known in general, and Bifrost in particular has already proven capable of utilizing PRACE Tier-0 resources. This makes it pertinent to examine the efficiency of its adaptation to candidate architectures for future exascale systems, such as KNL.

Investigations are made with two sample problems of 1923 and 3843 element sizes, in order to admit test cases which exhibit physics seen in production runs. KNL Vectorization (SIMD execution) features are identified as a key performance parameter: approx. 60% of CPU time is spent in vectorized code independent of the problem size, and this results in a 1.2x speedup on a Haswell multicore node, compared to 3x on KNL. KNL is equipped with a programmable high-speed local memory, which is sufficient to contain the entire 1923 problem case. The larger case requires selective allocation and use of this memory for specific data. Utilizing this memory results in speedup factors of 3.1x and 3.2x in comparison to a baseline where local memory is used as an automatic data cache.

It is also found that a pure MPI parallelization performs better than a hybrid MPI/OpenMP scheme, as computations are highly spatially localized in problem subdomains, leading thread parallelism to cause contention for physical resources. The high memory bandwidth and vectorization features of KNL make it an attractive architecture for Bifrost, displaying overall speedup factors of approx. 2x over a comparable Haswell compute node, as shown in Figure 7.

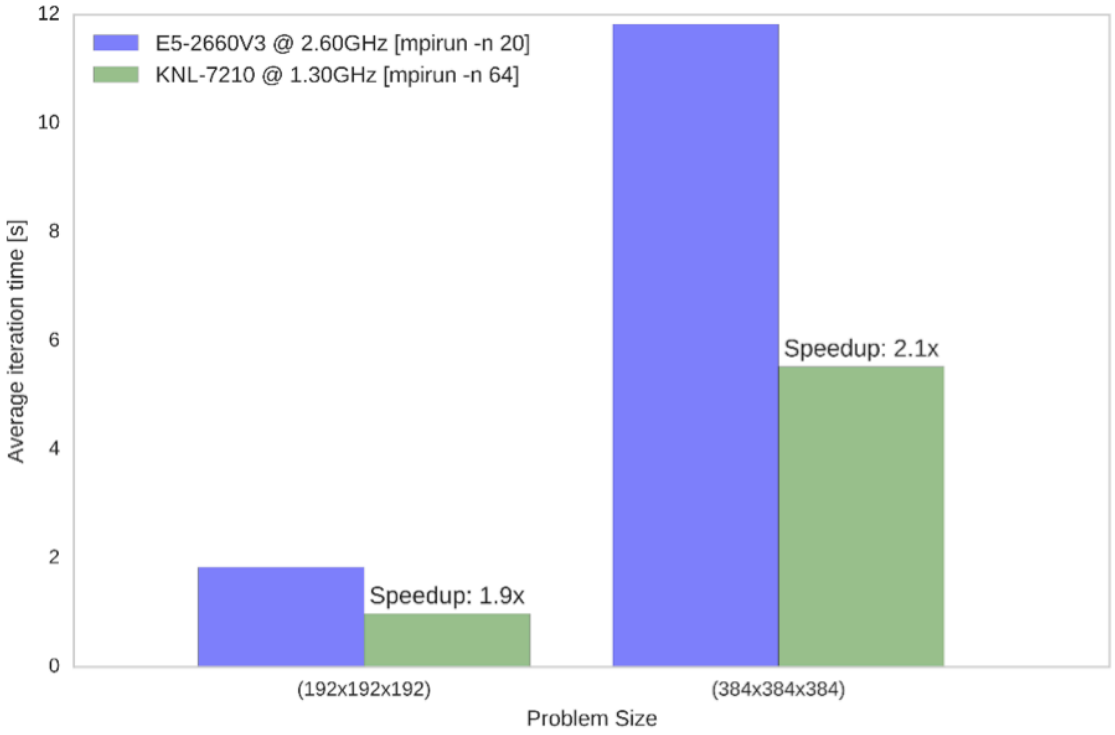

Figure 7: Performance comparison between E5-2660V3 and MIC-KNL

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP233: Stellar Atmosphere Simulation code Bifrost on Intel Xeon Phi Knights Landing [3].

3.4 Study of Xeon Phi Performance of a Molecular Dynamics Proxy Application

WP234: Study of Xeon Phi Performance of a Molecular Dynamics Proxy Application

Authors: Benjamin Andreassen Bjørnseth (NTNU), Jan Christian Meyer (NTNU), Lasse Natvig (NTNU)

HPC Tool/Technique: OpenMP

Application: CoMD

CoEs/Scientific Communities: Computer architecture and molecular dynamics research communities, including E-CAM and MaX

This work investigates the effectiveness of a range of optimization techniques when applied to the molecular dynamics proxy application CoMD on Intel Xeon Phi accelerator units. The enabling tools are the OpenMP programming model, and the vectorization features of the Intel compiler software suite.

CoMD is a molecular dynamics proxy application, which models the evaluation of interatomic potentials and corresponding forces, to represent the most computationally intensive portion of most molecular dynamics software. One of its purposes is to function as a test problem for estimating the performance of candidate HPC architectures, and broaden the range of programming techniques which can be evaluated without restructuring entire applications. Explorations of such techniques are an intersection of HPC platform design choices and the modeled scientific application, making the study relevant both to computer architecture and molecular dynamics communities working toward exascale computations.

The Intel Xeon Phi architecture is targeted because its highly parallel design has driven present petascale installations, and suggest that architectures with similar properties are relevant for future exascale platforms.

The study describes 14 different optimizations. Each of these optimizations is reported with the speedup it individually obtains over the default OpenMP implementation of CoMD, as shown in Figure 8.

The optimizations are categorized into 4 major categories; thread parallelism, memory optimizations, vectorization and algorithmic modifications. As a measure of their relative effectiveness, their respective fractions out of the sum of individual speed gains is shown in Figure 9, which shows that the optimizations relating to improving the effect of the Xeon Phi's high core count have the most pronounced effect, while efforts to improve the computational method and efforts to better exploit per-core vector units account for similar levels of potential improvement.

The results suggest that adapting MD applications to many-core architectures for the purpose of running exascale simulations not only require expressing the program in the relevant programming model, but also mandates revision of program design decisions which pertain to the problem representation and choice of algorithms.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP234: Study of Xeon Phi Performance of a Molecular Dynamics Proxy Application [3].

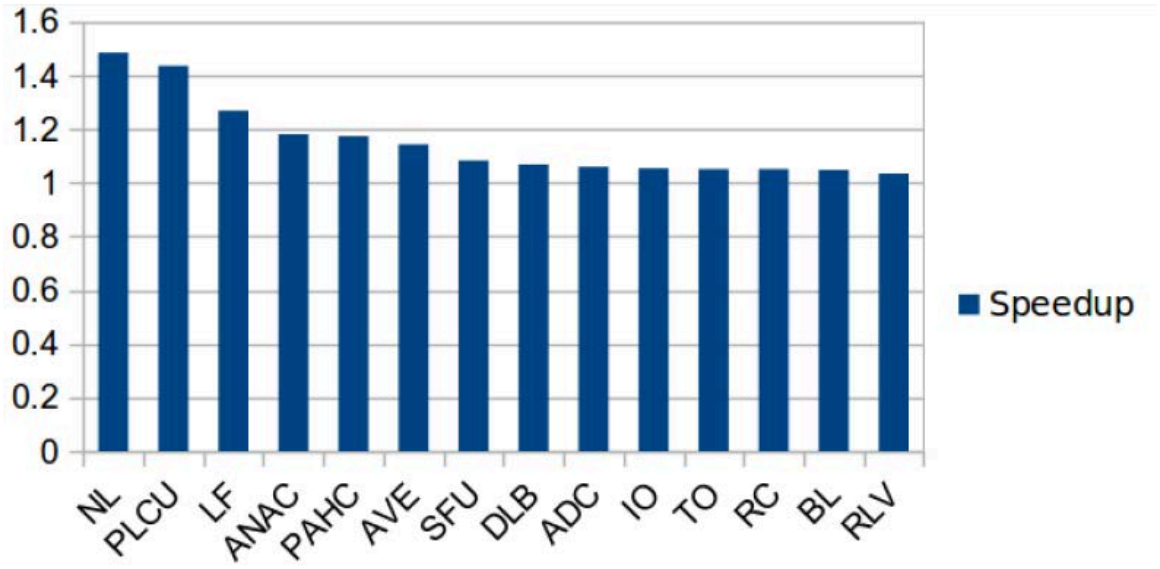

Figure 8: Performance improvements per optimization, 600K

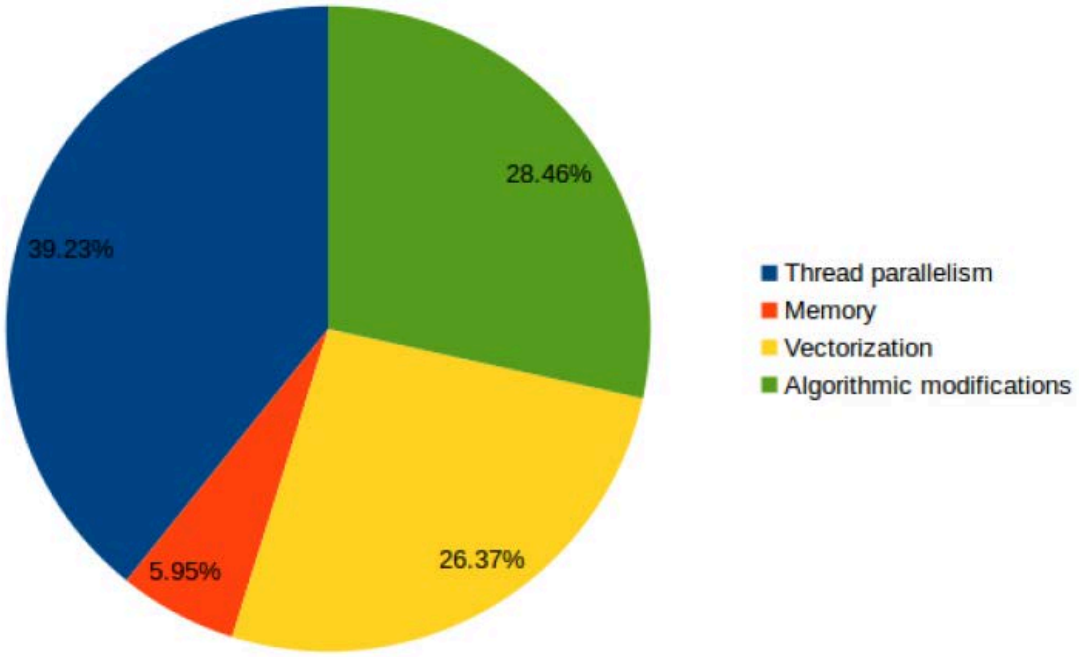

Figure 9: fraction of total improvement by optimization category

3.5 Characterization and optimization of sparse computations on Intel Xeon Phi

WP238: Characterization and optimization of sparse computations on Intel Xeon Phi

Authors: Athena Elafrou (NTUA), Georgios Goumas (NTUA)

HPC Tool/Technique: OpenMP

Application: Sparse matrix-vector multiplication (SpMV)

CoEs/Scientific Communities: MAX, BioExcel, ESiWACE, EoCoE

Sparse matrix-vector multiplication (SpMV) is a fundamental building block of popular iterative methods for the solution of sparse linear systems ($Ax = b$), and the approximation of eigenvalues and eigenvectors of sparse matrices ($Ax = \lambda x$). Optimizing SpMV has always been a challenging task due to a number of inherent performance limitations, as a result of the algorithmic nature of the kernel, the employed sparse matrix storage format and the sparsity pattern of the matrix. SpMV is characterized by a very low flop:byte ratio, indirect memory references as a result of storing the matrix in a compressed format and irregular memory accesses to the source vector due to sparsity.

