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

D7.3
Inventory of Exascale Tools and Techniques

Final

Version: 1.0
Author(s): Nicola Mc Donnell, ICHEC
Date: 14.03.2016

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: D7.3	
	Deliverable Nature: Report	
	Dissemination Level: PU*	**Contractual Date of Delivery:** 31 / March / 2016
		Actual Date of Delivery: 31 / March / 2016
	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: Inventory of Exascale Tools and Techniques**	
	ID: D7.3	
	Version: <1.0>	**Status:** ***Final***
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2010	
	File(s): D7.3.docx	
Authorship	**Written by:**	Nicola Mc Donnell, ICHEC
	Contributors:	Jan Christian Meyer, NTNU Stephane Lanteri, Inria Luc Giraud, Inria Cevdet Aykanat, Bilkent Kadir Akbudak, Bilkent Reha Oguz Selvitopi, Bilkent Mark Bull, EPCC Thomas Ponweiser, RISC Software/JKU Ruairi Short, ICHEC Emma Hogan, ICHEC Michael Lysaght, ICHEC
	Reviewed by:	Enver Ozdemir, UHEM Thomas Eickermann, FZJ
	Approved by:	MB/TB

Document Status Sheet

Version	**Date**	**Status**	**Comments**
0.1	05/February/2016	Draft	Initial version
0.2	08/March/2016	Draft	First Complete Draft
0.3	09/March/2016	Draft	Second Complete Draft

0.4	09/March/2016	Draft	For Internal Review
0.5	14/March/2016	Draft	Post Internal Review updates.
1.0	14/March/2016	Final	Final version

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, Centres of Excellence, Exascale, MPI, OpenMP, GPGPU, I/O Management.

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[1] CRESTA - https://www.cresta.org
[2] DEEP Project - http://www.deep-project.eu
[3] Mont Blanc Project - https://www.montblanc-project.eu
[4] NUMEXAS - http://www.numexas.eu
[5] EXA2CT - http://www.exa2ct.eu
[6] EPiGRAM - http://www.epigram-project.eu
[7] PRACE-3IP D7.2.1 Deliverable - http://www.prace-ri.eu/IMG/pdf/d7.2.1.pdf
[8] GROMACS - http://www.gromacs.org
[9] LAMPS - http://lammps.sandia.gov/
[10] NAMD - http://www.ks.uiuc.edu/Research/namd/
[11] ESPResSo - http://espressomd.org/, http://www.espresso-pp.de/
[12] Nvidia - http://www.nvidia.com/
[13] AMD - http://www.amd.com/en-gb
[14] HADDOCK - http://www.bonvinlab.org/software/haddock2.2
[15] CPMD - http://www.cpmd.org/
[16] NetLogo - https://ccl.northwestern.edu/netlogo/
[17] LevelSpace - http://ccl.northwestern.edu/rp/levelspace
[18] Pandora - http://xrubio.github.io/pandora/
[19] Repast - http://repast.sourceforge.net/
[20] Mason - http://cs.gmu.edu/~eclab/projects/mason/
[21] Hartree Centre - http://www.hartree.stfc.ac.uk/hartree/
[22] PETSc - https://www.mcs.anl.gov/petsc/
[23] Hypre - http://acts.nersc.gov/hypre/
[24] BSC-CTM – http://www.bsc.es/earth-sciences/nmmbbsc-project
[25] Dr. Hook – https://software.ecmwf.int/wiki/download/attachments/19661682/drhook.pdf?version=1&modificationDate=1443002579511&api=v2
[26] Optimising CP2K for the Intel Xeon Phi - http://www.prace-ri.eu/IMG/pdf/wp140.pdf

List of Acronyms and Abbreviations

AiiDA	Automated Interactive Infrastructure and Database for Computational Science
AMD APU	The AMD Accelerated Processing Unit (APU), formerly known as Fusion, is marketing term for a series of 64-bit microprocessors from AMD designed to act as a CPU and graphics accelerator (GPU) on a single chip.
API	Application Programming Interface
ARM	ARM, originally Acorn RISC Machine, later Advanced RISC Machine, is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments.
Big Data	Data sets that are so large or complex that traditional data processing applications are inadequate
BioExcel	Centre of Excellence for Biomolecular Research
Boost	A set of C++ libraries for linear algebra, pseudorandom number generation, multithreading, image processing, regular expressions, and unit testing.
BSC	Barcelona Supercomputing Center, Spain
CECAM	Centre Européen de Calcul Atomique et Moléculaire
Chapel	Cascade High Productivity Language, a parallel programming language developed by Cray
Charm++	A machine independent parallel programming system
Cilk Plus	A multicore and vector processing tool from Intel®
CMCC	Fondazione Centro Euro-Mediterraneo sui Cambiamenti Climatici, Italy
CoE	Centre of Excellence
CoeGSS	Center of Excellence for Global Systems Science
CP2K	A program to perform atomistic and molecular simulations of solid state, liquid, molecular and biological systems
CPMD	A computational chemistry software package for Car–Parrinello Molecular Dynamics (CPMD)
CPU	Central Processing Unit
CrayPat	A performance analysis tool offered by Cray for the XC platform
CRESTA	FP7 Exascale project: Collaborative Research into Exascale Systemware, Tools and Applications
CUDA	Compute Unified Device Architecture, a parallel computing platform and GPGPU application programming interface created by NVIDIA.
Darshan	A scalable HPC I/O characterization tool from LRZ.
DDT	Allinea DDT (Distributed Debugging Tool) is a commercial C, C++ and Fortran 90 debugger for parallel MPI and threaded programs
DEEP	FP7 Exascale project: Dynamical Exascale Entry Platform
DEEPer	Follow-on project to DEEP
Dimemas	Dimemas is an MPI performance analysis tool.
DKRZ	German Climate Computing Center (Deutsches Klimarechenzentrum)
DNA	Deoxyribonucleic acid

Dr.Hook	A simple, low-overhead instrumentation tool, which allows you to keep track of dynamic calling tree of a program and print it in the event of failure.
DSL	Domain Specific Language
DYNAMICO	Dynamico Atmospheric Dynamical Core Model, a hybrid MPI / OpenMP parallelized simulation code
E-CAM	An e-infrastructure for software, training and consultancy in simulation and modelling
EC-EARTH	An Earth System Model for climate simulation
ECMWF	European Centre for Medium-Range Weather Forecasts, UK
ELPA	Eigenvalue SoLvers for Petaflop-Applications
EoCoE	Energy oriented Centre of Excellence for computer applications
ESiWACE	Excellence in SImulation of Weather and Climate in Europe
ESPResSo	Extensible Simulation Package for Research on Soft matter
ESPResSo++	An extensible parallel simulation software for soft matter research. It has a common root with ESPResSo, but is free, open-source software published under the GNU General Public License (GPL).
ETSF	European Theoretical Spectroscopy Facility
EXA2CT	FP7 Exascale project: Exascale Algorithms and Advanced Computational Techniques
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
FFT	Fast Fourier Transform
FFTW	Fastest Fourier Transform in the West, a C FFT library; includes real-complex, multidimensional, and parallel transforms
FFTW3	FFTW version 3
FHI-aims	Fritz Haber Institute ab initio molecular simulations package
FLEUR	A feature-full, freely available FLAPW (full potential linearised augmented planewave) code, based on density-functional theory
Fortran	A general-purpose, imperative programming language that is especially suited to numeric computation and scientific computing
Fortran Co-arrays	A parallel processing extension to Fortran
FP7	The 7th Framework Programme funded European Research and Technological Development from 2007 until 2013
FTI	Fault Tolerance Interface
Glean	A gene predictor combiner tool
GPGPU	General-purpose computing on graphics processing units
GPL	General Public License
GPU	Graphics Processing Unit
GRIB	A concise data format commonly used in meteorology to store historical and forecast weather data
GRIB2	GRIB API version 2
GROMACS	GROningen MAchine for Chemical Simulations, a molecular dynamics package primarily designed for simulations of proteins, lipids and nucleic acids.

GSS	Global Systems Science
HADDOCK	High Ambiguity Driven biomolecular DOCKing, a Software package for integrative modelling of biomolecular complexes
Hadoop	An open-source software framework from Apache written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware
HDF5	A data model, library, and file format for storing and managing data
HPC	High Performance Computing
HPCToolkit	An open-source suite of tools for profile-based performance analysis of applications developed at Argonne National Laboratory
HPDA	High Performance Data Analysis
HTC	High Throughput Computing
I/O	Input output, refers to reading and writing files.
ICT	Information and Communications Technology
IFS	Integrated Forecast System, an operational global meteorological forecasting model developed by ECMWF
iotk	A toolkit used by the Quantum Espresso package to write the file in a XML format
IPSL	Institut Pierre-Simon Laplace, France
IPSLCM6	A full earth system model including atmosphere-land-ocean-sea ice model, carbon cycle, stratospheric and tropospheric aerosols chemistry
ITAC	Intel Trace Analyzer and Collector is a graphical tool for understanding MPI application behavior
KTH	Kungliga Tekniska högskolan (Swedish University) a.k.a. KTH Royal Institute of Technology
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
LevelSpace	A technical platform for running an arbitrary number of NetLogo models concurrently, and have them communicate. LevelSpace is a NetLogo extension, and will run Mac, Windows, and Linux.
MAP	A C/C++ profiler and Fortran profiler for high performance Linux code from Allinea
MaPHyS	Massively Parallel Hybrid Solver
MapReduce	A programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster.
Mason	A fast discrete-event multiagent simulation library core in Java, designed to be the foundation for large custom-purpose Java simulations, and to provide more than enough functionality for many lightweight simulation needs. It contains both a model library and an optional suite of visualization tools in 2D and 3D.
MaX	Materials design at the eXascale
Met Office	The UK's national weather service (officially the Meteorological Office until 2000).
MKL	Math Kernel Library
Mont Blanc	FP7 Exascale project: European Approach towards Energy Efficient High Performance
Mont Blanc 2	Follow-on project to Mont Blanc

MPI	Message Passing Interface, a standardised and portable message-passing system designed by a group of researchers from academia and industry to function on a wide variety of parallel computers.
MPI-IO	A library which provides parallel I/O support
MPI-M	Max-Planck-Institute for Meteorology, Germany
MPMD	Multi-Program Multi-Data
MUMPS	MUltifrontal Massively Parallel Sparse direct Solver
N-Body	A simulation of a dynamical system of particles, usually under the influence of physical forces
NAMD	Nanoscale Molecular Dynamics program
NEMO	Nucleus for European Modelling of the Ocean, a modeling framework for oceanographic research, operational oceanography seasonal forecast and climate studies
NetCDF	A set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.
NetLogo	A multi-agent programmable modeling environment developed at the Center for Connected Learning and Computer-Based Modeling (CCL) in Northwestern
NMMB	Nonhydrostatic Multiscale Meteorological Model on the B grid
NoMaD	The Novel Materials Discovery Laboratory
NorESM	Norwegian Earth System Model
NUMEXAS	FP7 Exascale project: Numerical Methods and Tools for Key Exascale Computing Challenges in Engineering and Applied Sciences
NVIDIA	An American technology company who specialise in graphics cards and GPGPUs.
OmpSs	A Programming model extending OpenMP with directives to support asynchronous parallelism and heterogeneity.
OpenACC	Open Accelerators, an Application Program Interface describing a collection of compiler directives to specify loops and regions of code in standard C, C++ and Fortran to be offloaded from a host CPU to an attached accelerator.
OpenCL	Open Computing Language, a framework for writing programs for heterogeneous platforms.
OpenMP	OpenMP (Open Multi-Processing) is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran
Pandora	A framework designed to create, execute and analyse agent-based models in high-performance computing environments.
Paraver	A trace performance analyzer tool.
PATC	PRACE Advanced Training Center (providing education and training opportunities for computational scientists in Europe)
PDI	Parallel Data Interface
Performance Report	A tool from Allinea which generates a single-page report on HPC program performance
PGAS	Partitioned Global Address Space, a parallel programming model
pHDF5	Parallel HDF5 (PHDF5) provides users with an option to have multiple processes perform I/O to an HDF5 file

pNetCDF	A library providing high-performance parallel I/O
PoC	Point of Contact
POP	Performance Optimisation and Productivity
PowerPC	A RISC instruction set architecture created by the 1991 Apple–IBM–Motorola alliance, known as AIM
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PRACE-3IP	PRACE Third Implementation Phase project
PRACE-4IP	PRACE Fourth Implementation Phase project
Psi-k	A network of researchers working on the advancement of first-principles computational materials science
Python	An interpreted programming language
Quantum ESPRESSO	A code for nanoscale electronic-structure calculations and materials modeling
Repast	A family of advanced, free, and open source agent-based modeling and simulation platforms
RISC	Reduced instruction set computing
ScaLAPACK	A library of high-performance linear algebra routines for parallel distributed memory machines
Scalasca	A tool to analyzing runtime behaviour and performance of parallel programs.
Score-P	A measurement infrastructure and easy-to-use tool suite for profiling, event tracing, and online analysis of HPC applications.
SIESTA	Spanish Initiative for Electronic Simulations with Thousands of Atoms
Simdemics	A commercial pandemic planning and response framework.
SIONLib	A scalable I/O library for the parallel access to task-local files
SME	Small and Medium-sized Enterprises, businesses whose personnel numbers fall below certain limits
SMHI	Swedish Meteorological and Hydrological Institute
SNIC	Swedish National Infrastructure for Computing
Spark	A fast and general engine for large-scale data processing from Apache
TAU	Tuning and Analysis Utilities, a portable profiling and tracing toolkit for performance analysis of parallel programs
TBB	Threading Building Blocks, a C++ template library developed by Intel for parallel programming on multi-core processors
TNG	A portable I/O library
TotalView	A serial and parallel debugger from Rogue Wave
UPC	Unified Parallel C, an extension of the C programming language designed for high-performance computing on large-scale parallel machines either with common global address space or with distributed memory.
VAMPIR	A trace analyser for sequential or parallel programs
VTune	Intel® VTune Amplifier provides is a tool to provide performance insight into CPU & GPU performance, threading performance & scalability, bandwidth.
WP	Work Package
X10	A programming language for parallel computing using the PGAS model.

x86	A family of instruction set architectures based on the Intel 8086 CPU
Xeon Phi	Intel accelerator technology
XIOS	XML – IO – SERVER: Library dedicated to IO management of climate code
Yambo	A FORTRAN/C code for Many-Body calculations in solid state and molecular physics
EPiGRAM	FP7 Exascale project: Exascale Programming Models
MAGMA	A simulation software for virtual experimentation and optimization in all metalcasting applications
US	United States of America

Executive Summary

The objective of the Partnership for Advanced Computing in Europe Fourth Implementation Phase project (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. Task 7.2, 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. The purpose of the analysis phase is to understand the needs of the HPC community and in doing so to develop links with the recently established Applications Centres of Excellence (CoEs) and build on existing links with European Exascale projects. A questionnaire was designed, which was circulated to through a Point of Contact (PoC) for each CoE. The analysed findings are the subject of this document. It is organised in two main sections, Section 2 :‘Review across questionnaires’, and Section 3 :‘Review across CoEs’.