In this project, we investigate the problem of efficiently porting SpMV to the Intel Xeon Phi coprocessor. The larger number of cores and shallower memory hierarchy of this platform compared to common multicore systems overly exposes inherent structural weaknesses of different sparse matrices, intensifying performance issues beyond the traditionally reported memory bandwidth limitation. We have identified the following bottlenecks, each represented by a class: MB for memory bandwidth, CML for cache miss latency, IMB for workload imbalance and CMP for computation. Figure 10 shows the class distribution for a suite of 125 matrices from the University of Florida Sparse Matrix Collection [4] on an Intel Sandy Bridge multicore processor and the Intel Xeon Phi manycore coprocessor, where every matrix is assigned to the class that represents its major performance bottleneck. It is clear that some bottlenecks are more profound on Xeon Phi and, thus, addressing every bottleneck whenever it emerges is determinant to achieving high performance.

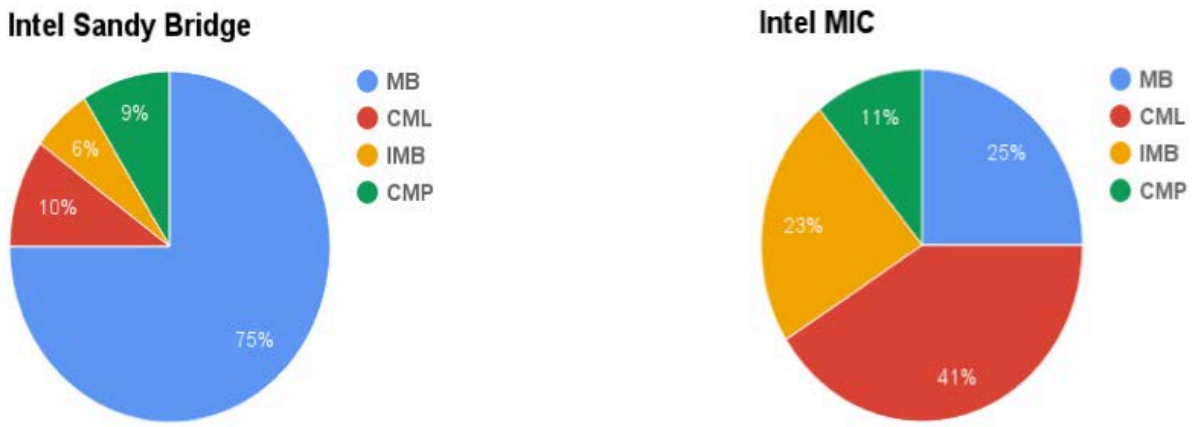

Figure 10: Class distribution on Intel Sandy Bridge and Intel MIC

We, thus, propose a matrix-adaptive optimization technique that leverages machine learning to achieve adaptivity. In particular, the proposed technique first identifies the major performance bottleneck of SpMV for the given matrix on-the-fly and then selects and generates code for a suitable optimization to tackle it. We provide two models for identifying the bottleneck: our first model, namely the profiling-based classifier, is a rule-based algorithm that requires performance bounds to be determined for the input matrix during an online profiling phase, while our second model, namely the feature-based classifier, is a classifier trained with machine learning algorithms (Decision Tree or Naive Bayes) that only uses comprehensive structural features of the sparse matrix. Our optimizations are based on the widely used Compressed Sparse Row (CSR) storage format and have low preprocessing overheads, making our overall approach practical even in the context of iterative solvers that converge in a small number of iterations.

We developed our approach in C++ using the OpenMP parallel programming interface along with the offload programming model of Intel Xeon Phi. In Figure 11 we compare our implementations, one using the profiling-based classifier (denoted as prof) and one using the feature-based classifier (denoted as feat), to the highly optimized Intel Math Kernel Library (referred to as MKL), a baseline CSR SpMV implementation with no vectorization and no prefetching applied (referred to as baseline) and a CSR SpMV implementation compiled with -O3 (referred to as -O3).

Our profiling-based classifier achieves a 2.2x average speedup over MKL, while the best-trained feature-based classifier achieves 2.15x respectively. The results indicate that an SpMV optimizer that adapts to the matrix characteristics is crucial to attaining higher performance for this kernel on Xeon Phi. We anticipate this trend will become even more relevant in future exascale systems that plan to adopt manycore architectures as their main processors, e.g., the current generation of Intel Xeon Phi, codename Knights Landing. More importantly, this adaptivity can come at a low cost, which is a requirement for problems that require a smaller number of iterations to converge, e.g., preconditioned solvers. However, for problems that can amortize higher preprocessing costs, better performance can be attained by incorporating more sophisticated optimizations and exploring combined optimizations for matrices that have competing bottlenecks.

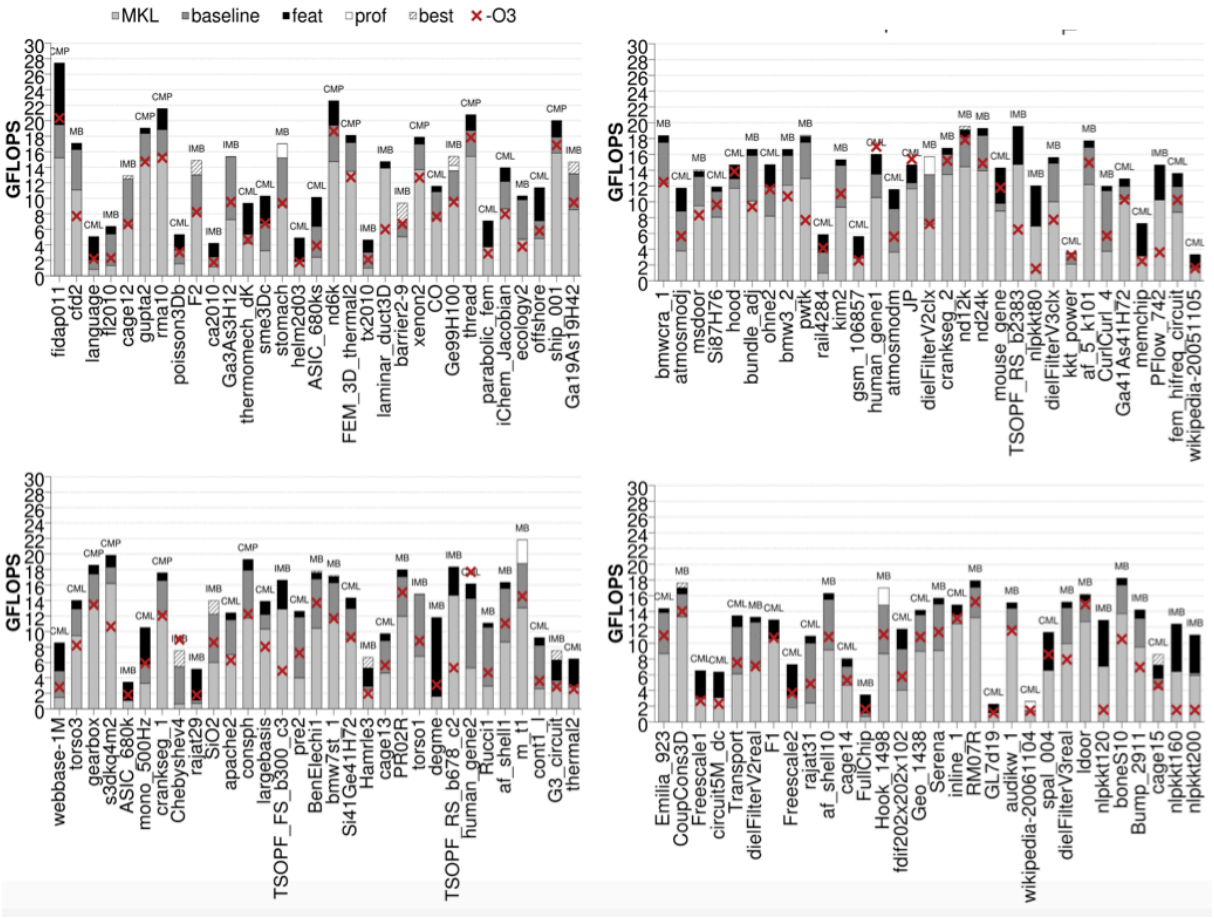

Figure 11: Performance landscape on Intel Xeon Phi

The work presented here has potential links to CoEs MAX, BioExcel, ESiWACE and EoCoE. Also, it can be integrated in widely used sparse solver libraries, including the PETSc [5], [6] and Trilinos [7] scientific toolkits.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP238: Characterization and optimization of sparse computations on Intel Xeon Phi [3].

3.6 Using GPU accelerators for improving performance and scalability in material physics simulations

WP235: Using GPU accelerators for improving performance and scalability in material physics simulations

Authors: Mariusz Hruszowiec (WCSS), Paweł Potasz (WCSS), Agnieszka Szymańska-Kwiecień (WCSS), Mariusz Uchroński (WCSS)

HPC Tool/Technique: CUDA

Application: Configuration interaction method

CoEs/Scientific Communities: MaX, NoMaD

This work focused on parallel simulation of electron-electron interactions in materials with non-trivial topological order (i.e. Chern insulators). A problem of electron-electron interaction systems can be solved by diagonalizing a many-body Hamiltonian matrix in a basis of configurations of electrons distributed among possible single particle energy levels - a configuration interaction method. The number of possible configurations exponentially

increases with a number of electrons and energy levels; 6 electrons occupying 24 energy levels corresponds to the dimension of Hilbert space about 10^5, for 12 electrons it gives 10^6 configurations. Solving such a problem requires effective computational methods and highly efficient optimization of the source code.

The main goal of this work, undertaken within PRACE-4IP project, was to prepare GPU implementation for improving performance and scalability in parallel simulations of electron-electron interaction in materials with a non-trivial topological order. Such systems are expected to be useful in study and understanding of new topological phases of matter, and in a further future can be used to design novel nanomaterials. During technical work most promising routines of Fortran/OpenMP code were identified and ported to the GPU accelerators using CUDA. This work can support scientific communities focused on research in condensed matter physics. Project results may be useful/related to the CoEs focused on identifying novel materials such as MaX or NoMaD.

The work undertaken within this task started from a basic code improvements such as: porting code from Fortran77 to Fortran90, code reorganization and refactoring. Also some effort was spend on improving implementation for generation of configurations of electrons distributed among possible single particle energy levels. The next step was compilation of the Modified Lanczos Method for Atom Annihilation Creation (MLM4AAC) application using gfortran from the GNU compiler suite (v4.9.2). The compilation was done successfully and the application was analysed with gprof (v2.20) tool. The medium size problem was taken as an example for performance tests and it was conducted on the NOVA system [8]. The analysis has shown that function twooper is using over 80% of the processor time. It was obvious choice to check if it can be improved, i.e. by implementation on CUDA device.

In the CUDA kernel function cu_inner_loop (Figure 12) each thread calculates a nonzero matrix element and multiplies the matrix by an initial vector. The matrix is in a block diagonal form with row index corresponding to distributions of particles on states which is called a configuration. Each configuration is represented by a binary number. Such configuration space is divided into subspaces. Every CUDA thread performs computation for one configuration of particles. In general each thread creates a new configuration by annihilation of “ones” within given configuration and creates new “ones”. Index p for a new configuration is calculated using a hash table. New configuration is checked if it is in the same subspace. Finally there is a matrix-vector product.

A set of computational experiments has been performed on two different architectures using three different problem's sizes (small, medium and large). Both OpenMP and CUDA performance were measured for different number of threads. For a reference in every measurement, the time measurement result of 1 OpenMP thread was considered. In the first example there were only 4 particles and 12 states with 160 maximum number of acceptable configurations with Kx, Ky subspaces lengths 3 and 4 respectively. In the second example there were 6 particles with 24 states and 5700 maximum number of configurations with Kx, Ky subspaces lengths 4 and 6 respectively. In the last test case there were 5 particles with 30 states and 150000 maximum configurations with Kx, Ky subspaces lengths 5 and 6.

```
__global__ void cu_inner_loop(const fint8 iconf, ... , const fint8 sdbl_vec_size)
{
  fint i, j, k, l; // indexes for each thread to work on
  fint8 p;
  fint8 empty[MAX_SIZE]; // N_st
  ...
  __shared__ fcomplex8 rr_pp[256]; // shared
  p = blockIdx.x * blockDim.x + threadIdx.x;
  rr_pp[threadIdx.x] = rr[p];
  __syncthreads();
  if (p < iconf) // check if indexes are valid
  { // single particle energies
    for(ii=0; ii<N_part; ++ii)
      cf[ii] = Mat_P[p + ii*iconf];
    for(k=0; k < N_part; ++k)
    {
      for(l=k+1; l < N_part; ++l)
      {
        for(int ii=0; ii < N_st-N_part; ++ii)
          empty[ii] = emp[p + ii * iconf];
        ...
        for(j=0; j < N_st-N_part+1; ++j)
        {
          for(i=j+1; i < N_st-N_part+2; ++i)
          {
            empi = empty[i]-1;
            bin2 = bin | (1ULL << (N_st-(empj+1)));
            bin2 |= (1ULL << (N_st-(empi+1)));
            ...
            rr_pp[threadIdx.x].real = rr_pp[threadIdx.x].real +
                (mac.real*vec1[s_dbl-1].real - mac.imag*vec1[s_dbl-1].imag);
            rr_pp[threadIdx.x].imag = rr_pp[threadIdx.x].imag +
                (mac.real*vec1[s_dbl-1].imag + mac.imag*vec1[s_dbl-1].real);
          }
        }
      }
    }
  }
  rr[p] = rr_pp[threadIdx.x];
  __syncthreads();
}
```

Figure 12: CUDA kernel function cu_inner_loop

Table 5 contains performance results for the OpenMP code executed on NOVA (AMD Opteron 6274) and BEM (Intel Xeon E5-2670 v3) systems. For eva34 test case, speedup values are relatively small and don't change significantly with the increasing number of OpenMP threads on both systems. The same behaviour can be observed for eva46 and eva56 test cases on NOVA system. However, speedup values increases significantly with increasing number of OpenMP threads on BEM system.

OpenMP threads	**speedup on NOVA**			**speedup on BEM**		
	eva34	eva46	eva56	eva34	eva46	eva56
2	1.02	1.56	1.63	1.70	2.20	2.52
4	1.45	2.68	2.84	1.96	3.88	4.27
8	1.66	3.83	4.34	2.34	6.44	6.54
16	1.46	3.44	4.36	2.47	7.24	8.06
32	1.33	3.09	3.63	2.33	8.33	9.34
64	1.03	3.18	3.34	0.81	10.13	10.70

Table 5: Performance results for OpenMP implementation

Table 6 contains performance results for the CUDA code executed on Tesla M2075 located within NOVA cluster for one and two GPUs. For eva34 test case, speedup values are very small and don't change with the increasing number of CUDA threads per block. For eva46 and eva56 test cases, best values of speedup was obtained for 64 CUDA threads per block. For eva46, best value of speedup is equal 20.76 for one GPU and 57.87 for two GPUs, which are respectively 2x and 5x better in comparison with best OpenMP speedups.

CUDA threads per block	speedup for one GPU			speedup for two GPUs		
	eva34	eva46	eva56	eva34	eva46	eva56
8	0.16	11.49	9.50	0.62	27.53	23.08
16	0.13	13.92	11.90	0.60	34.15	29.61
32	0.17	16.27	12.40	0.57	44.01	32.20
64	0.17	20.76	15.36	0.59	57.87	42.57
128	0.17	19.50	15.32	0.57	55.14	41.83
256	0.17	20.45	14.25	0.57	55.56	38.66

Table 6: Performance results for CUDA implementation

Best values of speedup for OpenMP and CUDA are shown on Figure 13. It can be seen that speedup values for the code executed on GPU are significantly larger than for the code executed on CPU.

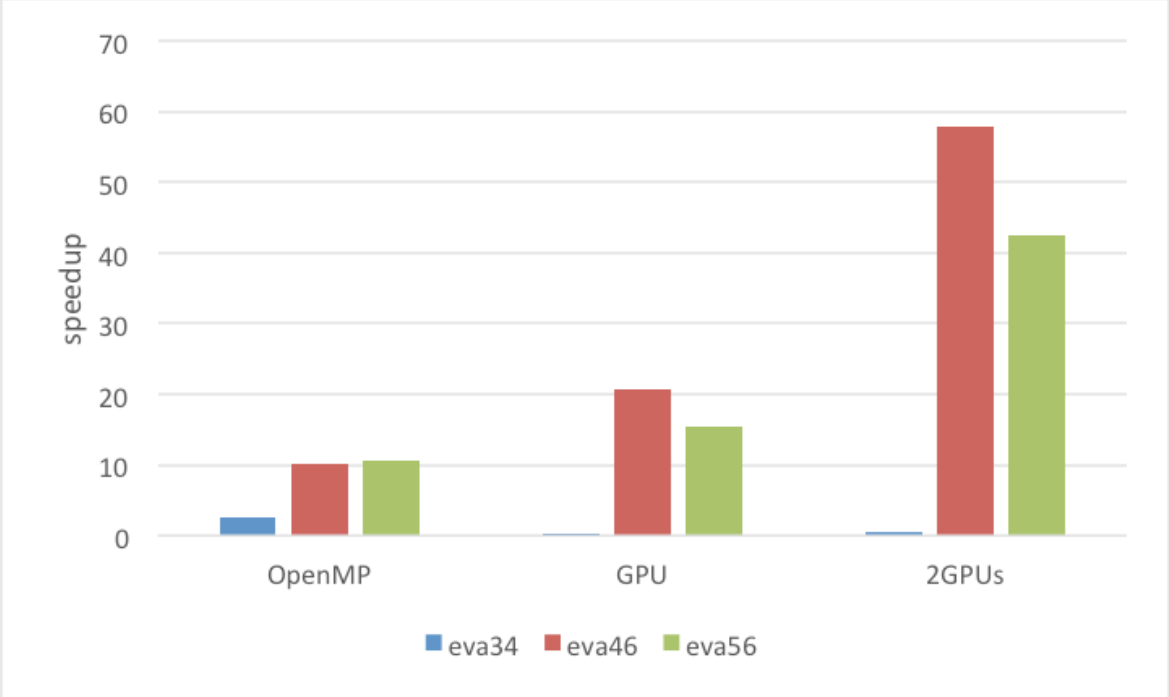

Figure 13: Best speedup values

In conclusion, our work has demonstrated the potential of using GPU accelerators for improving performance and scalability in material physics simulations. The main factor in obtaining high performance GPU computing is to identify promising areas of application that allow for massive parallelism. For this purpose gprof tool was used and a configuration interaction method was identified as the most promising to port on GPU. The GPU implementation provides significant increase of performance (~ x2 speedup) in comparison with parallel OpenMP base implementation. Nature of problem allows to enable GPU computational potential by enabling code execution on multiple GPUs. This goal has been achieved by using possibility of independent computations for each subspace of configurations space. Multi GPU approach results in next significant increase of performance (~ x5 speedup) on two GPUs.

Proposed approach result in shortage of computation time and give possibility to simulate larger electron-electron interaction systems in future exascale HPC systems with GPUs. Possible way of future evaluation of GPU acceleration in optimized code could be

implementation hybrid MPI+CUDA code which allows execute simulations on GPU clusters. Mentioned future work requires some changes in existing code but level of its complexity is average and code can be exploited on future exascale HPC systems.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP235: Using GPU accelerators for improving performance and scalability in material physics simulations [3].

3.7 Gabriel 1.3: a Fortran library for fast, verified and convenient message passing

WP239: Gabriel 1.3: a Fortran library for fast, verified and convenient message passing

Authors: John Donners (SURFsara)

HPC Tool/Technique: gabriel, MPI 3.0, Fortran 2015

Application: LES-COAST, NEMO

CoEs/Scientific Communities: ESiWACE

The gabriel library combines features in MPI-3.0 and Fortran 2015 with classic advantages of Fortran to improve the use of MPI by geophysical models on regular grids. The user can define a composition of arrays with one collective call. Another call then constructs a distribution of messages, e.g. to exchange halo regions. This greatly simplifies the coding of halo exchanges, since the developer never has to define the halos explicitly. It allows flexible decompositions and variable halo widths without additional code.

Gabriel supports irregular domain decompositions with periodic boundaries, holes in the domain and a variable halo width. See Figure 14. The local array is bounded by the thick black line. The local computational domain is green and neighbouring computational domains are light brown; halo regions are dark brown and the blue halo regions at the top are filled as periodic boundary by gabriel. The blue halo region in the center of the domain is not covered by any local computational domain and will therefore be ignored by gabriel. This flexibility has been used in the LES-COAST model, a large eddy simulation, to change a one-dimensional decomposition into a two-dimensional decomposition. This resulted in a maximum rank count about ten times higher for the application.