The participating CoEs are:

- 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.

In Section 2, we summarise our findings separately by topic: programming interfaces and standards, debuggers and profilers, scalable libraries and algorithms and I/O management techniques, European Exascale projects, and general questions, where we focus on the most salient points to consider for the subsequent exploitation phase of Task 7.2a.

Programming interfaces and standards

All the CoEs are using MPI (Message Passing Interface) and OpenMP (Open Multi-Processing), and all are using, or have an interest in, one or more of the GPU (Graphics Processing Unit) programming interfaces.

Debuggers and Profilers

A wide variety of tools are used, 17 in total, and the regularity with which they are used varies greatly across the centres. However, all centres still use manual or console based debugging and profiling to a high degree. Scalasca, a tool for analysing the runtime behaviour and performance of parallel programs, is also used by all centres, though with less regularity. Most tools were seen as being easy to use with the exception of Extrae/Paraver, trace performance analyzer tools, and Intel VTune, a performance insight tool.

Scalable Libraries and Algorithms

One library that has been identified as being commonly exploited by several CoEs is the FFTW library, a well-known FFT (Fast Fourier Transform) library. It is clear that classical HPC library domains (dense and sparse linear algebra, FFTs) remain important for a significant number of CoEs.

I/O Management Techniques

For the CoEs focused on materials science, such as MaX, E-CAM and BioEXCEL, Input/Output (I/O) is not typically found to be a major performance bottleneck on current petascale systems. Several CoEs have highlighted the desire for a common standard that would allow for the exchange of data between different materials science codes within the CoE and have identified HDF5, a data model, library and file format for storing and managing data, as a possible candidate due to its popularity. I/O Management is deemed to be the key to success to those CoEs with a Big Data (data sets that are so large or complex that traditional data processing applications are inadequate) focus.

European Exascale Projects

Many of the CoEs are aware of, or have partners contributing to, the larger European Exascale projects (CRESTA[1], DEEP/DEEP-er[2], Mont Blanc/Mont Blanc 2[3]), but there are fewer connections to the smaller projects (NUMEXAS[4], EXA2CT[5], EPiGRAM[6]). Only E-CAM has no connections to any Exascale projects.

General Questions

The most requested system performance characteristics are floating-point performance, high speed memory, and low latency networking. There is a strong interest in accelerator technology. The main areas of interest for PRACE interactions are computer expertise, training and documentation; there is a smaller need for domain expertise and general assistance.

The notable findings of Section 3 are outlined below.

- **BioExcel** may require from PRACE general assistance in performing large-scale benchmarks as well as expertise in the area of development, porting and tuning of codes for some specific platforms and temporally and spatially align their training events.
- PRACE is expected to support **CoeGSS** in terms of general assistance, training, online documentation, domain related expertise, and code development/porting/tuning.
- The exploitation of HPC systems at the **E-CAM** CoE will focus mainly on developing testing and benchmarking materials science codes, modules and workflows on supercomputers/HPC clusters, and their exploitation of accelerator hardware.
- **EoCoE** supports a large number of application codes, 23 in total. Several of the application codes considered are concerned with the solution of large sparse linear systems at some point. The sparse linear solver used for achieving this task greatly influences the overall scalability of the applications.
- **ESiWACE** signalled a strong need for general assistance and computer expertise in code development, porting and tuning, and a moderate need for training. There is not much experience in Xeon Phi (Intel's many core platform) and GPU profiling within ESiWACE yet, which could be an excellent opportunity for assistance from the side of PRACE-4IP Work Package7 (WP7).
- Due to the diversity of the actions planned within the **MaX** CoE, there where will be requirements in general processing, the exploitation of accelerator hardware, storage and I/O solutions as well as HPC software and tools stacks.
- **NoMaD** was unable to complete many questions as they are still at an early stage in defining their requirements. This highlights the need to re-engage with the CoE periodically to reassess the HPC requirements.
- **POP** needs fast, but short duration access to PRACE resources for approximately 10% of their 150 selected codes. POP is also interested in the questionnaire responses from the other CoEs to the 'Debuggers and Profilers' section.

In conclusion, it should be emphasised that all of the exploitation work proposed to be carried out during the next phase of Task 7.2a will continue to be informed and inspired by the on-going research across the various European and US (United States of America) exascale projects. 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, as we face into the exascale frontier.

Acknowledgements

The authors would like to acknowledge and thank the Centres of Excellence for their cooperation with and contribution to this deliverable. In particular, the authors thank the points of contact, those gave their time to complete the questionnaires and the wider teams who contributed.

1 Introduction

Purpose of the document

The objective of the Partnership for Advanced Computing in Europe Fourth Implementation Phase project (PRACE-4IP) Work Package 7 (WP7) 'Application Enabling and Support' is to provide enabling support for HPC applications codes which are important for European researchers and industry, to ensure that these applications can effectively exploit multi-petaflop systems and future PRACE Exascale systems. WP7 also supports European HPC (High Performance Computing) research communities through the provision of best practice guides, benchmarks, and example parallel codes with technical results disseminated in white papers. Task 7.2, 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 phase, and an exploitation phase. The purpose of the analysis phase is to examine the needs of the HPC community and in doing so to develop links with the emerging applications Centres of Excellence (CoEs) and build on existing links with European Exascale projects. The central goal of this task will be to provide cross-disciplinary support for the CoEs, although the outputs will also be useful within PRACE and for other European HPC users. To understand what support they need a questionnaire was designed which is in Annex. We identified a Point of Contact (PoC) for each CoE. These PoC are listed in Table 1.

Application Centres of Excellence	Acronym	Points of Contact
Centre of Excellence for Biomolecular Research	BioExcel	Rossen Apostolov, KTH
Center of Excellence for Global Systems Science	CoeGSS	Bastian Koller, HLRS
An e-Infrastructure for software training and consultancy in simulation and modelling	E-CAM	Michael Lysaght, ICHEC
Energy oriented Centre of Excellence for computer applications	EoCoE	Isabelle Dupays, IDRIS Dimitri Lecas, IDRIS
Excellence in Simulation of Weather and Climate in Europe	ESiWACE	John Donners, SURFsara
Materials Design@eXascale	MaX	Andrew Emerson, CINECA
The Novel Materials Discovery Laboratory	NoMaD	Hermann Lederer, RZG
Performance Optimisation and Productivity	POP	Judit Gimenez, BSC

Table 1 List of the Points of Contact for each of the Centres of Excellence

The PoCs, on receipt of the questionnaires, arranged for them to be completed and returned to the Task 7.2a team. These questionnaires were subsequently analysed and the findings are the subject of this document. The questionnaire consists of questions in the following six areas:

- General questions on codes & systems,
- Programming interfaces and standards,
- Debuggers and Profilers,
- Scalable Libraries and Algorithms,

- Input/Output (I/O) Management Techniques, and
- European Exascale Projects.

Structure of the document

The document has two main sections and one Annex. In the first section, 'Review across questionnaires' Section 2, the questionnaire responses to each of the 6 groups of questions are examined. The second section, 'Review across CoEs' Section 3, examines each questionnaire from the perspective of the CoEs. Each section begins with a short description of the CoE and then the specific needs of the CoE with respect to PRACE and exascale are explored. In the Annex in Section 5, we include the original questionnaire sent and the responses received for reference.

Intended Audience

The objective in preparing this questionnaire is to determine and document the needs of the CoEs. It is targeted at PRACE partners who will be involved in the exploitation phase of Task 7.2a and the CoEs they will support. In particular, the POP CoE is interested in the questionnaire responses by the other CoEs to the 'Debuggers and Profilers' section. It is also hoped that the report here will be of interest to European HPC users and anyone interested in the HPC aspects of the newly formed CoEs, such as the Exascale projects.

It is anticipated that a reader may be interested in only one section of this document, for example, 'Debuggers and Profilers', or just one CoE. Consequently, the sections are written to be standalone, though this may have the undesirable side-effect of some repetition for someone reading the document as a whole.

2 Review across questionnaire sections

In this section we examine the questionnaire responses across each of the six sections. These sections are:

- General questions on codes & systems;
- Programming interfaces and standards;
- Debuggers and Profilers;
- Scalable Libraries and Algorithms;
- I/O Management Techniques, and
- European Exascale Projects.

2.1 General questions (codes & systems)

The general questions can be classified into three groups: description of the CoE and its objectives, system requirements, and application support. The following subsections summarise the responses in each category (see Annex 5 for the individual questions).

Questions 1-3: Description of the CoE

Questions 1 and 2 invite free-text descriptions of CoE mandates and objectives. The responses indicate three broad functions, which each CoE aims to fill one or more of:

Competency centre

Most centres aim to work as a transversal meeting ground for interdisciplinary research, by involving researchers from different disciplines, as well as offering consultancy services and training to external clients.

Research and development

Many centres aim to do research and development work on hardware and software infrastructure, to better enable their application to particular scientific domains.

Data library

Some centres aim to aggregate and curate a collection of scientific data for use with applications within their domain.

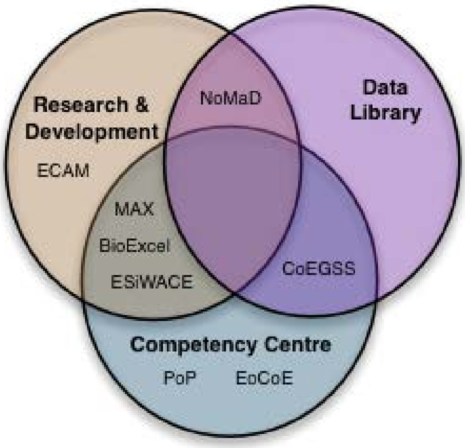

Figure 1 Classifying the responses with respect to these categories results

The scientific domains of the centres naturally show few commonalities, but it is noteworthy that the work of three centres (ECAM, MaX and NoMaD) all name material sciences as an important focus area.

Responses to question 3 are mostly homogeneous, suggesting almost universal focus on impacting the scientific domain through research publications and software development. Notable exceptions are that ECAM and MaX identify a goal of having impact on industrial/commercial applications, and that POP focuses on enabling performance improvements and productivity independent of specific application areas.

Questions 4-7, 10-11: Systems

In this group, POP responded only to question 5, as its requirements will be governed by application areas that remain unknown at the time of writing.

Question 4 addresses the class of hardware platform that each CoE expects to utilise. A clear majority of 6 in 7 respondents identify the need for conventional, highly parallel HPC systems, with NoMaD being the only exception. Half of the centres identify needs for the remaining system classes, i.e. data analytics clusters, large shared memory systems, and data sharing/cloud frameworks.

Question 5 addresses expected usage patterns. Expectations divide almost evenly among continuous (production) and sporadic requirements with 5 and 4 centres respectively, and an even 4 for both short and long term requirements. BioExcel and ESiWACE expect high-intensity use cases that require rapid access to a large fraction of available resources.

Question 6 addresses which system classes a CoE expects to utilise. Like question 5, the distribution between alternatives is very homogeneous, revealing a need for both capacity and capability installations. Half of the centres report an interest in interactive use in addition to batch processing, which should be noted because it is presently not the most widespread mode of interaction in HPC contexts.

Question 7 addresses performance requirements with respect to specific hardware subsystems. All responding centres report a need for processing floating-point data, and CoeGSS, ESiWACE, MaX and NoMaD additionally require integer processing. 6 in 7 require high-performance memory and low latency networking. In the context of exascale-enabling technology, it is interesting to note that all respondents except CoeGSS also report a need for accelerator units.

Question 10 requests known peak performance figures of the simulation software, which centers intend to make use of. This question only attracted a response from ESiWACE, noting that the numerical figure depends greatly on the characteristics of the simulation. Thus, question 10 unfortunately gives little information of any utility.

Question 11 asks which degree of floating-point precision the centres' simulations make use of. This is relevant to accelerator units, because the computing performance of many such technologies is vastly superior for single precision calculations. Those who responded to this question (BioExcel, CoeGSS, ECAM, ESiWACE and MaX) unilaterally require some double precision calculations; BioExcel, ESiWACE and NoMaD report that they also use single precision.

Questions 8, 9 and 12: Application support

Question 8 asks which types of support each CoE will rely on from PRACE. The 'Training' and 'Computer Expertise' options are most requested, with 6 centres signalling needs. 'Online documentation' follows with 5 interested respondents, while 4 centres are expecting general assistance and domain expertise.

Question 9 requests the names of widely used software packages which are already known to scale beyond 200,000 cpu cores. This gave rise to a list of 24 programs, listed below.