In the NEMO model, one of the core models in the ESiWACE CoE, the hand-coded halo exchanges could be replaced with only a few calls to gabriel:

```
! Initialization
    from=(/1+jpreci,1+jprecj/)
    to=(/iihom+jpreci,ijhom+jprecj/)
    call ptabbox%init(pt2d,from,to,mpi_comm_opa,offset=(/nimpp,njmpp/))
    call ptabdist%halo(ptabbox)
    call ptabdist%create
    ...
    ! Halo update
    call ptabdist%update(pt2d)
```

The performance of the gabriel library depends critically on the implementation of MPI derived datatypes in the underlying MPI-library, which is slower than expected. The MPI_Neighbor_alltoallw call, that is used in a gabriel update, does not show scalability bottlenecks in gabriel, nor in other applications. As a result, a lower overall model performance was measured for this case. However, gabriel offers scalability, verification and ease of use for models on regular grids.

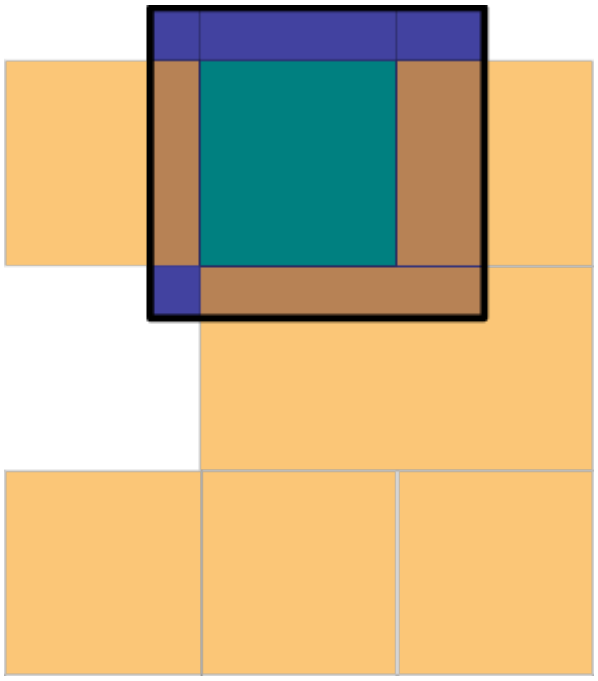

Figure 14: Irregular domain decomposition in gabriel

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP239: Gabriel 1.3: a Fortran library for fast, verified and convenient message passing [3].

4 Debuggers and Profilers

In this section, we report on two projects that have each focused on exploiting state-of-the- art profilers in order to enable applications for European multi-petaflop/future exascale systems. Each subsection provides a summary of the project along with a reference to the PRACE-4IP whitepaper associated with the project, as well as the CoEs that we expect will find the work of interest. We recommend that the reader also refers to the associated whitepaper for each project, which provides a more detailed report on the projects than is provided here. The list of profilers, applications targeted, as well as CoEs focused on, can be seen in Table 7.

HPC Tool/Technique	Application	Exascale/CoEs/Scientific Communities
HPCToolkit, Extrae, Paraver, SCALASCA, Intel Tools	GROMACS	NOMAD, BioExcel and E-CAM
Intel Advisor Cache Aware Roofline Model (CARM)	IFS, HARMONIE LAITRI	ESiWACE

Table 7: Debuggers and Profilers exploited along with corresponding applications

4.1 Profiling and Tracing Tools for Performance Analysis of Large Scale Applications

WP237: Profiling and Tracing Tools for Performance Analysis of Large Scale Applications

Authors: Jerry Eriksson (HPC2N), Pedro Ojeda-May (HPC2N), Thomas Ponweiser (RISCSW), Thomas Steinreiter (RISCSW)

HPC Tool/Technique: HPCToolkit, Extrae, Paraver, SCALASCA, Intel Tools

Application: GROMACS

CoEs/Scientific Communities: NOMAD, BioExcel and E-CAM

The usage of modern profiling and tracing tools is vital for understanding program behaviour, performance bottlenecks and optimisation potentials in HPC applications. Despite their obvious benefits, such tools are still not that widely adopted within the HPC user community. Evidence for this fact is given for example in the PRACE-4IP deliverable D7.3 "Inventory of Exascale Tools and Techniques" [1], summarizing the results of a survey on HPC needs of selected CoEs. One of the findings of the survey is that among all CoEs still the most favoured method for performance analysis is manual code instrumentation (i.e. time measurement) and console or log output (compare Figure 15). However, for a well-grounded and deep understanding of program performance behaviour and optimisation opportunities, the usage of specialised profiling tools is important already today and will get even more important in view of the ever-increasing complexity and heterogeneity of HPC systems on the road to Exascale.

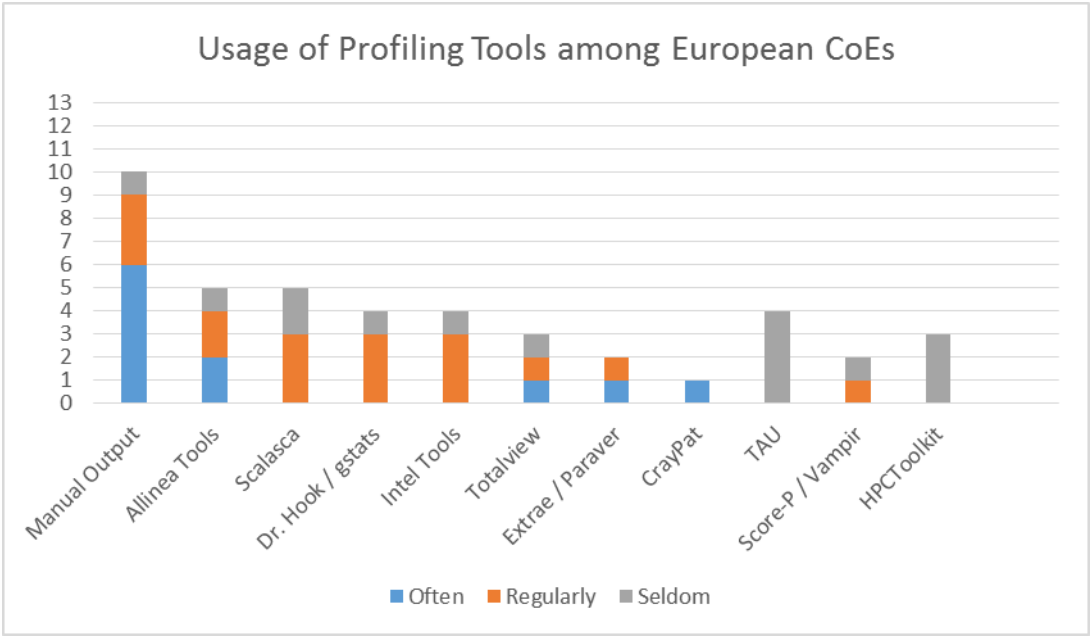

Figure 15: PRACE-4IP D7.3 survey result on the usage frequency of profiling tools among CoEs (Data from BioExcel, CoeGSS, E-CAM, ESiWACE and MaX is displayed. 13 points of contact in total, 9 of which are from ESiWACE.)

The work "Profiling and Tracing Tools for Performance Analysis of Large Scale Applications" addresses this issue by presenting and comparing the capabilities of four different performance analysis tools, which are 1) HPCToolkit, 2) Extrae and Paraver, 3) SCALASCA and 4) Intel Tools. The aim of the work is to raise the general awareness for the benefits of using specialized performance analysis tools and to lower the threshold for potential users to getting started with such tools.

For demonstrating the practical tool usage, the application code GROMACS has been selected as benchmark code due to its high relevance for the molecular dynamics community and in particular for the three CoEs NOMAD, BioExcel and E-CAM. Of course most results (like tool descriptions, hints, best practices, etc.) are not specific to applications with GROMACS and are therefore also useful for an even broader audience.

The main result of the work is a characterization and comparison of the different tools and their primary application areas: HPCToolkit is well suited for intra-node performance optimization using profiling or tracing. In addition, HPToolkit's profile viewer can also be

used for analysing scalability losses by using user-defined metrics and a method called "differential profiling".

Extrae and Paraver supports a wide range of analysis types. Analyses are based on highly customizable trace and histogram visualizations of different metrics such as hardware counters. Also MPI messages can be investigated in the trace view. SCALASCA supports graphical and statistical performance analysis and is especially useful to locate events which lead to unbalanced workloads across processes. Such events are for instance Late Sender or Late Receiver where sending or receiving of messages is not properly synchronized, causing a decrease in overall software performance. Intel Trace Analyzer and Collector (ITAC) is a tool for MPI communication analysis and is primarily helpful for optimizing inter-node performance and scalability. In addition, ITAC can also be used for MPI correctness checking. Complementary to ITAC, Intel Vtune can be used for node-level performance optimization.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP237: Profiling and Tracing Tools for Performance Analysis of Large Scale Applications [3].

4.2 Exploiting Intel Advisor's Cache-Aware Roofline Model (CARM) tool to enable European Weather Forecasting Applications on Intel Knights Landing

WP241: Optimization of IFS Subroutine LAITRI on Intel Knights Landing

Authors: Oisín Robinson (ICHEC), Alastair McKinstry (ICHEC), Michael Lysaght (ICHEC)

HPC Tool/Technique: Intel Advisor Cache Aware Roofline Model (CARM)

Application: IFS, HARMONIE, ESCAPE Dwarves

CoEs/Scientific Communities: ESiWACE

Improving energy/time costs in weather forecasting codes is a key objective of the FET-HPC project, ESCAPE (Energy-efficient Scalable Algorithms for weather Prediction at Exascale), and this includes targeting IFS by the 'divide-and-conquer' approach in optimizing the most heavily used subroutines. This goal is predicated on a move to Exascale computing, which is expected to exploit massively parallel node-level chip architectures. An example of an emerging many-core platform that may serve as a stepping stone to future node-level architectures on exascale systems is the Intel Xeon Phi Knights Landing (KNL) processor.

This work investigates performance optimization of a heavily-used weather code subroutine on the Intel Xeon Phi Knights Landing platform, aided by the use of a beta version of the Intel Advisor tool, which provides a useful means of identifying node-level performance bottlenecks and guidance on attainable performance by way of the recently developed 'Cache Aware Roofline Model (CARM)'. The CARM shows a plot of arithmetic intensity vs. memory bandwidth, taking account of the influence of the cache hierarchy on performance, and gives an immediate metric showing whether processor/cache/memory are being pushed to their theoretical performance limit. The CARM represents a development beyond the original 'roofline model', which plots the same quantities, but with respect to main memory bandwidth only – leaving a sloping 'memory bound' part, and a flat 'compute bound' part instead of multiple slopes/levels. A screenshot of the CARM generated by Intel Advisor is shown below in Figure 16, where we see (explicitly) that actual performance is bounded by L2 bandwidth, meaning that increasing performance should involve utilising L1 cache more effectively. The CARM is now available in the latest (2017) version of the Intel Advisor. We can also see that to push the processor to its limit, we would need to make use of the vector add and FMA (Fused Multiply-Add) capabilities.

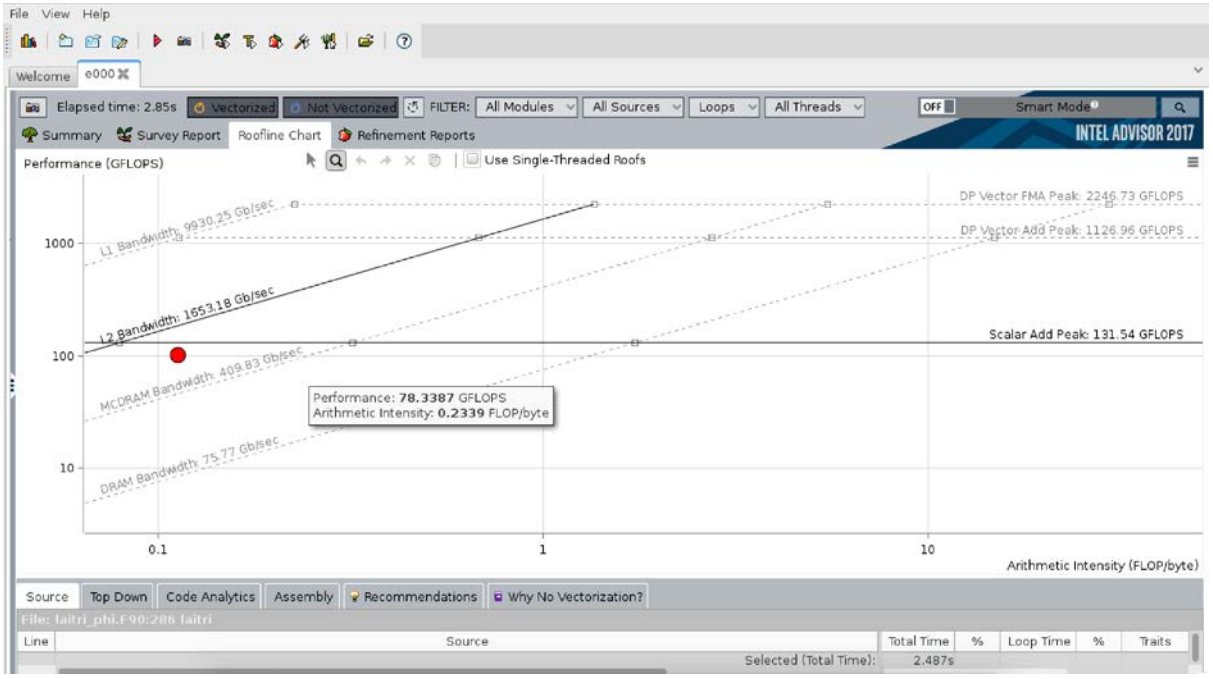

Figure 16: Intel Advisor CARM

The application we focus on is the weather code ESCAPE dwarf, known as LAITRI, which combines a driver harness and subroutine of the ECMWF numerical weather prediction (NWP) suite, IFS. It accounts for about 4% of the actual runtime of IFS, which is high compared to the (many) other subroutines in use. We give an overview of its actual function which is the description of 'advection' of wind/temperature/pressure values in an interpolation cube in Figure 17 and describe how our optimization methods have been guided by Intel Advisor's implementation of CARM. We also demonstrate that KNL performance is highly competitive relative to the Intel Xeon Phi Knights Corner (KNC) co-processor and that performance is also competitive relative to a 2-socket Ivy Bridge node (see Figure 18).

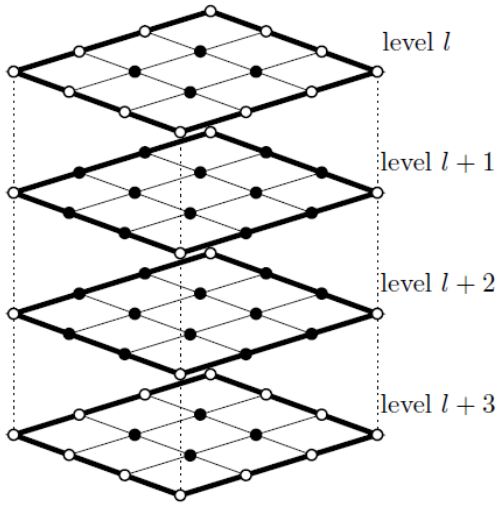

Figure 17: LAITRI interpolation stencils in advection cube

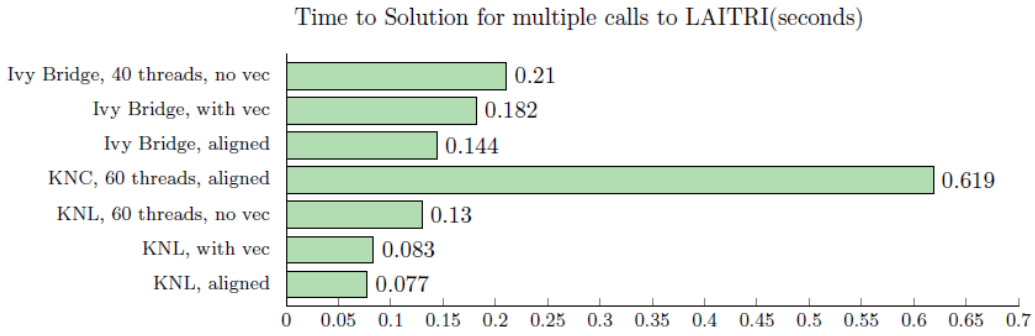

Figure 18: Performance Comparison of LAITRI on various platforms

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP241: Optimization of IFS Subroutine LAITRI on Intel Knights Landing [3].

5 Scalable Libraries and Algorithms

In this section, we report on three projects that have each focused on exploiting state-of-the-art libraries and algorithms in order to enable applications for European multi-petaflop/future exascale systems. Each subsection provides a summary of the project along with a reference to the PRACE-4IP whitepaper associated with the project, as well as the CoEs that we expect will find the work of interest. We recommend that the reader also refers to the associated whitepaper for each project, which provides a more detailed report on the projects than is provided here. The list of libraries and algorithms, applications targeted, as well as CoEs focused on, can be seen in Table 8.

HPC Tool/Technique	Application	Exascale/CoEs/Scientific Communities
MPI, MUMPS, PaStiX, MaPHyS	TRACES	EoCoE, Computational geology community
MPI, MUMPS, PaStiX, MaPHyS	HORSE	EoCoE, Computational electromagnetics community
Store-and-Forward Scheme, MPI	Sparse matrix-vector multiplication (SpMV)	EoCoE

Table 8: Scalable Libraries and Algorithms exploited along with corresponding applications

5.1 Hybrid iterative-direct solution strategy for improving the scalability of nuclear waste management simulations

WP228: Hybrid iterative-direct solution strategy for improving the scalability of nuclear waste management simulations

Authors: Emmanuel Agullo (Inria Bordeaux), Luc Giraud (Inria Bordeaux), Matias Hastaran (Inria Bordeaux), Stéphane Lanteri (Inria Sophia Antipolis), Laurent Loth (ANDRA), Ludovic Moya (Inria Sophia Antipolis), Florent Pruvost (Inria Bordeaux), Jean Roman (Inria Bordeaux), Olivier Rouchon (CINES)

HPC Tool/Technique: MPI, MUMPS, PaStiX, MaPHyS

Application: TRACES

CoEs/Scientific Communities: EoCoE, Computational geology community

This work is concerned with improving the scalability of nuclear waste management simulations performed using the TRACES software [9]. TRACES (Transport of RadioACtive Elements in Subsurface) is simulation software used by ANDRA (French National Agency for Radioactive Waste Management) for phenomenological description and performance/safety assessment of the integrated disposal repositories and their geological environments. This software is of interest to the EoCoE. It is written in Fortran 95 and 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. The application solves PDEs modeling groundwater flow and radionuclide transfer in (un)saturated porous media. Darcy's law and diffusivity in confined aquifer equations at one or several steady states describe groundwater flow. Radionuclide transfer is described with advection, diffusion and dispersion, with linear adsorption and radioactive decay. The corresponding sets of PDEs are linear and mixed hyperbolic-parabolic. Spatial discretization of 3d problems is based on a discontinuous finite element method for the advection part and mixed hybrid finite element method for the other parts, formulated on conforming, structured or unstructured grid with hexahedral elements. Time integration relies on explicit or implicit schemes and in the latter case; a Newton method is used for the linearization of discrete system leading to the formulation of large sparse real coefficients linear systems of equations.

For the solution of these systems, TRACES makes use of parallel preconditioned iterative solvers offered by the Hypre library [10] developed on the Center for Applied Scientific Computing at Lawrence Livermore National Laboratory, namely CG/AMG and GMRES/AMG (i.e., algebraic multigrid preconditioned Krylov iterative solvers). However, although being a very efficient preconditioner from the numerical point of view, AMG is also known to scale poorly on massively parallel systems when it comes to solve general sparse systems, which presents a challenge when enabling the TRACES application on current and future extreme-scale sytems. In this project, we study the possibility of improving the scalability of TRACES by considering the use of an algebraic hybrid iterative-direct solver whose design is based on domain decomposition principles. This approach is implemented in the algebraic hybrid iterative-direct solver named MaPHyS (Massively Parallel Hybrid Solver) [11], [12].

The new version of the TRACES simulation software has been evaluated on the Occigen system at CINES. For that purpose, we considered a use case involving a realistic geological medium depicted in Figure 19. This domain contains 28 layers and covers a domain of size 600 km x 600 km x 1000 m. The underlying mesh contains 5,944,891 cells and 17,858,966 faces. The whole simulation workflow consists of two steps: i. computation of the steady flow conditions, i.e. the hydraulic pressure and the velocity field; ii. these flow conditions are used for the simulation of the transport of radinucleiudes. Here, we only consider the first phase of this workflow that amounts to solving Darcy's equation.

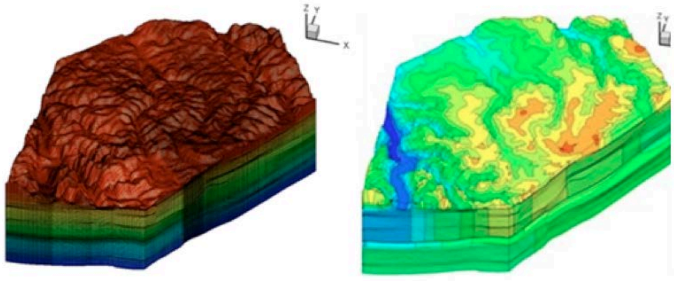

Figure 19: Multi-layer geological medium for the simulation of hydraulic glow conditions with the TRACES software

A strong scalability assessment of two solution strategies is shown in Figure 20. The reference solution strategy is the combination CG/AMG of the Hypre library. We observe that the parallel speedup obtained when using the MaPHyS solver is slightly better, thus confirming the expected behavior. However, from the wall clock time viewpoint, the simulation based on the CG/AMG solution strategy is more than 8 times faster than the one based on the MaPHyS solver. In order to explain this result, we compare the time per iteration for each solution strategy in Figure 21. On this graph we clearly show that the MaPHyS hybrid iterative-direct solver is faster than CG/AMG on a per iteration basis. However, with the MaPHyS solver, the number of iterations to convergence increases with the number of subdomains (here, one subdomain is assigned to one core). This numerical scalability issue with domain decomposition solvers is very well known. Appropriate strategies have been designed for symmetric positive definite systems in the form of so-called coarse grid correction. Such a correction has been recently introduced in the MaPHyS solver, but has not been exploited for the simulations discussed here [13].