- GROMACS
- Quantum ESPRESSO
- Haddock
- Pandora
- REPAST
- MASON
- LAMMPS
- MPI-M
- Python
- Cylc
- Rose
- Dynamico
- EC-Earth
- NMMB/BS-CTM
- IFS
- NEMO
- CP2K
- FHI_aims
- Fleur
- SIESTA
- Yambo
- Aiicu
- Simdemics
- GLEAN

It is worth noting that GROMACS, a molecular dynamics package, and Quantum ESPRESSO, a code for nanoscale electronic-structure calculations and materials modelling, both were listed by three respondents. Moreover, GROMACS, Quantum ESPRESSO, NEMO, an oceanographic modelling framework, and CP2K, a program for molecular simulations, are featured in the PRACE Unified European Benchmark Suite maintained by WP7. Simdemics, a pandemic planning and response modelling framework, and GLEAN, is a gene predictor combiner tool, are the only commercial application packages.

Question 12 requests a set of exascale-relevant performance metrics to be ranked in terms of importance to the CoE. This permits a weighted sum to be drawn from the responses, which makes it clear that inter-node and intra-node performance characteristics both are highly valued by all respondents, and in sum, approximately twice as important as the next criterion. Resilience and energy efficiency are highly rated by EsiWACE and MaX, and moderately by ECAM.

Summary of the responses

In summary, the primary objectives of the centres are to publish research results and contribute to software and tools. Continuous availability of computational resources to achieve long-term goals is slightly favoured, but there is also a marked need for access patterns with short time frames and high intensity. Similarly, capability systems are slightly favoured over capacity systems, but there is a prominent need for both. It is noteworthy that half of the centres wish for interactive use in addition to batch scheduling. The most requested system performance characteristics are floating-point performance, high speed memory, and low latency networking. There is a strong interest in accelerator technology. The main areas of interest for PRACE interactions are computer expertise, training and documentation; there is a smaller need for domain expertise and general assistance.

It is clear that all the CoEs are working in areas that will wish to make use of Exascale HPC systems. In the short to medium term, future hardware architectures seem likely to deliver improved floating point and memory performance, but the prospects for lower latency networks are less clear.

2.2 Programming interfaces and standards

Of the eight CoEs, EoCoE and NoMaD did not provide any responses to this section. POP, which is developing tools rather than applications, interpreted questions 13 and 15 in terms of which programming Application Programming Interfaces (APIs) and hardware platforms are supported in their tools (and did not answer question 14). ESiWACE gave individual responses for every application code, which have been condensed into a single response in the following analysis.

Table 2 summarises the responses to question 13 “Which of the following programming tools do you exploit in the development of your codes?, where a ‘X’ denotes current use and ‘(X)’denotes interest or future plans.

	BioExcel	CoeGSS	E-CAM	ESiWACE	MaX	POP
MPI	X	X	X	X	X	X
Fortran Co-arrays				(X)		
OpenMP	X	X	X	X	X	X
OpenCL	X		(X)			X
CUDA	X	(X)	(X)	X	X	X
OpenACC		(X)	(X)	X	(X)	
OmpSs	X			X	(X)	X
Spark		(X)				
Other		Hadoop, Storm		Multi-executable		SHMEM, pthreads

Table 2 Summary of responses to question 13

All the CoEs are using Message Passing Interface (MPI) and Open Multi-Processing (OpenMP), and all are using (or have an interest in) one or more of the Graphics Processing Unit (GPU) programming interfaces. Apart from some plans in ESiWACE for Fortran Co-arrays, a parallel processing extension to Fortran, there is no use of Partitioned Global Address Space (PGAS) models, nor of Threading Building Blocks (TBB) and Cilk Plus, A multicore and vector processing tool from Intel. Only one CoE, CoeGGS, is using MapReduce, a programming model for processing large datasets in parallel.

It seems probable that as we approach exascale many applications are likely to use two or three different programming models together: typically MPI + OpenMP/OmpSs; OmpSs is an extension to OpenMP to support asynchronous parallelism and heterogeneity. It will therefore be crucial to have good interoperability between different models operating inside the same application code - aspects of this have been considered in most of the Framework Programme 7 (FP7) European Exascale projects, and there are continuing efforts in the H2020 FET-HPC projects, the H2020 Future and Emerging Technologies (FET) Proactive call for HPC.

Table 3 summarises the responses to question 14 “How important are the following features of a programming tool to you?”

	BioExcel	**CoeGSS**	**E-CAM**	**ESiWACE**	**MaX**
Productivity	Very Important	Very Important	Somewhat Important	Very Important	Very Important
Open Standard	Very Important	Very Important	Very Important	Somewhat & Very Important	Very Important
Sustainability	Very Important	Very Important	Very Important	Somewhat & Very Important	Very Important
Portability	Very Important	Very Important	Very Important	Somewhat & Very Important	Very Important

Table 3 Summary of responses to question 14

All responding CoEs considered almost all the features as very important. It is interesting to note that there is a desire for open standards and portability even at the highest levels of application scalability: bespoke and/or proprietary APIs may not gain much popularity in these application communities.

Table 4 summarises the responses to question 15 “Which of the following platforms are your codes currently targeting or planning to target?” where a cross denotes current use and a cross in brackets denotes interest or future plans.

	BioExcel	**CoeGSS**	**E-CAM**	**ESiWACE**	**MaX**	**POP**
x86	X	X	X	X	X	X
Power	X	X	X	X	X	X
Xeon Phi	X	X		X	X	X
NVIDIA GPGPU	X	X	(X)	X	X	X
AMD GPGPU/APU	X	X			X	X
Other	ARM	ARM		NEC SX		Fujitsu, Cray

Table 4 Summary of responses to question 15

This indicates that all the responding CoEs are using (or are interested in using) most or all of the commonly available hardware platforms, and are actively engaged with future hardware technologies.

2.3 Debuggers and Profilers

The questions relating to Debuggers and Profilers were answered by 5 of 8 CoEs. It was not answered by EoCoE or NoMaD. POP indicated their status as developers of Scalasca, a tool for analyzing the runtime behavior and performance of parallel programs, and Paraver, a trace performance analyzer tool, and thus felt that their responses would be atypical. They are however interested in the answers to the questions provided by the other centres. CoeGSS stated that only manual console output is used for debugging and profiling as they have not found tools to be available in Python, an interpreted programming language. However it was noted that the cProfile feature of High Performance Python will become important.

Table 5 summarises the responses to question 16 "For analysing the performance of your application codes, which of the following tools do you use and how often?" where a cross denotes current use and a cross in brackets denotes interest or future plans.

	BioExcel	E-CAM	ESiWACE	MaX
Manual/Console	X	X	X	X
Scalasca	X	X	X	X
TAU		X	X	X
HPCToolkit		X	X	X
Extrae/Paraver	(X)		X	
Intel VTune			X	X
Allinea - DDT	X		X	
Other	Valgrind		Dr Hook, Gstats, MAP, Allinea perf report, MPI perf snapshot VAMPIR, ITAC, Totalview	

Table 5 Summary of responses to question 16

A wide variety of tools are used, 17 in total, and the regularity with which they are used varies greatly across the centres. However, all centres still use manual or console based debugging and profiling to a high degree. Scalasca is also used by all centres, though with less regularity.

The manner in which the tools are used for analysis is in line with the intended use. For example, Allinea Distributed Debugging Tool (DDT) is used for debugging and Extrae/Paraver is used for various types of profiling; Extrae is a package to generate Paraver trace-files for a post-mortem analysis. It is noted that only BioExcel and Max use the tools for profiling of Xeon Phi / GPU applications; Xeon Phi is Intel's accelerator technology.

In general, the CoEs have not been limited by the scalability of the tools, but more by the resources available. The targeted scale is often quoted to be higher than the typical usage scale, though reasons for this are not given. For the most part it has simply not been tested. The exception is ESiWACE, where issues such as licensing of TotalView, a serial and parallel debugger from Rogue Wave, and the overhead associated with Scalasca or VAMPIR, a trace analyser for sequential or parallel programs, limiting the number of processes which can be used.

With respect to rating the tools, manual or console debugging scored highly in all centres. This is largely due to the ease of use and the portability. However, this will not be an effective solution on exascale systems as it does not scale well and does not provide as clear a view of the barriers to achieving performance as other methods. Most tools were seen as being easy to use with the exception of Extrae/Paraver, trace performance analyzer tools, and Intel VTune, a performance insight tool. For example, Paraver was quoted as hard to install and configure by ESiWACE. All tools were seen as being reliable, and most scale acceptably or very well.

Among the tools listed in Table 5, TAU (Tuning and Analysis Utilities) is a portable profiling and tracing toolkit for performance analysis of parallel programs; Scalasca, VAMPIR, TotalView and DDT were identified in PRACE Third Implementation Phase project (PRACE-3IP) deliverable D7.2.1[7] as demonstrated to run on systems in excess of 100,000 cores. Additionally, it is mentioned that Extrae/Paraver has been installed on Tier-0 systems, and has proven useful to performance analysis studies within the European exascale project DEEP, the FP7 Exascale project 'Dynamical Exascale Entry Platform'.

2.4 Scalable Libraries and Algorithms

The computational problems addressed in the scalable libraries and algorithms are grouped under the following topics:

- Dense linear algebra;
- Sparse linear algebra;
- Fast Fourier Transform (FFT);
- N-Body, a simulation of a dynamical system of particles, usually under the influence of physical forces;
- Mesh generation/partitioning, and
- Adaptation/repartitioning.

Although many codes and libraries used by the CoEs address these computational problems internally, some make use of explicit libraries that are solely parallelised and optimised for a specific type of area and proved to be scalable. In this section, we also briefly describe the typical domains, software packages and workloads (as well as libraries) that are of interest to each of the CoEs.

E-CAM contains well-known parallel molecular dynamics tools such as GROMACS[8], Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)[9], Nanoscale Molecular Dynamics program (NAMD)[10], and Extensible Simulation Package for Research on Soft matter (ESPResSo)[11]. GROMACS is primarily designed for simulations of proteins, lipids and nucleic acids and it can be run in parallel either by threads or MPI. LAMMPS is a classical molecular dynamics code and it can model atomic, polymeric, biological, metallic, granular, and coarse-grained systems and the parallelism is achieved with MPI. NAMD is a molecular dynamics code for biomolecular modelling and the parallelization is based on Charm++ parallel objects, a machine independent parallel programming system. ESPResSo++ is a library for classical molecular dynamics simulation with short and long ranged pair, angular or dihedral interactions and the parallelism is based on MPI. The applications consist of FFTs, N-Body simulations, and mesh adaptation/repartition. The simulation applications require high accuracy so the need for scalability becomes imperative.

MaX CoE contains the application codes Quantum ESPRESSO; Yambo, a Many-Body code; Spanish Initiative for Electronic Simulations with Thousands of Atoms (SIESTA); FLEUR, a full potential linearised augmented planewave (FLAPW) code; and Automated Interactive Infrastructure and Database for Computational Science (AiiDA). Quantum ESPRESSO is a software suite for electronic-structure calculations and materials modelling at the nanoscale and it can run on almost all architectures. Yambo is designed for many-body calculations in solid state and molecular physics and it uses MPI for parallelism. SIESTA performs electronic structure calculations and ab initio molecular dynamics simulations of molecules and solids. FLEUR is a full potential linearised augmented planewave code based on density-functional theory. In MaX, AiiDA is used for database operations. Among these codes, Quantum ESPRESSO, Yambo and FLEUR contain dense linear algebra computations and SIESTA contains sparse linear algebra computations. FFTs are utilised in all codes except AiiDA. This project utilises ELPA, MAGMA and FFTW libraries; MAGMA is a simulation software for virtual experimentation and optimization in all metalcasting applications.

The types of applications in CoeGSS can be classified as follows:

1. Build a model using large amount of data;
2. Data analytics;
3. Multi-agent-based simulations.

The computational problems of HPC applications within BioExcel typically fall into the following three categories: sparse linear algebra, Fast Fourier transforms (FFTs) and n-body methods. The FFTW library is used in BioExcel. This library is for discrete Fourier transforms and delivers adequate scalability and performance for BioExcel's applications. FFTW is rated as "very good" in the scalability, performance, resilience, productivity and sustainability criteria, and "good" in the portability criterion.

ESiWACE points out scalability as one of the biggest challenges from a high performance computing perspective. The applications code in this project are Unified Model/ICON (Icosahedral non-hydrostatic), 4DVar, IPSLCM and IFS/OpenIFS. All these codes contain dense linear algebra computations. Unified Model/ICON and IFS/OpenIFS contain sparse linear algebra computations as well. IPSLCM and IFS/OpenIFS utilise FFTs. This project also utilises the FFTW library.

The 'Scalable Libraries and Algorithms' sections of the questionnaires belonging to EoCoE, PoP, and NOMAD do not include any information. Table 6 provides and overview of the scalable algorithms/techniques of interest to the various CoEs.

	MaX	CoeGSS	BioExcel	ESiWACE	E-CAM
Dense LA	X			X	
Sparse LA	X		X	X	
FFT	X		X	X	X
N-Body			X		
Mesh Generation/Partitioning				X	
Adaption/Repartitioning				X	X
Database search	X				
Multi-agent		X			

Table 6 Summary of responses to question 21

Table 7 provides an overview of the libraries of interest to the various CoEs.

	MaX	CoeGSS	BioExcel	ESiWACE
ELPA	X			
MAGMA	X			

	MaX	CoeGSS	BioExcel	ESiWACE
SuperLU				
MUMPS				
Hypre				
DUNE				
FEAST				
MLD2P4/PBLAS				
FFTW	X		X	X
FFTE				
PETSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Agent based		X		
MaPHyS				

Table 7 Summary of responses to question 23

One library that we have identified as being commonly exploited by several CoEs is the FFTW library. It is used in different levels of parallelism (threading, distributed memory parallelism, and accelerators such as GPUs and Xeon Phi's). Currently, some projects found it to scale to around 10,000 cores and target a scalability of around 100,000 cores. It is rated quite well in performance, portability and productivity criteria. Its pro is listed as the support for any FFT vector length and the cons are listed as the compatibility issues with FFTW, FFTW3 (FFTW versions 3.x) and Math Kernel Library (MKL) and issues with different batch sizes.

Eigenvalue SoLvers for Petaflop-Applications (ELPA) library, as also pointed out in PRACE-3IP D7.2.1 deliverable, uses the same matrix structure of ScaLAPACK with a complete re-implementation of parallel linear algebra routines; ScaLAPACK is a library of high-performance linear algebra routines for parallel distributed memory machines. In that report, ELPA was found to be 2-3x faster than ScaLAPACK. The project MaX has been using ELPA in a distributed setting and they scaled it to 10,000 cores. They found it to enhance the scalability by a factor of 2 (which is in agreement with the findings of the previous deliverable). However, it is pointed out that ELPA has a bleak future as it has neither a clear release plan nor a clear license.

It is clear that classical HPC library domains - dense and sparse linear algebra and FFTs - remain important for a significant number of applications. Scalability improvements in these library domains, both at the implementation and algorithm levels, as considered by the European FP7 Exascale projects Numerical Methods and Tools for Key Exascale Computing Challenges in Engineering (NUMEXAS)[4] and Engineering and Applied Sciences and Exascale Algorithms and Advanced Computational Techniques (EXA2CT)[5], will be critical for future large scale HPC systems.

2.5 IO Management Techniques

As identified in PRACE-3IP D7.2.1[7], many of the applications that have exploited the PRACE infrastructure to date have been limited by I/O bottlenecks. Due to the increasing observational datasets from many scientific fields and, as reflected by the demands across many of the European CoEs, these problems are not expected to be easily solved soon (instead, they are expected to become even more acute on the road to Exascale). To confront some of these challenges, in PRACE-3IP WP7, several state-of-the-art I/O tools that should help to improve the I/O performance and data management of European applications for multi-petascale systems were surveyed. In particular, there was a focus on characterizing and exploiting several high-level I/O libraries and file format standards: PNetCDF, a library providing high-performance parallel I/O; HDF5, a data model, library, and file format for storing and managing data; XML IO Server (XIOS), a library dedicated to I/O management of climate codes; and SIONlib, a scalable I/O library for the parallel access to task-local files; as well as I/O profiling tools, such as Darshan, a scalable HPC I/O characterization tool. As part of Task 7.2a, it is planned to build on the work carried out on I/O management in PRACE-3IP WP7 and to continue to exploit emerging I/O tools and techniques, for the benefit of PRACE, the various European CoEs and as well as European HPC application users and developers more generally. A high-level view of which I/O libraries are in use/of interest to the CoEs can be seen in Table 8.

	BioExcel	EoCoE	ESiWACE	MaX
MPI-I/O		X		
pNetCDF		X	X	
HDF5			X	X
pHDF5		X		
XIOS		X	X	
SIONlib		X		
FTI				
GROB API			X	
GRIB 2			X	
TNG	X			
iotk				X

Table 8 I/O libraries are in use/of interest to the CoEs

Based on the analysis of the answers provided within the I/O section of the Task 7.2a questionnaire, several key points have been identified, which should inform WP7 during the Task 7.2a exploitation phase and which are summarised below.

In terms of the state-of-the-art I/O tools that were investigated in PRACE-3IP, some evidence of the CoEs' interest in pNetCDF, HDF5, XIOS was found, as well as interest in other custom I/O libraries (iotk, a I/O toolkit is used by the Quantum Espresso package and TNG, a portable I/O library that appears to be developed by BioExcel members for specific codes of interest), which we will aim to further exploit in Task 7.2a.

For the CoEs focused on materials science, such as MaX, E-CAM and BioEXCEL, I/O is not typically found to be a major performance bottleneck on current petascale systems (this may obviously change over the lifetime of projects within the CoEs). While high counts of input and output files are used per production run (up to 10,000 files for some codes), average file sizes tend to be small (10kB-1GB) and read and write frequencies per production run tend to be low (unless checkpointing is employed). While input and output strategies for I/O are typically listed as "parallel", the high number of files opened and closed during production runs suggests a one-file per process approach and possible high meta-data overhead for many codes of interest.

Several CoEs have highlighted the desire for a common standard that would allow for the exchange of data between different materials science codes within the CoE and have identified HDF5 as a possible candidate due to its popularity. Supporting such a common standard across codes may be of interest to other CoEs, which may be worth exploring further as part of Task 7.2a activity.

Some CoEs use codes that have their own specific I/O libraries and have highlighted a willingness to assist with the exploitation of such libraries within PRACE and other CoEs, which could be pursued as part of the exploitation phase of Task 7.2a.

I/O Management is deemed to be the key to success to those CoEs with a Big Data focus; Big Data refers to data sets that are so large or complex that traditional data processing applications are inadequate. While the I/O focus and tools of such CoEs may reflect requirements outside of those most familiar to PRACE to date, the "Big Data meets HPC" focus as well as the I/O tools and techniques of CoEs like NoMAD and CoeGSS may offer WP7 with an interesting opportunity for cross fertilization of ideas and techniques during the Task 7.2a exploitation phase.

The ESiWACE CoE, which reflects many of the requirements of the European weather and climate community typically works with both small and large input and output data files (where large is ~100-300 GB) and exploits a wide range of I/O management tools (XIOS, HDF5, PNetCDF, NetCDF4, GRIB2) across several codes (IFS, NEMO, NorESM) for handling what is mainly parallel I/O strategies (there is also evidence of serial and hybrid I/O strategies in several codes); NetCDF4 is version 4 of a set of libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data and GRIB2 is version 2 of a concise data format commonly used in meteorology to store historical and forecast weather data. From the answers to the Task 7.2a questionnaire, it is evident that efficient checkpointing techniques are a high requirement for this CoE, which may be worth further investigation during the exploitation phase of Task 7.2a (e.g., within the context of emerging burst buffer approaches if available at this time).

2.6 European Exascale Projects

There were a number of Exascale projects, funded under FP7, to address the hardware and software challenges for future generations of HPC systems. DEEP[2] and Mont Blanc[3] (and their successors DEEPer and Mont Blanc 2) were focused on the co-design of hardware and software, while Collaborative Research into Exascale Systemware, Tools and Applications (CRESTA)[1], was concerned with the co-design of system software and applications. EPiGRAM looked at evolving the current set of programming models towards Exascale, while NUMEXAS[4] and EXA2CT[5] considered new numerical methods, and their implementation.
Table 9 summarises the responses to question 28: "Are the CoE's members acquainted with or have any of them contributed to any of the following projects?"

	CRESTA	**DEEP/ DEEP-er**	**Mont Blanc/ Mont Blanc2**	**NUMEXAS**	**EXA2CT**	**EPiGRAM**
BioExcel	X	X				X
CoeGSS	X		X			

	CRESTA	DEEP/ DEEP-er	Mont Blanc/ Mont Blanc2	NUMEXAS	EXA2CT	EPiGRAM
E-CAM						
EoCoE			X			
ESiWACE	X	X	X			
MaX		X	X			X
NoMaD	X	X	X			
POP	X	X	X		X	

Table 9 Summary of responses to question 28

Many of the CoEs are aware of, or have partners contributing to, the larger European Exascale projects (CRESTA, DEEP/DEEP-er, Mont Blanc/Mont Blanc 2), but there are fewer connections to the smaller projects (NUMEXAS, EXA2CT, EPiGRAM[6]). Only E-CAM has no connections to any Exascale project.

Responses to question 29: "Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?" were mostly rather brief. Several responses were negative, but positive points of note include:

- Parts of the work in CRESTA are directly related to BioExcel objectives.
- For CoeGSS, the findings of CRESTA and MontBlanc/MontBlanc 2 may form the basis for efficient co-design approaches and in addition, novel architectures evaluation.
- MaX will try to use tasking approaches and multiple levels of parallelization that showed their importance in the achievement of the MontBlanc and DEEP results.

3 Review across CoEs

In this section, each questionnaire is examined from the perspective of the CoEs. A short description of the CoE is provided and then the specific needs of the CoE with respect to PRACE and exascale are explored.

3.1 BioExcel

BioExcel's expertise is in the area of biomolecular modelling and simulations. This includes the usage of HPC and High Throughput Computing (HTC) infrastructures for structural and functional studies of proteins, DNA (Deoxyribonucleic acid), saccharides, membranes, solvents and small molecules applied to both fundamental research and industrial applications in e.g. drug development, biotechnology, food, and chemical industry.

The aim of BioExcel is fourfold:

- Improving the *performance, efficiency and scalability of widely used software packages* will greatly contribute towards the progress of biomolecular research.
- Improvements in the *usability of workflow platforms for data handling and analysis* will increase the productivity of researchers as well as uptake of advanced Information and Communications Technology (ICT) technologies.
- *Training and consultancy efforts* will ensure the adoption of recommended tools and associated best practices, which will improve the efficiency of HPC resource usage.

- Successful implementation of *sustainable operation* will give opportunity for *long-term application development and provisioning of support* to the academic and industrial usage communities.

In accordance with these goals, the expected main outcome of BioExcel is *scientific domain impact* through the publication of *research papers, development of software and tools* as well as *contributions to open-source software projects* for *industrial and research applications*.

The following opportunities for collaboration between PRACE and BioExcel have been identified:

- BioExcel may require from PRACE general assistance in performing large-scale benchmarks as well as expertise in the area of development, porting and tuning of codes for some specific platforms.
- BioExcel expressed interest to temporally and spatially aligning their training events with PRACE events, especially with PRACE Advanced Training Centers (PATCs) that provide education and training opportunities for computational scientists in Europe, but also with PRACE seasonal schools. Note that BioExcel already is co-organizer of the PRACE Spring School "HPC in the Life Sciences", which will take place early spring 2017 in Sweden, organised by Swedish National Infrastructure for Computing (SNIC) and Kungliga Tekniska högskolan (KTH), a Swedish University.
- BioExcel expressed interest in collaborating with PRACE with respect to online documentation in order to link BioExcel's and PRACE's knowledge bases.

HPC Applications and Requirements

The most relevant computing facilities for BioExcel are HPC systems on one side and cloud-based platforms for workflow management, data handling and data analysis on the other side.

BioExcel's usage of HPC systems will focus on testing and benchmarking biomolecular community codes. Consequently, their HPC accesses will typically be for achieving short-term goals and will potentially be highly resource-intensive (for very-large-scale benchmarking). On the other hand, continuous HPC access for longer periods of time is very relevant for the research groups BioExcel is supporting.

HPC applications relevant for BioExcel will rely on both the capability and capacity of the HPC system and will typically be run in batch mode. They require memory-intensive processing (general and floating-point), high-bandwidth/low-latency network interconnects as well as usage of accelerator hardware. Both inter-node and intra-node performance are very important, whereas resilience is currently considered of little importance only. The importance of energy-efficiency has not been indicated in the questionnaire.

Targeted platform architectures for codes are: x86, PowerPC, NVIDIA[12] GPGPU, AMD[13] GPGPU/APU and ARM.

The three most widely used large-scale HPC applications within BioExcel are:

- GROMACS[8] which potentially could scale up to 100K cores according to BioExcel's experience;
- HADDOCK[14] (High Ambiguity Driven biomolecular DOCKing), a Software package for integrative modelling of biomolecular complexes, which is a high throughput code and is not considered ready for 100K or beyond; and
- CPMD[15] (Car–Parrinello Molecular Dynamics), a computational chemistry software package for parallelised ab initio molecular dynamics, which currently could scale up to approximately 50K cores.

An estimation of the currently achieved percentage of peak performance could not be given for any of these codes.

Programming interfaces and standards
The following programming tools are relevant for the development of HPC applications within BioExcel:

- For inter-node parallelization, MPI is most widely used. The usage of PGAS approaches, such as: Fortran CoArrays; UPC (Unified Parallel C); Chapel (Cascade High Productivity Language), a parallel programming language developed by Cray; or X10, a programming language for parallel computing using the PGAS model developed by IBM; has not been indicated.
- For intra-node parallelization, i.e. threading, OpenMP is used. Future use of Intel TBB is under consideration.
- For exploitation of accelerator hardware (GPU / Xeon Phi), OpenCL (Open Computing Language), CUDA (Compute Unified Device Architecture) and OmpSs is used. The usage of OpenACC has not been indicated.

For the above programming tools, all of the characteristics productivity, openness (open standard), sustainability and portability are considered very important.

Debuggers and Profilers
Currently, there seems to be limited experience with profiling tools for HPC applications within BioExcel. According to the questionnaire answers, most of the profiling analysis is done with manual time measurement and corresponding console or log output. Apart from that, there is some experience with TAU (Tuning and Analysis Utilities), a portable profiling and tracing toolkit for performance analysis of parallel programs, however, this tool is not used regularly. Interest has been indicated for future regular use of the Extrae and Paraver toolset in order to analyse all kinds of performance aspects (intra- and inter-node performance, I/O performance, and performance of GPU or Xeon-Phi accelerated computation kernels) as well as MPI communication patterns. For debugging purposes, valgrind and DDT are used.

All of the above debugging and profiling tools are typically used on a very limited number of nodes (about one to four) currently, but ideally also profiling and debugging of applications at their scaling limit would be of interest for BioExcel. BioExcel's conclusion on these tools is that they currently provide very rich functionality, however they tend to fail for complex use-cases, as for example for hybrid applications using MPI, OpenMP and CUDA, due to inherent limitations or bugs.

Scalable libraries and algorithms
Computational problems of HPC applications within BioExcel typically fall into the three categories Sparse linear algebra, Fast Fourier transforms (FFTs) and n-body methods.

BioExcel was able to give feedback to just one numerical library, namely FFTW (Fastest Fourier Transform in the West), which is a library for discrete Fourier transforms and delivers adequate scalability and performance for BioExcel's applications. FFTW is rated as "very good" in the categories scalability, performance, resilience, productivity and sustainability and "good" in the category portability.