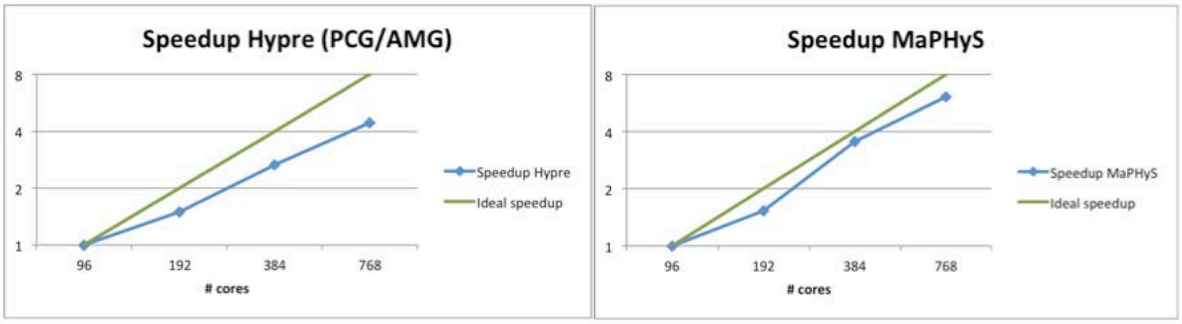

Figure 20: Performance results on the Occigen system at CINES (Speedup)

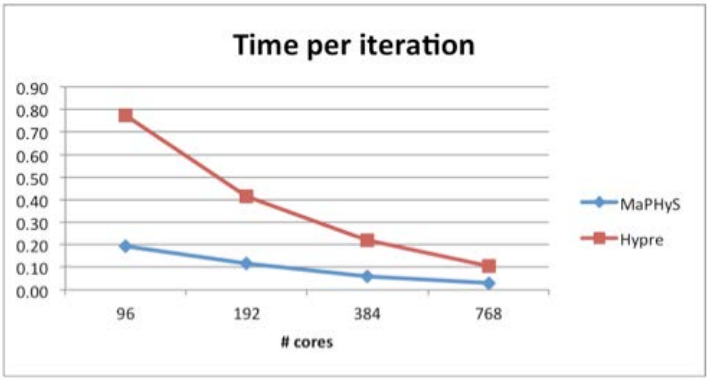

Figure 21: Performance results on the Occigen system at CINES (Time per solver iteration)

In summary, although some theoretical issues still need to be addressed for improving the numerical scalability of the algebraic MaPHyS solver (i.e. minimizing the increase of the

number of iterations with the number of subdomains), the new version of the TRACES simulation software has demonstrated promising capabilities for simulating very large problems relevant to nuclear waste management in a scalable way. The work presented here is currently consolidated by ANDRA and will soon benefit from a new version of the MaPHyS solver with better numerical scalability properties.

The work presented here has potential links with the EoCoE. In particular, advanced scalable numerical linear algebra black-box solvers such as MaPHyS is considered in a task of the WP1 transversal workpackage in EoCoE work plan. In EoCoE, MaPHyS is further extended and exploited for several application domains. For example, it has been integrated in the general purpose FEM CFD code Alya developed at BSC [14] as part of the application pillar workpackage on 'Meteorology for Energy' and a scalability study is scheduled for 2017. In addition, preliminary experiments on matrices provided by a partner from the EoCoE workpackage on 'Water for Energy' have been performed, where the integration of MaPHyS in the corresponding simulation software is discussed to assess its efficiency at scale.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP228: Hybrid iterative-direct solution strategy for improving the scalability of nuclear waste management simulations [3].

5.2 High order finite element schemes and domain decomposition solvers for large-scale simulations in electromagnetics

WP229: High order finite element schemes and domain decomposition solvers for large-scale simulations in electromagnetics

Authors: Emmanuel Agullo (Inria Bordeaux), Luc Giraud (Inria Bordeaux), Matthieu Kuhn (Inria Bordeaux), Stéphane Lanteri (Inria Sophia Antipolis), Ludovic Moya (Inria Sophia Antipolis), Jean Roman (Inria Bordeaux), Olivier Rouchon (CINES)

HPC Tool/Technique: MPI, MUMPS, PaStiX, MaPHyS

Application: HORSE

CoEs/Scientific Communities: EoCoE, Computational electromagnetics community

This work is concerned with improving the scalability of large-scale simulations of frequency-domain electromagnetic wave propagation based on a recently developed innovative simulation software. The software combines a high order finite element discretization scheme formulated on an unstructured tetrahedral grid, and scalable sparse linear solvers. The enabling numerical tool is a domain decomposition solution strategy for the sparse system of linear equations resulting from the spatial discretization of the underlying PDEs (Partial Differential Equations), that can be either a purely algebraic algorithm working at the matrix operator level (i.e. a black-box solver), or a tailored algorithm designed at the continuous PDE level (i.e. a PDE-based solver). The PDEs at hand here are the frequency-domain (or time-harmonic) Maxwell equations.

The underlying simulation software is called HORSE (High Order solver for Radar cross Section Evaluation). This software aims at solving the full set of 3d time-harmonic Maxwell equations modeling the propagation of a high frequency electromagnetic wave in interaction with irregularly shaped structures and complex media. It relies on an arbitrary high order Hybridized Discontinous Galerkin (HDG) method [15] designed on an unstructured possibly non-conforming tetrahedral mesh, and leads to the formulation of an unstructured complex coefficients sparse linear system of equations. This software is written in Fortran 95 and 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 PaStiX [16]. 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. Two domain decomposition solution strategies are considered in this study: the first one works at the matrix operator level and materializes as the algebraic hybrid iterative-direct solver named MaPHyS (Massively Parallel Hybrid Solver) [11], [12]; the second approach is a variant designed at the continuous PDE level by taking into account the intrinsic characteristics of the system of Maxwell equations.

Two concrete and different applications are considered for illustrating the modeling capabilities of the simulation software and assessing its parallel performances on the road to Exascale: i. the scattering of a plane wave by an aircraft; ii. the interaction of an electromagnetic wave with a heterogeneous model of head tissues. For the first application, the unstructured tetrahedral mesh used in the simulations consists of 1,645,874 elements and 3,521,251 faces. The size, in terms of numbers of DoF (Dgrees of Freedom), for the discrete hybrid variable and electromagnetic field components are summarized in Table 9 for several polynomial interpolation orders in the HDG method. Performance figures (strong scalability analysis) are given in Table 10 and Table 11. We present results of simulations performed using MPI parallelization mode only (combined MPI and multithreaded parallel execution mode will be considered in the future for larger problem sizes). These simulations have been performed on the Occigen system at CINES.

HDG method	# DoF hybrid variable	# DoF EM field
HDG-P1	21,127,506	39,500,976
HDG-P2	42,255,012	98,752,440
HDG-P3	70,425,020	197,504,880

Table 9: Number of degrees of freedom (DoF) of the discrete global and trace systems

HDG method	# Cores	# Iterations	Fact. Time (sec)	Sol. Time (sec)	Wall Time (sec)	Speedup
HDG-P1	384	3	2.6	3.7	6.8	1.0
-	768	4	0.8	2.3	3.4	2.0
HDG-P2	384	10	16.7	40.5	58.7	1.0
-	768	12	5.1	21.5	27.1	2.15
HDG-P3	768	23	18.8	102.1	122.6	1.0
-	1536	26	5.1	52.0	58.7	2.1

Table 10: Scalability for PDE-based Schwarz solution strategy with PaStiX as a subdomain solver

HDG method	# Cores	# Iterations	Fact. Time (sec)	Prec. Time (sec)	Sol. Time (sec)	Wall Time (sec)	Speedup
HDG-P1	384	93	26.3	23.5	21.5	73.4	1.0
-	768	194	7.2	8.1	18.6	34.7	2.1
-	1536	374	2.3	2.5	18.1	23.5	3.1
HDG-P2	768	5000	53.1	55.2	2353.0	2463.0	1.0
-	1536	5000	13.9	16.4	1243.0	1276.0	1.95

Table 11: Scalibility for MaPHyS solution strategy with PaStiX as a subdomain solver

We observe that the PDE-based Schwarz algorithm is the most efficient, and is scalable despite the increase of the number of iterations to convergence when increasing the number of subdomains. This numerical scalability issue with domain decomposition solvers is very well known. Appropriate strategies have been designed for symmetric positive definite systems in the form of so-called coarse grid correction. Such a correction has been recently introduced in the MaPHyS solver, however it cannot be exploited here in its current design because the matrix operator of the HDG hybrid variable system does not have the required mathematical property [13].

In summary, although some theoretical issues still need to be addressed for improving the numerical scalability of the algebraic MaPHyS solver (i.e. minimizing the increase of the number of iterations with the number of subdomains), the new version of the HORSE simulation software, based on domain decomposition algorithms for sparse linear system of equations associated to the HDG formulation, is now capable of solving very large size problems on several thousands of cores in a scalable way. As a future work, we plan to be able to simulate 1 billion DoF problems.

The work presented here has potential links with the EoCoE. In particular, advanced scalable numerical linear algebra black-box solvers such as MaPHyS is considered in a task of the WP1 transversal workpackage in EoCoE work plan. In EoCoE, MaPHyS is further extended and exploited for several application domains. For example, it has been integrated in the general purpose FEM CFD code Alya developed at BSC [14] as part of the application pillar workpackage on ‘Meteorology for Energy’ and a scalability study is scheduled for 2017. In addition, preliminary experiments on matrices provided by a partner from the EoCoE workpackage on ‘Water for Energy’ have been performed, where the integration of MaPHyS in the corresponding simulation software is discussed to assess its efficiency at scale.

Moreover, innovative scalable DG-based solvers such as the one discussed here for the HORSE simulation software can be exploited for the simulation of problems related to photovoltaic; more precisely, for the realistic numerical modelling light trapping in complex solar cell structures. As a matter of fact, a variant devised for the solution time-domain Maxwell equations is considered in a task of the WP1 transversal workpackage in EoCoE work plan, in close interaction with physicists from the Photovoltaik (IEK-5) at the Forschungszentrum Juelich GmbH, in the context of an application pillar workpackage on ‘Meteorology for Energy’.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP229: High order finite element schemes and domain decomposition solvers for large-scale simulations in electromagnetics [3].