I/O Management and Techniques
Unfortunately, BioExcel was not able to respond in detail to our questions in the I/O Management Techniques section. However, the use of a portable I/O library called TNG within the GROMACS code was highlighted, with an API for C/C++ and Fortran support. The TNG library is known to be very lightweight but highly customisable and could potentially be useful to other CoEs as well as within PRACE. The BioEXCEL CoE is willing

to assist with the exploitation of the library within PRACE and other CoEs, which could be pursued as part of the exploitation phase of Task 7.2a.

3.2 CoeGSS

CoeGSS provides advanced decision-support in the face of global challenges. It brings together the power of high-performance computing and some of the most promising thinking on global systems in order to improve decisions in business, politics, and civil society.

There are currently three pilot areas in CoeGSS:

- Pilot-1 Health habits: The simulation software will be a powerful tool in the hands of policymakers to evaluate the impact of health programs and to increase their efficiency so that healthcare expenditures are decreased and life expectancies will be increased.
- Pilot-2 Green growth: This method allows to model and understand the global diffusion of innovations like different kinds of electric cars. It can later be adapted for studying the global dynamics of renewable energies, energy efficient buildings and the whole range of green growth opportunities.
- Pilot-3 Global urbanization: Via creating synthetic populations with realistic statistical distributions of characteristics' values will prove precious simulations for clarifying influences between different elements of the city and point to possible or more efficient levers to improve cities' everyday life.

The following publicly available tools simulate particular and clearly delimited aspects of reality that address specific areas of interest:

- NetLogo[16], a multi-agent programmable modeling environment developed at the Center for Connected Learning and Computer-Based Modeling (CCL) in Northwestern University;
- LevelSpace[17], a NetLogo extension for running an arbitrary number of NetLogo models concurrently, and have them communicate;
- Pandora[18], a framework designed to create, execute and analyse agent-based models in high-performance computing environments;
- Repast[19], a suite of advanced, free, and open source agent-based modeling and simulation platforms; and
- Mason[20], a fast discrete-event multi-agent simulation library core in Java, designed to be the foundation for large custom-purpose Java simulations, and to provide more than enough functionality for many lightweight simulation needs. It contains both a model library and an optional suite of visualization tools in 2D and 3D.

General objectives and expected impact of the CoE

- Build up an expertise centre to handle global problems related with life and social sciences.
- Collect accurate and multi-domain data in order to analyse impacts of decisions and actions.
- Use HPDA (High Performance Data Analysis) in conjunction with HPC in order to obtain large-scale accurate data sets for enabling GSS (Global Systems Science) simulations.

HPC Applications and Requirements

CoeGSS will make use of HPC systems in two ways:

1. Generating data which will be basis of simulations;

2. Simulations and data analytics based on the generated data.

HPC applications in CoeGSS need supercomputers, clusters for big data analytics (Spark, a fast and general engine for large-scale data processing, and Hadoop, a Java framework for distributed storage and distributed processing of very large data sets, clusters), large shared memory systems possibly with visualisation support. Their needs can be continuous, short or long term. They depend on capability, interactiveness, and batch mode of the underlying systems. These applications will contain floating-point operations, intense I/O operations, and operations on integer, text and image. They also need low-latency interconnects, local/remote memory accesses.

PRACE is expected to support CoeGSS in terms of general assistance, training, online documentation, domain related expertise, and code development/porting/tuning.

Programming Interfaces and Standards
The applications in CoeGSS mainly use MPI and OpenMP. Usage of CUDA and OpenACC can be investigated. For these programming tools, all of the characteristics: productivity, openness (open standard), sustainability and portability are considered to be very important.

Debuggers and Profilers
Standard methods for debugging such as console outputs are considered to be enough. Since Python is widely used in the GSS community, the Python profiling tool, cProfile, is important.

Scalable Libraries and Algorithms
The applications within CoeGSS make use of constructing databases out of various kinds of data sources composed of different volumes and velocities in order to build up a proper model. Data sources and the number of agents in the simulations need to be increased in order to mitigate the risk of wrong decisions.

I/O Management Techniques
I/O will be the key to speedup CoeGSS since several terabytes of data are read and written and upto 50 data sources are required to form a single agent-based model.

3.3 E-CAM

E-CAM will create, develop and sustain a European infrastructure for computational science applied to simulation and modelling of materials and of biological processes of industrial and societal importance. Building on the already significant network of 15 CECAM (Centre Européen de Calcul Atomique et Moléculaire) centres across Europe and the PRACE initiative, it will create a distributed, sustainable centre for simulation and modelling at, and across, the atomic, molecular and continuum scales. The ambitious goals of E-CAM will be achieved through three complementary instruments:

1. Development, testing, maintenance, and dissemination of robust software modules targeted at end-user needs;
2. Advanced training of current and future academic and industrial researchers able to exploit these capabilities; and
3. Multidisciplinary, coordinated, top-level applied consultancy to industrial end-users (both large multinationals and Small and Medium-sized Enterprises, SMEs).

The development of this infrastructure will also impact academic research by creating a training opportunity for over 300 researchers in computational science tailored to their domain expertise. It will also provide a structure for the optimisation and long-term maintenance of important codes and provide a route for their exploitation. Based on the requests from its industrial end-users, E-CAM will deliver new software in a broad field by creating over 200 new, robust software modules. The modules will be written to run with maximum efficiency on hardware with different architectures, available at four PRACE centres and at the Hartree Centre[21] for HPC in Industry. The modules will form the core of a software library (the E-CAM library) that will continue to grow and provide benefit well beyond the funding period of the project.

HPC Applications and Requirements
The exploitation of HPC systems at the E-CAM CoE will focus mainly on testing and benchmarking materials science codes and workflows on supercomputers/HPC clusters. The CoE's HPC utilisation will typically be continuous, but may also demand sporadic access, with a need for both capacity and capability computing.

Due to the diversity of the actions planned within the E-CAM CoE, there will be requirements in general processing, the exploitation of accelerator hardware, as well as existing and emerging HPC software and tool stacks. The codes of interest within the CoE nearly always use double precision for floating point operations, with one code of interest (GROMACS) sometimes using both double and single precision.

As expected, both inter-node and intra-node performance are considered very important, with resilience considered to be somewhat important and energy efficiency deemed of little importance for the CoE. The targeted platform architectures for codes within E-CAM are x86, Power, Xeon Phi and NVIDIA GPGPUs.

Based on the answers to the Task 7.2a questionnaire it appears that the HPC community codes of most interest within E-CAM are currently GROMACS, LAMMPS, NAMD and ESPResSo++, all of which have proven to be highly scalable on large-scale systems to date. ESPResSo++ is an extensible parallel simulation software for soft matter research. It has a common root with ESPResSo, but is free, open-source software published under the GNU General Public License (GPL). Since the CoE is aiming to develop a wide range of software modules targeting the materials community throughout its lifetime, it is expected that the current list of community codes of interest may grow over time.

The E-CAM CoE will seek PRACE support through Training and Workshops, porting and tuning and online documentation.

Programming interfaces and standards
The following programming tools are relevant for the development of HPC applications within E-CAM:

- For inter-node parallelization, MPI is most widely used. There seems to be no current interest in PGAS approaches such as Fortran CoArrays, UPC, Chapel or X10. For intra-node parallelization, i.e. threading, OpenMP appears to be the model of choice for all codes of interest.
- For exploitation of manycore hardware (GPU / Xeon Phi), CUDA, OpenACC and OpenCL are of interest.
- For the above programming tools, openness (open standard), sustainability and portability are considered very important, with productivity considered somewhat important.

Debuggers and Profilers
There is some evidence of interest in the exploitation of profiling tools for HPC applications within E-CAM. Scalasca, TAU and HPCToolkit are sometimes exploited; HPCToolkit is an open-source suite of tools for profile-based performance analysis of applications developed at Argonne National Laboratory. However, in most cases manual profiling via (console/log output) is used. All in all, debuggers and profilers have seldom been used to date by E-CAM members.

Scalable libraries and algorithms

The computational problems within the codes of interest at E-CAM typically fall into the categories of FFTs (GROMACS, LAMMPS, NAMD, ESPResSo++), N-Body (GROMACS, LAMMPS, NAMD, ESPResSo++) and Mesh-based algorithms, with a particular interest in adaptation and repartitioning.

In terms of third party libraries, E-CAM has no interest in any of the libraries investigated as part of the survey carried out in PRACE-3IP, although interests in FFT-based algorithms suggests that leveraging third party FFT libraries or exploring new FFT techniques would be of interest to the CoE (This should also be the case of mesh libraries).

I/O Management Techniques

Little information is provided on I/O management requirements within E-CAM in the Task 7.2a questionnaire, due to the fact that these have not yet been clearly identified at this early stage of the CoE. I/O is not typically expected to be a major performance bottleneck for E-CAM. As is typical of many materials science codes, average file sizes tend to be small (10kB-1GB) and read and write frequencies per production run tend to be low (unless checkpointing is employed). Further investigations on I/O requirements for E-CAM may be carried out at later stages of interaction between PRACE and the CoE.

3.4 EoCoE

General objectives and expected impact of the CoE

EoCoE stands for Energy Oriented Centre of Excellence. EoCoE aims to exploit the prodigious potential offered by the ever-growing computing infrastructure to foster and accelerate the European transition to a reliable and low carbon energy supply. The rationale of EoCoE is that the current revolution in hardware technology calls for a similar paradigm change in the way application codes are designed. EoCoE will assist the energy transition via targeted support to four renewable energy pillars: Meteo4Energy, Water4Energy, Materials4 Energy and Fusion4Energy, each with a heavy reliance on numerical modelling. These four pillars are anchored within a strong transversal multidisciplinary basis providing high-end expertise in applied mathematics and HPC. EoCoE is structured around a central Franco-German hub coordinating a pan-European network, gathering a total of 8 countries and 23 teams. Its partners are strongly engaged in both the HPC and energy fields; a prerequisite for the long-term sustainability of EoCoE and also ensuring that it is deeply integrated in the overall European strategy for HPC. The primary goal of EoCoE is to create a new, long lasting and sustainable community around computational energy science. At the same time, EoCoE is committed to deliver high-impact results within the first three years. It will resolve current bottlenecks in application codes, leading to new modelling capabilities and scientific advances among the four user communities; it will develop cutting-edge mathematical and numerical methods, and tools to foster the usage of Exascale computing. Dedicated services for laboratories and industries will be established to leverage this expertise and to foster an ecosystem around HPC for energy. EoCoE will give birth to new collaborations and working methods and will encourage widely spread best practices.

The EoCoE project is guided by four high-level objectives with a long-term perspective, with clear milestones:

1. Enabling scientific breakthroughs in the energy domain by re-designing existing simulation application codes for the four selected user communities;
2. Develop cutting-edge mathematical, numerical and computational methods and tools to foster precise simulation and visualisation for future applications using Exascale computing;

3. Adapt Services activities (outputs of objectives 1. and 2. above) to laboratories, industries and SMEs, including training activities for reducing the skill gap;
4. Foster HPC and energy oriented scientific and industrial communities' ecosystem.

EoCoE also includes a transversal activity associated to a transversal work package in the project work plan: in order to help supply the high-end scientific and industrial research and demands. It comprises a cross-sectional transversal basis:

1. HPC related expertise in numerical methods and applied mathematics,
2. Linear algebra,
3. System tools for HPC,
4. Advanced programming methods for Exascale, and
5. Tools and services for HPC.

Activities in relation to these topics are motivated by the specific energy related issues in order to produce user-driven modules (software platform, libraries) to be run on HPC infrastructures.

HPC Applications and Requirements
The EoCoE applications belong to four thematic pillars:

- **Meteorology for Energy,** as a means to predict variability of solar and wind energy production;
- **Materials for Energy,** dedicated to photovoltaic cells, batteries and supercapacitors for energy storage;
- **Water for Energy,** as a vector for thermal or kinetic energies, focussing on geothermal and hydropower;
- **Fusion for Energy,** for electricity plants as a long term alternative energy.

Each of these pillars is associated with a work package of the project work plan. Altogether, this represents a total of 23 application codes, some of them being open source while others are in-house codes, resulting in a wide variety of requirements from the point of view of scalability improvement. For the corresponding research activities, the targeted compute facilities are mostly supercomputers or high performance computing systems. Similarly, following the variability of mathematical and numerical methods underlying the involved application codes, platforms based on x86, Power, Xeon Phi and GPU are relevant for the activities planned in EoCoE.

Programming Interfaces and Standards
Due to the large number of application codes considered in the project, several programming interfaces and standards are adopted. MPI, OpenMP, OpenCL and CUDA are parallel programming models, which are used in most of the cases.

Debuggers and Profilers
Nearly all the major applications in the application pillar work packages suffer from one or more performance bottlenecks. Moreover, rapid hardware evolution poses a challenge for software to maintain high performance; especially for coupled, multi-scale programs underlying the inter-disciplinary challenges addressed in the applications pillars. The cooperative actions within the transversal work package in EoCoE also focus on various aspects of profiling, performance evaluation, continuous code integration and benchmarking. In most of the situations, the tools used for this purpose will be identified on the fly during the project. It is worth noting the use of the Domain Specific Language (DSL), Boost, for the automatic generation of optimised kernels; Boost is a set of C++ libraries for linear algebra, pseudorandom number generation, multithreading, image processing, regular expressions, and unit testing.

Scalable Libraries and Algorithms

The transversal work package of EoCoE includes a task dedicated to scalability improvement of the numerical schemes used for the discretisation of systems of differential equations modeling the physical phenomena at the heart of the application pillars. The corresponding activities can be either working on the parallelisation aspects without changing the adopted numerical scheme, or adopting a new scheme which demonstrates better parallel performances. Three families of numerical algorithms are considered: high order finite element methods, particle methods and parallel in time integrators.

Linear equation systems are also an integral part of many of the application codes found in the application pillars of EoCoE, which means that these applications have to rely on robust, high-performance, portable solvers when run on supercomputers. Some of the application codes exploit existing solvers from well-known numerical linear algebra libraries such PETSc[22] (developed at Argonne National Laboratory) and Hypre[23] (developed at Lawrence Livermore National Laboratory). In some cases, a sparse direct solver such as MUMPS (MUltifrontal Massively Parallel Sparse direct Solver) is adopted because robustness is the main issue. Such a sparse direct solver will not scale on highly parallel systems therefore EoCoE includes an activity on the core development of advanced linear algebra solvers on emerging HPC architectures and their integration in applications from the pillars. These solvers can be very specific to the mathematical model and numerical scheme for discretising the associated system of differential equations (such as a geometric multigrid solver or a particular preconditioned iterative solver) or they can be black-box algebraic solvers such as MaPHyS (Massively Parallel Hybrid Solver) developed at Inria which is a hybrid iterative-direct solver relying on domain decomposition principles in order to maximise scalability on large number of cores.

I/O Management Techniques

Performing efficient I/O for very large datasets on current supercomputers is already challenging and will become more challenging for the next supercomputer generations. If fault tolerance is not yet a key technology, it will likely become one in the coming generation of supercomputers. In EoCoE, short term and concrete support will be given to all application codes from the application pillars. Each application code will receive specific tuning guidelines with small impact on the code in order to improve their I/O performance on different supercomputers. Three software libraries have been selected for supporting this goal: XIOS, SIONlib and FTI (the Fault Tolerance Interface). A longer-term activity will consist of the design and the first implementation of PDI (Parallel Data Interface), which will thus be a software result of the project. The aim of this interface is to decouple as much as possible the parallel application from the I/O and/or fault tolerance package, which actually performs the work. As a result, applications will implement a single interface (the PDI) and parallel I/O packages (like MPI-IO, Parallel HDF5 and pNetCDF) and fault tolerance/checkpointing packages (like FTI) come as plug-ins for PDI; MPI-IO is a library which provides parallel I/O and Parallel HDF5 (PHDF5) provides users with an option to have multiple processes perform I/O to an HDF5 file.

Conclusion

Several of the application codes considered in EoCoE are concerned with the solution of large sparse linear systems at some point. The sparse linear solver used for achieving this task greatly influences the overall scalability of the application. On one hand, if a direct solver is adopted, then the computational complexity and numerical efficiency of the solve phase are almost independent of the number of processing cores but the parallel speedup degrades for highly parallel configurations because of communication costs. On the other hand, if a preconditioned iterative solver is adopted, parallel scalability can be achieved but the numerical efficiency (i.e. the required number of iterations to convergence) strongly depends on the quality of the preconditioner. In the projects proposed by Inria, this issue is studied by considering a hybrid iterative-direct sparse linear solver, MaPHyS, which is based on domain decomposition principles, and two application codes that are not related to energy production but that require the solution of large sparse linear systems. The results that will be obtained will in some sense constitute guidelines for some of the application codes considered in EoCoE.

3.5 ESiWACE

In this section we summarize the answers to our questionnaire from ESiWACE, i.e. the Centre of Excellence in Simulation of Weather and Climate in Europe. In constrast to other CoEs ESi/wace provided collated responses from nine different partner institutions:

1. MPI-M, the Max-Planck-Institute for Meteorology (DE)
2. DKRZ, the German Climate Computing Center (Deutsches Klimarechenzentrum; DE)
3. Met Office (UK)
4. IPSL, Institut Pierre-Simon Laplace (FR)
5. The Geophysical Institute, University of Bergen (NO)
6. SMHI, the Swedish Meteorological and Hydrological Institute (SE)
7. BSC, the Barcelona Supercomputing Center, (ES)
8. CMCC, Fondazione Centro Euro-Mediterraneo sui Cambiamenti Climatici (IT)
9. ECMWF, the European Centre for Medium-Range Weather Forecasts (UK)

General objectives and expected impact of the CoE
The biggest challenges in meteorology and climatology are to increase the accuracy and local detail of predictions and to represent and incorporate more relevant processes for global and regional climate and weather.

ESiWACE pursues the following objectives:

- Improving the efficiency and productivity of numerical weather and climate simulations on HPC platforms;
- Supporting the end-to-end workflow of global Earth system modelling for weather and climate simulation in HPC environments;
- Providing appropriate services to the European weather and climate science community;
- Fostering the interaction between industry and the weather and climate community on the exploitation of high-end-computing systems, application codes and services;
- Increasing the competitiveness and growth of the European HPC industry.

In conclusion, ESiWACE will support and foster research that will improve the ability to protect against weather events and climate change either directly or through better planning for resilience.

The expected main outcome of ESiWACE is *scientifc domain impact* through the publication of *research papers*, the development of *software and tools* as well as through *contributions to existing open source projects*.

HPC Applications and Requirements
The most relevant computing facilities to ESiWACE are mainly HPC systems, but also large shared-memory systems as well as (Hadoop- or Spark-) clusters for big data analytics. The way ESiWACE will make use of HPC facilities does not seem to be classifiable clearly in the categories continuous or sporadic access, long- or short-term goals, high- or low-intensive use. In fact, working groups which develop workflow solutions, such as the UK Met Office, will require relatively low-intensive but very continuous access, whereas working groups

which apply these workflows, already at the time of writing, have a demand for high intensive and more sporadic access.

HPC applications of ESiWACE can be characterised as follows: The operation mode is typically batch mode; the relevance of interactive mode was only indicated by MPI-M. Targeted architectures for ESiWACE's applications are most importantly x86 and PowerPC. With respect to accelerator hardware, Xeon Phi seems to be of greater interest than NVIDIA GPGPUs - see also Figure 2. ESiWACE's applications typically rely on double or mixed precision arithmetic. By trend, mixed precision seems to get more important and even testing with single precision is on-going, as indicated by ECMWF. Particular importance for ESiWACE lies in high performance storage systems, as emphasised by UK Met Office. In fact, this is also in line with the answers to question 7 "What types of resources will your CoE need?" as summarised in Table 10 sorted by decreasing resource type relevance (taking the number of indications from the nine members of the ESIWACE consortium as measure).

Type of resource	No. of indications
Storage and I/O-requirements	6 / 9
General processing, floating point	5 / 9
Memory size and bandwidth requirements	4 / 9
Low latency interconnect	3 / 9
Software or tools	3 / 9
Local or remote access	2 / 9
Use of accelerator hardware	1 (2) / 9
General processing, non-floating point	1 / 9

Table 10 Responses to question 7 for ESiWACE

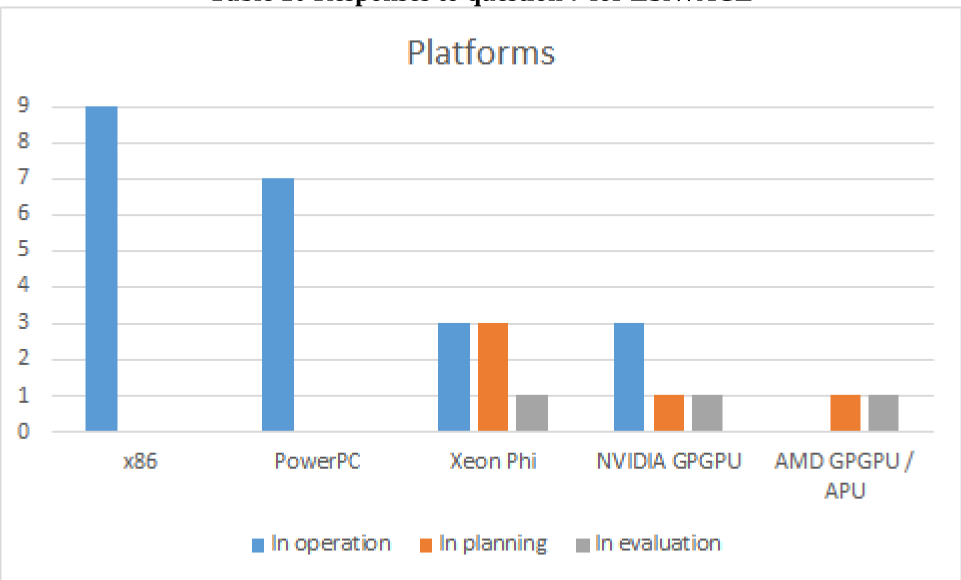

Figure 2 ESiWACE Platforms

The most widely used simulation codes within ESiWACE are:

1. IFS (Integrated Forecast System developed by ECMWF), which has been run on 229k cores on TITAN (Oak Ridge National Laboratory) and achieved about 7% of the system's peak performance; and
2. DYNAMICO (Dynamico Atmospheric Dynamical Core Model, a hybrid MPI / OpenMP parallelized simulation code) (IPSL France), which scales up to about 60k cores;

as well as three codes mentioned by BSC:

3. EC-EARTH, an Earth System Model for climate simulation;
4. NMMB (Nonhydrostatic Multiscale Meteorological Model on the B grid)/ Barcelona Supercomputing Centre- Chemical Transport Model (BSC-CTM)[24]; and,
5. NEMO (Nucleus for European Modelling of the Ocean), a oceanographic modeling framework;

all of which have not yet been tested at very large core counts.

Generally, on the development of their codes towards Exascale, the representatives of ESiWACE estimate intra-node performance to be the most important optimization criteria, followed closely by inter-node performance. Resilience and Energy-efficiency are not considered to be that important currently - see also Figure 3.

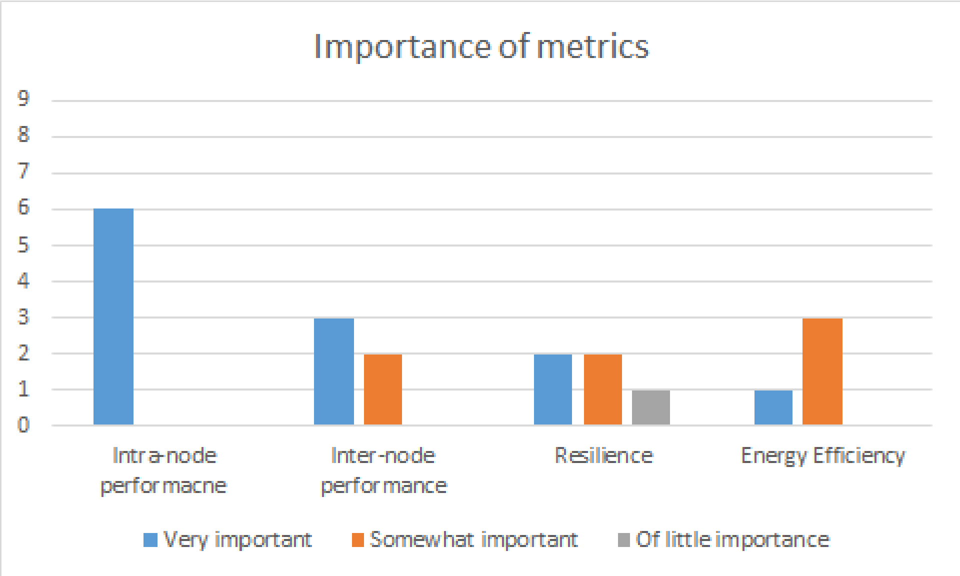

Figure 3 ESIWACE Importance of metrics

Lastly, in response to question 8, "What kind of support from PRACE will the CoE be relying on?" ESiWACE signaled strong need (4 out of 9 indications respectively) in the categories "General Assistance", "Online documentation" as well as "Computer expertise in code development, porting or tuning" and moderate need (just 2 out of 9 indications respectively) in the categories "Training, workshops" and "Domain related experience". In particular, IPSL France indicated special interest a Multi-Program Multi-Data (MPMD) demonstration case for a coupled simulation based on 3 different MPI and OpenMP parallelised executables for which they will need close collaboration with PRACE experts for setting up and running.

Programming Interfaces and Standards

Figure 4 gives an overview on programming interfaces and standards currently used within ESiWACE's applications. As unanimously indicated by all nine representatives of ESiWACE, MPI and OpenMP are unsurprisingly the two predominant programming paradigms. For exploitation of accelerator hardware, primarily OpenACC and CUDA seem to be the most relevant techniques. The current use of OmpSs as alternative has been indicated just by BSC and possible future use of OpenCL is considered by SMHI. Obviously, currently not so relevant is the use of PGAS approaches such as UPC, Chapel, Global Arrays or X10. The only exception here is SMHI, who is planning to use CoArray Fortran within the EC-Earth simulation code. Generally, for the choice of programming techniques portability and productivity are the most important criteria for ESiWACE. Sustainability and "open standard" are considered as slightly less important - compare also Figure 5.