5.3 Reducing latency and bandwidth costs in parallel sparse linear solvers

WP230: Reducing latency and bandwidth costs in parallel sparse linear solvers

Authors: Oguz Selvitopi (Bilkent), Cevdet Aykanat (Bilkent)

HPC Tool/Technique: MPI

Application: Sparse matrix-vector multiplication (SpMV)

CoEs/Scientific Communities: EoCoE

This work takes on the communication challenges offered by latency-bound irregular applications, i.e., the applications characterized with high number of average and/or maximum messages per processor. One of the most prominent features of irregular applications is that it is difficult to predict the overhead of the communication operations incurred in parallel solvers due to the irregular sparsity pattern of the coefficient matrix. This makes efficient parallelization difficult for such applications and scaling them becomes more challenging compared to their regular counterparts. One of the key factors for scalability on large-scale systems is the correct encapsulation of the communication cost model. This work considers two important components of the cost model simultaneously in order to achieve better scalability. Our results indicate that achieving Exascale performance necessitates addressing as many components as possible and finding a balance between these components. Our aim is to improve the parallel peformance of a key kernel found almost in any scientific application that is irregular and sparse.

This works investigates a virtual processor mesh structure in order to bound and reduce the latency overhead besides reducing inter-process communication overhead. The idea is to impose a structured communication pattern onto the irregular communication operations and eliminate the irregularity in these operations, which may manifest themselves as high latency or bandwidth overhead. Assuming a prior partitioning phase is performed –although not necessary– with the aim of reducing inter-process communication our approach then aims to bound the latency overhead at the expense of increasing volume. Specifically, we focus on a two-dimensional virtual processor mesh structure and propose an algorithm to perform irregular point-to-point (P2P) communication operations of parallel SpMV in a structured manner on this topology. In a way, our approach is similar to the partitioning models that obtain a Cartesian distribution, with the crucial difference being the proposed method achieves it not in partitioning but in realizing sparse communications.

Our approach necessitates a store-and-forward scheme to realize P2P operations in which a processor sends its messages via other processors it directly communicates with. In the mentioned 2D mesh, a processor can directly communicate only with the processors that are in the same row or column with the processor. The example in Figure 22 depicts the communication stages of store-and-forward scheme for P_5 on a 4×4 processor mesh. The processors that need vector elements from P_5 are $\mathrm{SendSet}(P_5) = \{P_1, P_3, P_4, P_7, P_8, P_{10}, P_{13}\}$. These processors are colored in red in the left portion of the figure indicating they have not received corresponding elements yet. A green color indicates the target processors have received their messages from P_5.

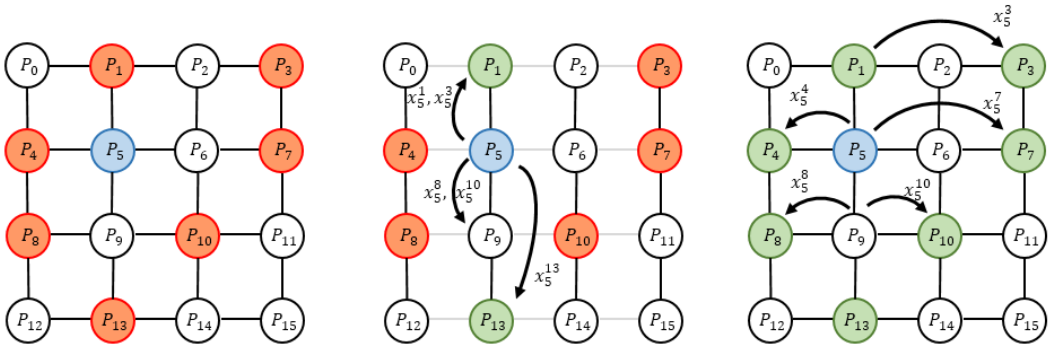

Figure 22: The store-and-forward scheme on 4 × 4 2D mesh

Table 12 compares the described method (referred to as M2D) with the baseline method (BL) in which a processor may communicate with any other processor directly. We also compare another method (BDE) that uses a different virtual processor structure to realize communications. BDE is more aggressive in reducing the latency overhead [17]. The table presents communication statistics and parallel SpMV time of three schemes on 256 processors (vavg: average volume, mavg: average number of messages, mmax: maximum number of messages).

When three schemes are compared in terms of parallel SpMV time, the best performing scheme is clearly M2D. This is due to the fact that this scheme achieves a trade-off between bandwidth and latency costs and hence is able to reduce the parallel runtime around 50% on 256 processors. We expect the EoCoE, BioExcel, ESiWACE CoEs will be interested in these investigations and may be able to exploit some of the findings in future enablement work.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP230: Reducing latency and bandwidth costs in parallel sparse linear solvers [3].

	vavg			**mavg**			**mmax**			**parallel time (usec)**		
matrix	**BL**	**BDE**	**M2D**	**BL**	**BDE**	**M2D**	**BL**	**BDE**	**M2D**	**BL**	**BDE**	**M2D**
144	367	841	463	10.4	8	9.9	34	8	22	102	71	57
598a	295	685	365	10.3	8	9.8	32	8	22	80	68	51
case39	959	4185	1765	30.2	8	12.3	224	8	30	921	164	158
crystk03	282	712	359	12.7	8	10.9	24	8	19	57	68	46
fe_rotor	268	649	330	11.9	8	11.0	34	8	25	71	66	51
gas_sensor	304	740	388	11.6	8	10.8	25	8	19	57	67	46
H2O	1400	3759	1962	26.8	8	17.3	44	8	25	207	97	79
net125	1440	5983	2521	69.5	8	22.0	129	8	28	696	136	122
opt1	292	758	379	11.6	8	10.6	31	8	20	88	70	51
pcrystk03	282	712	359	12.7	8	10.9	24	8	19	59	66	45
pkustk07	427	1266	607	18.2	8	13.9	38	8	23	115	84	68
raefsky4	226	564	286	10.7	8	9.6	25	8	18	67	67	41
ramage02	456	1315	644	18.4	8	14.0	35	8	25	109	86	66
TSOPF_FS_b162_c4	2271	9830	4219	50.5	8	15.3	230	8	30	1015	269	198
wave	416	1044	540	12.8	8	12.0	29	8	23	64	75	55
normalized wrt BL (geomean)	***1.00***	***2.83***	***1.39***	***1.00***	***0.46***	***0.71***	***1.00***	***0.18***	***0.52***	***1.00***	***0.66***	***0.50***

Table 12: Communication statistics and parallel SpMV runtime on 256 processors

6 I/O Management Techniques

In this section, we report on a project that has focused on exploiting state-of-the-art I/O management techiques in order to enable applications for European multi-petaflop/future exascale systems. The subsection provides a summary of the project along with a reference to the PRACE-4IP whitepaper associated with the project, as well as the CoEs that we expect will find the work of interest. We recommend that the reader also refers to the associated whitepaper for the project, which provides a more detailed report on the project than is provided here. The I/O management techiques, applications targeted, as well as CoEs focused on, can be seen in Table 13.

HPC Tool/Technique	Application	Exascale/CoEs/Scientific Communities
MPI-IO, HDF5, NetCDF, Lustre, GPFS, Panasas	benchio	ESiWACE

Table 13: I/O Management Techniques exploited along with corresponding applications

6.1 Parallel I/O Performance Benchmarking and Investigation on Multiple HPC Architectures

WP236: Parallel I/O Performance Benchmarking and Investigation on Multiple HPC Architectures

Authors: Bryan Lawrence (Reading), Chris Maynard (Met Office, UK), Andy Turner (EPCC), Xu Guo (EPCC), Dominic Sloan-Murphy (EPCC)

HPC Tool/Technique: MPI-IO, HDF5, NetCDF, Lustre, GPFS, Panasas

Application: benchio

CoEs/Scientific Communities: ESiWACE

Parallel I/O performance plays a key role in many high-performance computing (HPC) applications. I/O bottlenecks are an important challenge to understand and, where possible, eliminate on both current, petascale resources and looking forward to exascale computing [18]. It is therefore necessary for research communities with high I/O requirements to understand the parallel I/O performance of existing HPC systems and applications to be suitably equipped to make informed plans for future procurements and software development projects. The results of this work are of particular relevance to the ESiWACE CoE, as the originators of this project, but, given the ubiquity of I/O in HPC domains, the findings will be of interest to most researchers and members of the European scientific computing community.

This work presents benchmarks for the write capabilities of the following HPC systems:

- ARCHER: the UK national supercomputing service, with a Cray Sonexion Lustre file system [19].
- COSMA: one of the DiRAC UK HPC resources, using a DDN implementation of the IBM GPFS file system [20].
- UK-RDF DAC: the Data Analytic Cluster attached to the UK Research Data Facility, also using DDN GPFS [21].
- JASMIN: a data analysis cluster delivered by the STFC, using the Panasas parallel file system [22].

We run benchio, a parallel benchmarking application which writes a three-dimensional distributed dataset to a single shared file. On all systems, we measure MPI-IO performance and, in select cases, compare this with HDF5 and NetCDF equivalent implementations.

We find a reasonable expectation is for approximately 50% of the theoretical system maximum bandwidth to be attainable in practice. Contention is shown to have a dramatic effect on performance. MPI-IO, HDF5 and NetCDF are found to scale similarly but the high-level libraries introduce a small amount of performance overhead. See Figure 23.

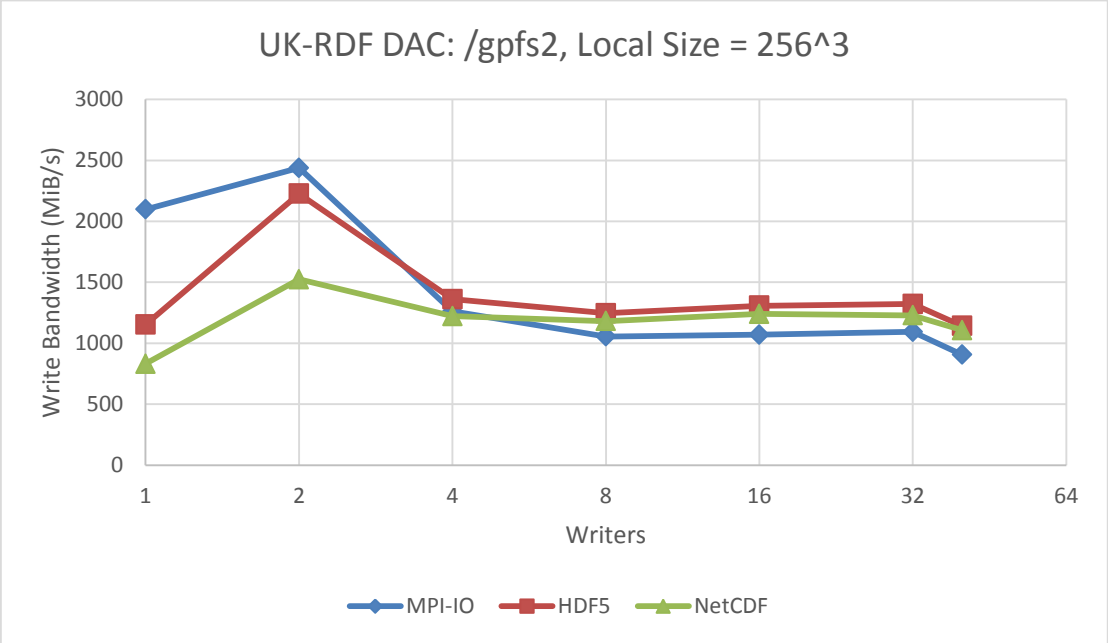

Figure 23: All backends bandwidth for UK-RDF DAC (File system: 4.4PB /gpfs2 mounted as /epsrc.)

For the Lustre file system, on a single shared file, maximum performance is found by maximising the stripe count and matching the individual stripe size to the magnitude of I/O operation performed. HDF5 is discovered to scale poorly on Lustre due to an unfavourable interaction with the H5Fclose() routine. See Figure 24.

For more details on this project, including a detailed description of the methodology and results, we refer the reader to PRACE Whitepaper WP236: Parallel I/O Performance Benchmarking and Investigation on Multiple HPC Architectures [3].

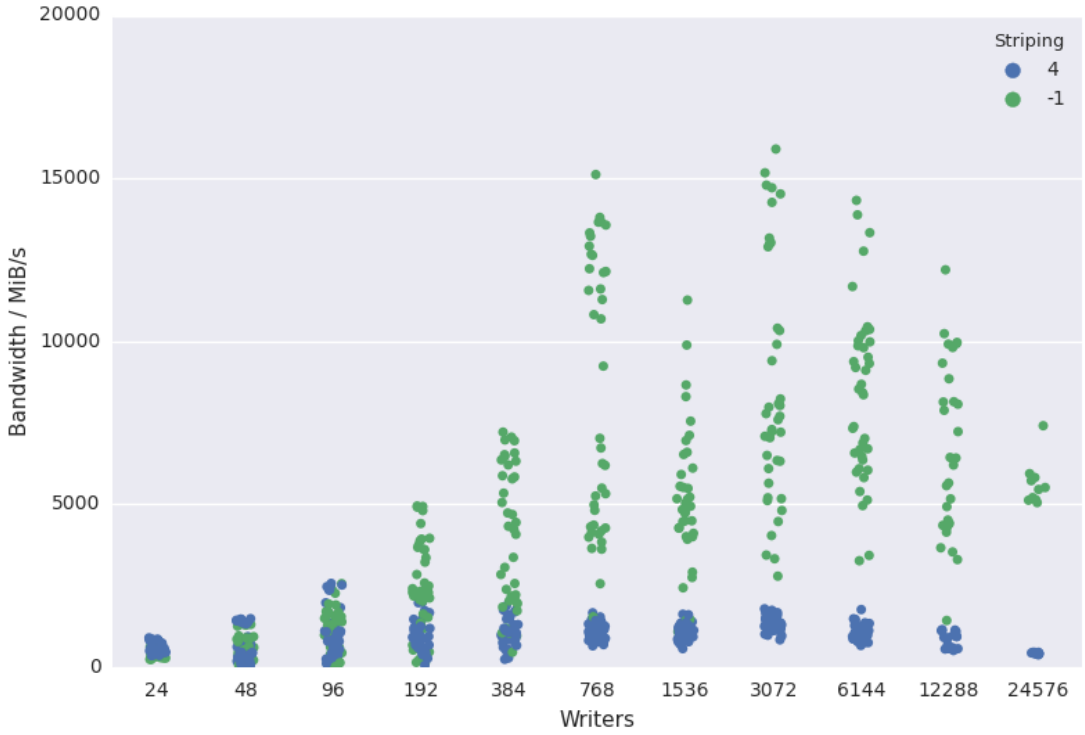

Figure 24: Results spread for ARCHER (Lustre) MPI-IO maximum striping (-1)

7 Summary

This deliverable reports on the 15 projects undertaken as part of the exploitation phase of PRACE-4IP T7.2a 'HPC Tools & Techniques', where the projects are aligned with five separate topics that we consider relevant to enable applications on current multi-petascale and future exascale systems, and where inspiration has been taken from the comprehensive survey of CoE HPC requirements and state-of-the-art HPC tools and techniques carried out during the first phase of T7.2a, which was reported on in D7.3 [1]. In this section, we summarise our findings separately by topic: energy efficiency, programming interfaces and standards, scalable libraries and algorithms, debuggers and profilers, and I/O management techniques. In-depth conclusions for each of the projects reported on in this deliverable can be found in associated whitepapers and so here we list only what we think are the most salient findings made during the exploitation phase.

Energy Efficient Computing

We have reported the exploitation of a prototype tool suite, known as READEX, being developed by the H2020 FET-HPC project (also named READEX). The work investigated a pre-alpha prototype of the READEX tool suite to identify existing dynamism in proxy applications as well as highly scalable European applications that regularly use large-scale PRACE resources to determine their tuning potential for improving energy efficiency. We also apply the prototype tool to a proxy version of the well-known molecular dynamics code, NAMD, which is of interest to the ECAM and MaX CoEs. Several of the CoEs are keenly focused on improving the energy efficiency of their applications running on extreme-scale systems, including MaX, ESiWACE and PoP (as well as the ESCAPE FET-HPC project) and we expect our investigations on the early-stage prototype to also be of interest to the wider community, particularly in the context of the PRACE PCP and future European extreme-scale systems. On the topic of energy efficiency, we have also reported on the development and

application of a low cost open system for high-resolution energy analysis of applications runnining on emerging accelerator platforms.

Programming Interfaces and Standards

As highlighted in the survey of CoE HPC requirements in PRACE 4IP D7.3 [1], currently, one programming model still dominates CoE applications more than any other, namely, Single Program Multiple Data (SPMD) message passing using MPI for internode communication. However, we feel that the clear lack of MPI 3.0 exploitation within the CoEs is a concern, which has motivated us investigate some of the modern features of MPI on large-scale systems through the exploitation of the recently developed Gabriel library. The Gabriel library allows for an efficient use of modern MPI 3.0 features, such as neighbourhood collectives within the NEMO application, which is in turn of interest to the ESiWACE CoE.

While combining MPI and OpenMP is still considered to be the hybrid programming method of choice, the recent advent and rapid adoption of many-core coprocessors/accelerators in the design of multi-petaflop systems must increasingly be considered in order to exploit the full potential of the compute hardware space on emerging European multi-petascale/future exascale systems. To date, the challenge of exploiting such heterogeneous systems has typically been met within the CoEs by augmenting the MPI/OpenMP hybrid model with an additional third model that targets the Single Instruction Multiple Thread (SIMT) architecture of GPUs thereby forcing the further extraction of hierarchical levels of parallelism in current European applications. In this deliverable we have reported on the exploitation of OpenMP 4, OpenCL and CUDA on many-core platforms, including the recently released Intel Knights Landing processor.

With a view to future exploitation of programming models within WP7, we feel that there is an important role for WP7 to play in carrying out investigative work on more novel prototype-level programming models and tools, as well as to increase awareness of such programming models within the CoEs. This would include models and tools emerging from the FET-HPC INTERTWINE and ALLSCALE projects, for example. With this ambition in mind, we are happy to report that WP7 has recently established an interaction with the INTERWINE project through an INTERWINE-organised Exascale workshop at the University of Manchester in Oct 2016 and aims to strengthen engagement with the FET-HPC community through continued bidirectional knowledge exchange at this level at upcoming workshops during the lifetime of PRACE-5IP.

Debuggers and Profilers

The exploitation of modern profiling and tracing tools is vital to understand application behaviour, performance bottlenecks and optimisation potentials in HPC applications. Despite their obvious benefits, such tools are still not that widely adopted within the HPC user community, including at the majority of the CoEs. With this challenge in mind, we have presented and compared the capabilities of four state-of-the-art scalable performance analysis tools, where the aim of our work has been to raise the general awareness for the benefits of using specialized performance analysis tools and to lower the threshold for potential users to getting started with such tools. We have also reported on how we have investigated the recently developed on-node cache-aware roofline model (CARM) profiling technique to better understand where bottlenecks exist for applications targeting the Intel Xeon Phi Knights Landing platform (in this case we focused on applications that are of interest to the ESiWACE CoE and FET-HPC ESCAPE project). Obtaining an accurate view on bottlenecks and their associations to hardware resources, including increasingly complex memory hierarchies is crucial for improving nodel-level performance, particularly on emerging many-core platforms. In this respect, the CARM as implemeneted in the state-of-the-art Intel Advisor tool provides a much more detailed analysis relative to the original roofline model,

particularly with regards to bounding expectations on realistically attainable performance on such platforms.

Scalable Libraries and Algorithms

We have reported on how we have focused on improving the scalability of two heavily used application codes within the EoCoE CoE by investigating the exploitation of new algebraic hybrid iterative/direct solvers within these codes as an alternative to algebraic multigrid preconditioned Krylov iterative solvers and preconditioners, which are known to scale poorly at large scale. We have also described applications for illustrating the modeling capabilities of the simulation software supported by EoCoE and assessing their parallel performance on the road to Exascale. As part of our focus on scalable libraries and algorithms we have also investigated a virtual processor mesh structure in order to bound and reduce the latency and intercommunication overhead for sparse linear algebra workloads that could be of interest to applications within the EoCoE and possibly other CoEs.

I/O Management Techniques

The increasing data needs of scientific and engineering applications mean that the problems associated with reading, writing, analysing, storing and sharing large amounts of data are becoming more relevant to the full range of CoEs. Unfortunately, it is still challenging to extract good performance from most current parallel I/O libraries for structured file formats. It is therefore necessary for the CoEs with high I/O requirements to understand the parallel I/O performance of existing HPC systems and applications to be suitably equipped to make informed plans for running on future extreme-scale systems. With this challeng in mind, we have worked closely with the ESiWACE CoE to present benchmarks for the write capabilities for a range of large-scale European HPC systems. Given the ubiquity of I/O in HPC domains, the findings are explectedto be of interest to most researchers and members of the European scientific computing community.

It should be kept in mind that while we have made the utmost effort to focus on tools, techniques and applications that are of direct relevance to the CoEs, the centres were only just being established at the time that WP7 work began. For this reason, many of the CoEs were not yet in a position to strongly identify their requirements around extreme-scale HPC. Nevertheless, we feel that all of the tools and techniques we have investigated during the exploitation phase should be of interest to the vast majority of the CoEs and the European HPC community more widely. Furthermore, the requirements of the CoEs' are expected to change over their lifetime and as such communication between PRACE and the CoEs should be maintained, if not strengthened, as we face into the Exascale frontier together. Finally, we should re-emphasise that, as we look to PRACE 5IP and beyond, WP7 will continue to be informed and inspired by the ongoing research across the various European and US exascale projects.