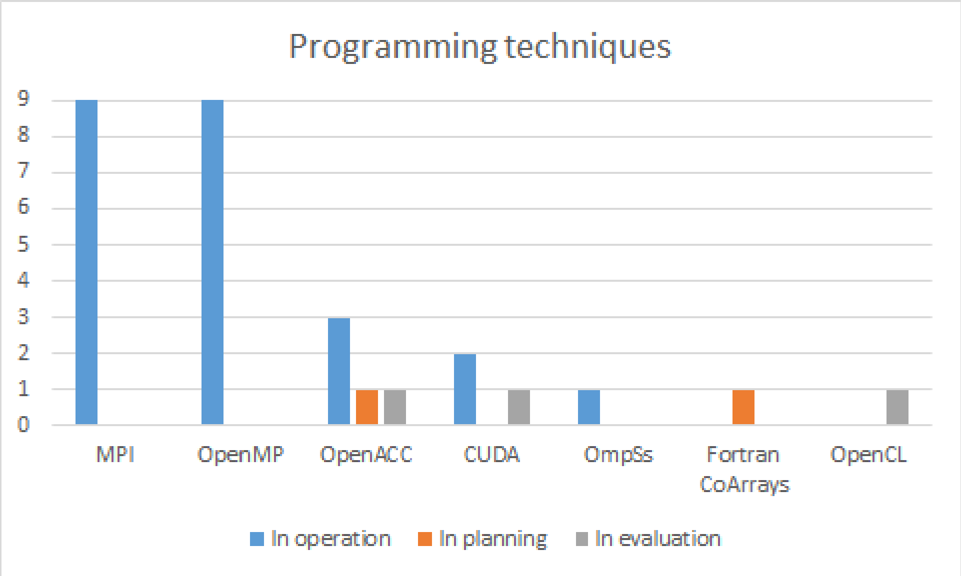

Figure 4 ESiWACE Programming techniques

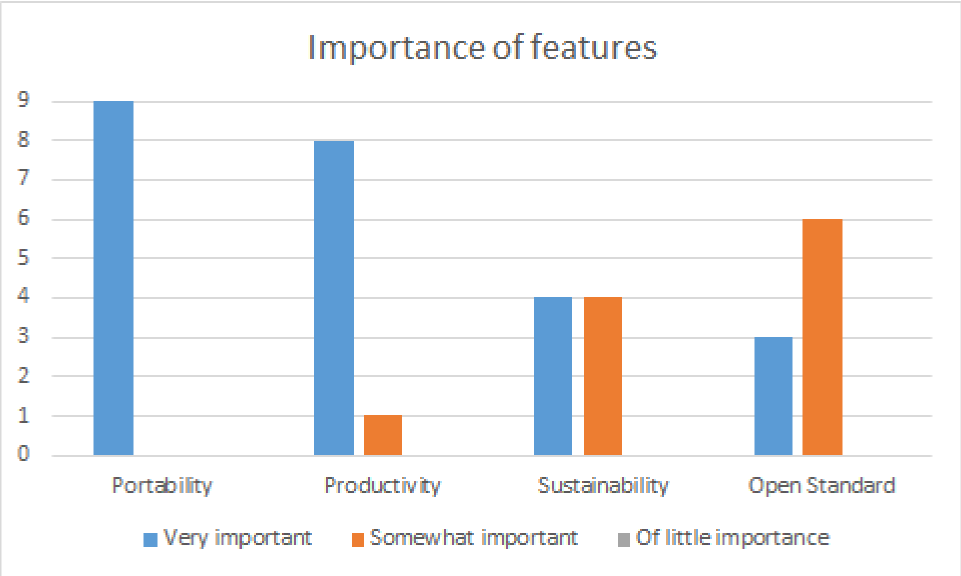

Figure 5 ESiWACE Importance of features

Debuggers and Profilers

Figure 6 gives an overview on the currently used tools for debugging and profiling within ESiWACE. For performance analysis, the most prominent method is still manual output to log files, followed by Allinea tools (MAP and Performance Report), Dr. Hook[25] and gstats (which is a manual instrumentation toolset for Fortran and C with limited functionality developed by ECMWF), Extrae / Paraver, Intel tools (VTune and Intel Trace Analyzer and Collector) and Scalasca; Intel Trace Analyzer and Collector (ITAC) is a graphical tool for understanding MPI application behavior and Dr. Hook is a simple, low-overhead instrumentation tool, which allows you to keep track of dynamic calling tree of a program and print it in the event of failure. Moreover, CrayPat (a performance analysis tool offered by Cray for the XC platform), Score-P / Vampir and are used occasionally; Scope-P is a measurement infrastructure and easy-to-use tool suite for profiling, event tracing, and online analysis of HPC applications. HPCToolkit or TAU are rarely ever used. Also for debugging manual output is obviously the most favored method. Apart from that, Totalview and Allinea DDT are used quite regularly.

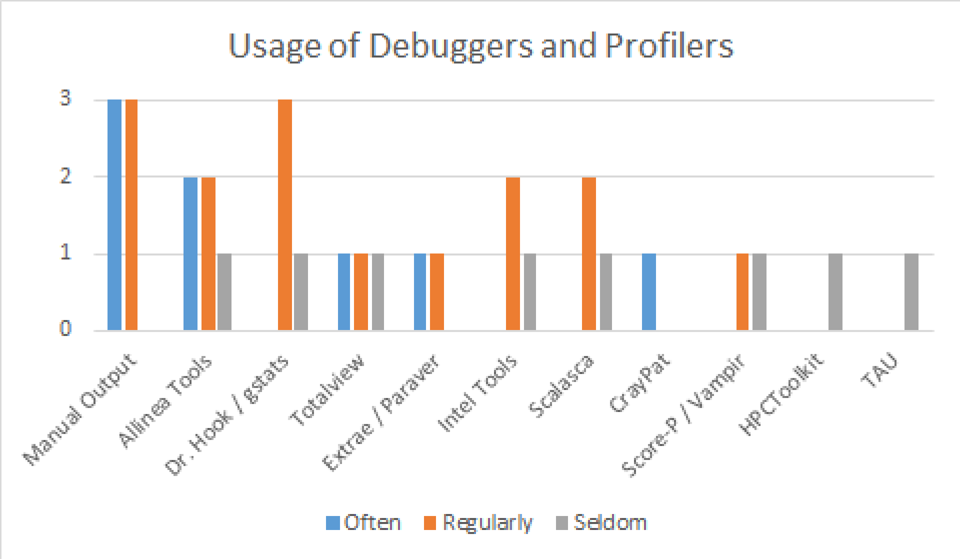

Figure 6 ESiWACE Use of debuggers and profilers

In terms of scalability, most of these tools are typically used below 100 (debugging tools) or below 5000 parallel processes (profiling tools). For debugging tools, runs with several hundred parallel processes would be of interest and for profiling purposes, full production runs in the order of ten-thousands of parallel processes should ideally be supported. Obviously this shows quite a gap between the typical usage and the targeted scale. Summarizing the answers to questions 19 and 20, the feedback to the above-mentioned tools is generally quite positive. Some of the more notable feedbacks are:

- The Extrae / Paraver toolset comes off quite badly in terms of ease of use (including configuration and installation), however it is considered a very powerful and feature-rich tool;
- The scalability of Score-P, Scalasca and Vampir is limited by the introduced instrumentation overhead, however very detailed profiling, tracing and load balancing information is generated;
- Allinea MAP introduces less overhead than e.g. Scalasca, but generates not so detailed profiles;
- Usage of gprof and gprof2dot is helpful in some situations, even though not designed for distributed applications;
- Dr. Hook, as it is directly built into the application (manual routine-level instrumentation), is easy to use and principally does not incur any scalability issues. However, functionality and degree of detailedness is limited.

Finally, looking at question 17, “For which analysis aspects do you use the above tools”, the most striking fact is that none of the nine ESiWACE representatives mentioned any tool used for Xeon Phi or GPU performance analysis. This, and also the answers to question 19 on the rating of performance tools (in particular with respect to the criteria accelerator support) indicate that currently there is not much experience in Xeon Phi and GPU profiling within ESiWACE yet, which could be an excellent opportunity for assistance from the side of PRACE-4IP WP7.

Scalable Libraries and Algorithms

Computational problems within applications of ESiWACE typically fall into the categories dense and sparse linear algebra, fast Fourier transforms and mesh generation. Note that mesh adaptation or repartitioning is currently not a topic, but might become one in the future.

With respect to numerical libraries, ESiWACE indicated the use of two libraries for discrete Fourier transforms:

1. FFTW, which internally provides thread- and node-level parallelism via POSIX threads, OpenMP and MPI; and
2. cufft, which is the NVIDIA CUDA Fast Fourier Transform library for GPU-based FFTs.

For FFTW, some more detailed feedback has been provided by ECMWF and IPSL: In current applications, both, thread- and node-level parallelism of FFTW is exploited. FFTW is also used on Xeon Phi coprocessors, however from the questionnaire answers it is not clear if FFTW is used directly or if only its API is linked against Intel's implementation within MKL (according to Fiona Reid and Ian Bethune PRACE White Paper "Optimising CP2K for the Intel Xeon Phi"[26], the latter one in general gives better performance). The currently achieved scalability of FFTW is in the order of ten thousands of processes, whereas the targeted scale for the near future is somewhere in the order of hundred thousands of processes. In terms of scalability and performance, FFTW has been rated as "very good" and in terms of portability, productivity and sustainability as "good". What is seen as very positive is the ability of FFTW to efficiently deal with vector lengths of arbitrary size, even if large prime factors are occurring. A minor issue for some applications however is that results are not bit-reproducible, unless batch sizes of one are used.

I/O Management Techniques

Within ESiWACE's application codes, several I/O libraries are currently used, that is: HDF5, PNetCDF and NetCDF4, XIOS, GRIB and GRIB2. The following received feedbacks to these libraries are worth mentioning:

- HDF5 received positive feedback throughout and is in particular valued for its portability and built-in compression support;
- PNetCDF achieves good scalability; however getting near-to-optimal performance is difficult;
- XIOS is valued for its efficiency and performance as well as for its ease of use.

With more specific focus on ESiWACE's applications, the I/O behaviour of their simulation codes has been characterised as follows: Essentially all codes read their input data only once at startup-time from not more than 50 input files, which typically make up a total data volume of up to 500 GB. Although some codes (at least partially) use serial input methods, which may turn out to be problematic on the road to Exascale, at the current stage this seems to not be a problem.

During the simulation, large files are written (file sizes of up to 200 GB) with an output frequency of approximately once per minute quoted in the questionnaire. Weather and in particular climate simulations are typically very long-running, so large numbers of output cycles are required. For example, a run of the climate model IPSLCM6 lasting for one year and covering a simulation timeframe of 3 to 4 thousand years is generating a total of 1.75 Petabytes (i.e. 350 GB every two wallclock-hours); IPSLCM6 is a full earth system model including atmosphere-land-ocean-sea ice model, carbon cycle, stratospheric and tropospheric aerosols chemistry. This (as well as explicit feedback from IPSL's representative) highlights high demands of ESiWACE and more generally the climate and weather community for efficient tools and powerful infrastructure for data management, data transfer and data archiving.

Lastly, due to the long-running nature of most simulations in this research field, the generation of checkpoint and restart data is fundamental. For three of the simulation codes (IPSLCM6, IFS and NEMO), the number of generated files per checkpoint is proportional to the number of cores being used (for IPSLCM6, it is even 20 files per process and checkpoint). Clearly this puts high pressure on the file- and storage-system and will not be a viable solution for future exascale systems. From this it is evident that efficient checkpointing techniques are a high requirement for ESiWACE, which may be worth further investigation during the exploitation phase of Task 7.2a.

3.6 MaX

General objectives and expected impact of the CoE

MaX (Materials design at the Exascale) is a CoE in materials' modelling, simulations, and design, created to endow researchers and innovators with powerful new instruments to address the key scientific, industrial and societal challenges that require novel materials.

Materials are crucial to scientific and technological advances and industrial competitiveness, and to tackle key societal challenges – from energy and environment to health care, information and communications, manufacturing, safety and transportation. The current accuracy and predictive power of materials' simulations allow a paradigm shift for computational design and discovery, in which massive computing efforts can be launched to identify novel materials with improved properties and performance; behaviour of ever-increasing complexity can be addressed; sharing of data and workflows accelerates synergies and empowers the science of big-data; and services can be provided in the form of data, codes, expertise, turnkey solutions, and a liquid market of computational resources.

This CoE is a user-focused, thematic effort supporting the needs and the vision of its core communities: domain scientists, software scientists and vendors, end-users in industry and in academic research, and high-performance computing centres. The CoE is structured along two core actions: (1) Community codes, their capabilities and reliability; provenance, preservation and sharing of data and workflows; the ecosystem that integrates capabilities; and hardware support and transition to exascale architectures. (2) Integrating, training, and providing services to core communities, while developing and implementing a model for sustainability, with the core benefit of propelling materials simulations in the practice of scientific research and industrial innovation.

HPC Applications and Requirements

MaX aims at enabling the exascale transition, by evaluating and putting in practice advanced programming models, novel algorithms, domain-specific libraries, in-memory data management, software/hardware co-design and technology transfer actions.

The exploitation of HPC systems at MaX will focus on testing and benchmarking materials science codes and workflows across a more or less evenly distributed combination of extreme-scale HPC systems, clusters for Big Data/ Data Analytics (Hadoop, Spark clusters), large shared memory systems, as well as data sharing in a cloud or Jupiter like framework. The CoE's HPC utilisation will typically target long-term goals and will be highly resource-intensive with interest in both capability and capacity computing.

Due to the diversity of the actions planned within the MaX CoE, there where will be requirements in general processing, the exploitation of accelerator hardware, storage and I/O solutions as well as HPC software and tools stacks. As expected, the CoE has a particular interest in exploiting advances in floating point computation, memory bandwidth and capacity, as well as low latency interconnects.

Both inter-node and intra-node performance are very important for the CoE. Unlike many other CoEs both resilience and energy efficiency are also deemed to be very important, which may indicate the extreme-scale nature of many of the simulations planned during the lifetime of MaX. The CoE plans to target x86, Power, Xeon Phi, NVIDIA GPGPUs and AMD GPGPU/APUs.

The most widely used large-scale HPC applications within MaX are Quantum Espresso, Fleur, SIESTA , Yambo and AiiDA, the majority of which have proven to be highly scalable on large-scale systems and where each code can run using either single or double precision (the latter is the most significant for MaX).

The MaX CoE seeks PRACE support through training and workshops, domain-related expertise, along with the porting and tuning of materials science codes.

Programming interfaces and standards

The following programming tools are relevant for the development of HPC applications within Max:

- For inter-node parallelization, MPI is most widely used. There seems to be no current interest in PGAS approaches such as Fortran CoArrays, UPC, Chapel or X10. For intra-node parallelization, i.e. threading, OpenMP appears to be the model of choice for all codes of interest.
- For the targeting of manycore hardware platforms (GPU / Xeon Phi), CUDA and OpenACC are of interest, where some of the codes (e.g., Quantum Espresso) have branches for targeting GPUs via CUDA and other library-based approaches. It is worth noting that there is also interest in exploiting the more experimental OmpSs directive-based programming model within MaX.
- For the above programming tools, productivity, openness (open standard), sustainability and portability are considered very important.

Debuggers and Profilers

There is clear evidence of interest in the exploitation of profiling tools for HPC applications within MaX, where there is particular interest in the Scalasca and Intel VTune profilers (TAU, HPCToolkit and manual profiling are also used, but less regularly).

MaX members have so far used Scalasca for performance profiling of codes on Xeon Phi and inter-node scaling, where its analysis of communication patterns is found to be particularly useful. The TAU tool has been used by MaX members to profile intra-node performance, communication patterns, as well as for correctness checking and debugging. Finally, the Intel VTune profiler has been used by MaX members to profile performance on Xeon Phi, intra-node performance as well as used for code correctness checking. In terms of reliability/accuracy all profiling tools are considered to be very good and only the VTune profiler is considered poor in terms of ease of use (it is worth noting that Intel have made many improvement in terms of the usability of the VTune over the last year). The one disadvantage of the Scalasca tool noted by MaX is that sometimes it can be too I/O intensive. In the case of TAU, there is a concern about the lack of documentation available on the tool.

Scalable libraries and algorithms

The computational problems within the codes of interest at MaX typically fall into the categories of dense linear algebra (Quantum Espresso, Yambo and Fleur), sparse linear algebra (SIESTA) and Fast Fourier transforms (Quantum Espresso, Yambo, SIESTA and Fleur). There is also a focus on database search within the AiiDA code.

In terms of 3rd party libraries, MaX currently aims to exploit ELPA (distributed memory), MAGMA (multi-threaded), FFTW (multi-threaded; targeting Xeon Phi). Using the ELPA library, codes within MaX have already been able to achieve good scaling across ~10,000 MPI processes. The ELPA library is scored highly by MaX in terms of scalability, performance, portability and productivity. However, the current downside of the library noted by MaX is that it seems to have no clear license, release plan or future, so suffers in terms of sustainability.

I/O Management Techniques

In the case of MaX members, I/O is not typically found to be a major performance bottleneck on current petascale systems (this may obviously change over the lifetime of the CoE). While high counts of input and output files are used per production run (up to 10,000 files for some codes), average file sizes tend to be small (10kB-1GB) and read and write frequencies per production run tend to be low (unless checkpointing is employed). While, input and output strategies for I/O are typically listed as “parallel”, the high number of files opened and closed during production runs suggests a one-file per process approach and possible high meta-data overhead for many codes of interest.

The MaX CoE has the desire for a common standard that would allow for the exchange of data between different materials science codes within the CoE and have identified HDF5 as a possible candidate due to its popularity.

3.7 NoMaD

General objectives and expected impact of the CoE

NoMaD, The Novel Materials Discovery Laboratory, aims to develop advanced tools for Big-Data Analytics to facilitate the discovery, creation and utilisation of new synthetic materials. Every new commercial product, be they smartphones, solar cells, batteries, transport technology, artificial hips, etc., depends on improved or even novel materials. Materials science and engineering is the exploration of how materials behave and how they may be utilised in technological systems. Computational materials science is increasingly influential as a method to identify such critical materials for Research and Development. This field is characterised by a healthy but heterogeneous ecosystem of many different codes that are used at all HPC centers worldwide, with millions of CPU hours spent every day, some of them at petascale performance. Enormous amounts of data are already stored in repositories scattered across Europe.

NoMaD enables access to this data and delivering powerful new tools to search, retrieve and manage it. It fosters sharing of all relevant data, building on the unique CECAM, Psi-k and European Theoretical Spectroscopy Facility (ETSF) communities, to put Europe ahead of materials science in other continents; Psi-k is a network of researchers working on the advancement of first-principles computational materials science. It aims to integrate the leading codes and make their results comparable by converting existing inputs and outputs into a common format, thus making these valuable data accessible to academia and industry. NoMaD plans to develop Big Data analytics for materials science. This will require novel algorithms, e.g., for statistical learning based on the created materials encyclopedia, offering complex searches and novel visualisations. It aims to become a crucial tool for atomistic simulations and multi-scale modelling in the physical, materials, and quantum-chemical sciences.

HPC Applications and Requirements

The initial focus of NoMaD will be to develop new Big Data analytics algorithms to identify gaps in the Materials Encyclopaedia. For this, clusters and large shared memory systems with visualisation equipment will be needed. Once developed the new analytics codes will be optimised and parallelised for use on HPC infrastructure; though this will not be needed in the first few years. This data mining and knowledge extraction will highlight relevant gaps in the existing data sets, the Materials Encyclopaedia. The second focus of NoMaD is to fill in these gaps by obtaining the missing data through dedicated simulation runs with the well known highly scalable simulation codes (e.g. Quantum Espresso, CP2K, FHI-aims, GROMACS) on Tier-0 or Tier-1 machines; FHI-aims is the Fritz Haber Institute ab initio molecular simulations package. However, as these exact needs are not known at this stage NoMaD was unable to complete questions 10 to 26 of the questionnaire. This highlights the need to re-engage with the CoE periodically to reassess the HPC requirements.

3.8 POP

General objectives and expected impact of the CoE

POP is a CoE in the area of Performance Optimisation and Productivity. High Performance Computing is becoming a fundamental tool for the progress of science and engineering and as such for economic competitiveness. The growing complexity of parallel computers is leading to a situation where code owners and users are not aware of the detailed issues affecting the performance of their applications. The result is often an inefficient use of the infrastructures.

Even when the need to get greater performance and efficiency is perceived, code developers may not have sufficient insight on causes of the poor performance to effectively address the problem. This may lead to fruitless attempts to restructure codes. POP will provide a service to its customers, the code developers, which will precisely assess the performance of computing applications running on a few hundred to many thousand processors. It will highlight the issues affecting the performance of the code and the most optimal way to alleviate them. The estimated population of such applications in Europe is 1.5 thousand. Within the 30 months project lifetime, POP aims to work with 150 of the codes.

HPC Applications and Requirements

As a rough estimate, POP anticipates around 10-20%, so 15-30 codes of the target codes, will require access to PRACE systems. At present, the 150 target codes have yet to be selected, as such POP will have need of PRACE support, but it will not know what will be involved (codes, libraries, I/O etc.) until they are selected. This prevented them from completing questions 17 to 27 of the questionnaire.

The support services for the POP customers are anticipated to last from one to a few months. POP would like to be able to apply for access to PRACE systems with a fast evaluation procedure. Currently, PRACE has dedicated 0.5% of PRACE resources to the CoEs in each call. Access is through the PRACE access calls. There *may* be no scientific evaluation needed, unless the CoE allocation is oversubscribed. While having only a technical review *might* speed up the evaluation to about 3 months, in the worst case scenario a POP customer could be waiting for six months for access to PRACE resources. A solution to this would be for POP to apply to PRACE for the anticipated core hours and then allocate as needed to their customers by means of short-lives user accounts.

Scalable libraries and algorithms

While members of the POP consortium develop debugger and profilers tools i.e Scalasca and Paraver/Dimemas, they indicate their status as developers rather than users would make their responses atypical and therefore misleading (note: Dimemas is an MPI performance analysis tool). However, POP is interested in the questionnaire responses by the other CoEs to the 'Debuggers and Profilers' section.

4 Recommendations & Conclusions

In this deliverable we have surveyed the current and anticipated HPC requirements of a wide range of European researchers, with a particular focus on domains and communities represented by the recently established European CoEs. We have surveyed these requirements by way of a comprehensive questionnaire, followed, in all cases, by direct interviews with points-of-contact at each of the CoEs.

Building on top of the work carried out in PRACE-3IP T7.2, this survey of HPC tools and techniques requirements has been carried out in the context of preparing European researchers for future extreme scale systems, as we continue to face the challenges of Exascale computing. With this challenge in mind, the survey, as reflected in the Task 7.2a questionnaire, has covered four separate classes of HPC tools and techniques that we continue to consider most relevant to prepare applications for near-term deep-petascale systems as well as future Exascale systems. Just as was the case in PRACE-3IP T7.2, the central aim of this initial "surveying phase" of Task 7.2a is to gain insight into how WP7 can best direct its expertise to help prepare and enable applications most relevant to European researchers.

We summarise our findings separately by topic: programming languages and standards, debuggers and profilers, scalable libraries and algorithms and I/O management techniques, where we focus on what we think are the most salient points to consider for the subsequent exploitation phase of Task 7.2a.

Where typical large-scale HPC codes are of interest to the CoEs, we have found that, unsurprisingly, the exploitation of the MPI model dominates, often supplemented by the shared memory OpenMP model. It is not entirely clear as to how many CoEs are already considering the modernisation of their codes with the latest version of both of these standards. Since both versions of the open standards have only been fully implemented by compilers and libraries very recently (and not at the time of PRACE-3IP T7.2), we recommend that the latest features of these widely employed standards should be investigated during the exploitation phase of Task 7.2a and with CoE applications in mind. In particular, implementations of the MPI 3 standard offer improvements in terms of new non blocking collective communications, improvements to one-sided communication semantics, new memory models, shared memory windows as well as other features. In terms of OpenMP that latest version of the standard now offers support for heterogeneous systems (via an offloading model) as well support for portable directive guided vectorisation.

As reflected in the answers to the Task 7.2a (and as anticipated in PRACE-3IP D7.2.1), the interest in OpenACC as a high-level directive based approach to target GPUs continues to grow and investigating the latest version of the standard should be considered as part of Task 7.2a exploitation activity where GPUs are being targeted. It is also worth highlighting again that the more experimental OmpSs programming model is of interest to some CoEs and should also be strongly considered when targeting heterogeneous platforms for identified workloads within those CoEs. In terms of more novel programming models (at least from the HPC perspective), such as PGAS languages, we have seen very little evidence of exploitation across the CoEs, but do note, once again that the ESiWACE CoE has shown interest in further exploiting Fortran Co-arrays, implementations of which are now more available since PRACE-3IP. We feel that these powerful languages should continue to be explored as part of the Task 7.2a exploitation phase, particularly with regard to finding opportunities for exploitation on real world workloads. Finally, we note that the CoeGSS CoE is interested in the Apache Spark cluster computing framework, which has an increasing amount of traction in the Big Data/analytics community and provides features that are of interest to the HPC community in terms of productivity, scalability and fault tolerance, and which should also be considered for further investigation within WP7.

As part of this report, we have surveyed the exploitation of some of the most commonly used profiling and debugging tools in HPC. In general, we have found that, although many of the European CoEs are aware of some of the more popular profiling tools, such as Scalasca, TAU, Vampir, HPCToolKit, Intel VTune and Allinea MAP, these tools are not exploited on a regular basis (the ESiWACE CoE is one exception to this rule). However, if the CoEs expect to achieve optimal performance on current and future extreme scale systems, they will need to take full advantage of the increased insight that these tools provide. With an anticipated increase in new memory types in emerging HPC platforms, as well as the ever-increasing importance of efficient vectorization, profiling tools are becoming more important to both the developer and user of community codes. One possible aim for Task 7.2a during the exploitation phase is to explore the capabilities of profiling tools at scale in more detail, targeting a range of real world CoE applications and to report experiences with the tools to the wider community.

As a consequence of the move towards large multi-petascale heterogeneous systems, there is an increasing demand for new and improved scalable, efficient, and reliable numerical algorithms and libraries that confront existing and upcoming complexities associated with such systems, including complex memory hierarchies, the overhead of data movement and fault tolerance.

The list of scalable libraries and algorithms in the Task 7.2a questionnaire was roughly divided into scalable numerical algorithms/methods (Direct Solvers, Iterative Solvers and FFT Libraries), higher-level libraries and other mesh/graph partitioning tools. The list by no means represented an exhaustive survey of scalable libraries and algorithms, but covered libraries and algorithms that were investigated in PRACE-3IP T7.2 and which were, and continue to be, of interest, mainly due to their potential to scale on extreme-systems. While, we found that many of the CoEs could not fill this section of the questionnaire in detail, some useful information was provided. In particular, we have found that there continues to be a huge amount of interest in scalable and efficient FFT methods and libraries across the CoEs, with several CoEs targeting scalability above 100,000 cores. Exploring scalable FFT implementations, particularly in the context of non-blocking collective communications now being available as part of MPI 3 implementations, should be considered during the Task 7.2a exploitation phase. Direct and indirect methods that rely on sparse linear algebra are also of significant interest across the CoEs and investigating efficient implementations of such methods within real world applications of interest will also be a key focus of Task 7.2a during the exploitation phase.

As pointed out in PRACE-3IP D7.2.1, the increasing data needs of European 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 user community within PRACE. The answers to the questions posed in the I/O section of the Task 7.2a questionnaire, strongly suggest that I/O performance along with the management of data will also be a pressing concern for many of the European Centres of Excellence (CoEs) when targeting current and future extreme-scale systems on Europe's road to Exascale computing. In fact, for several CoE's efficient I/O performance is considered to be one of the main keys to success. Many of the CoEs are keen to exploit parallel I/O libraries such as SIONlib, pNetCDF4, pHDF5 and XIOS, which were investigated as part of PRACE-3IP T7.2 activity and should be considered for further exploitation work as part of Task 7.2a. It is worth noting once again, that several CoEs have also highlighted the desire for a common standard that would allow for the exchange of data between different materials science codes and have identified HDF5 as a possible candidate due to its popularity. Supporting such a common standard across codes may be of interest to other CoEs, which may be worth exploring further as part of Task 7.2a activity. It has been noted that several communities, as represented by the CoEs, are developing their own I/O libraries which, to our knowledge, have so far not featured as part of PRACE exploitation work to date, and which we will aim to investigate further during the Task 7.2a exploitation phase.

Finally, in terms of future-looking approaches on the road to Exascale, WP7 should continue to keep in mind that economic realities will drive the architecture, performance, and reliability of the hardware that will comprise I/O architectures on extreme-scale systems. Due to this, the peak performance of I/O is expected to come at a premium, where specialised subsystems will be required to handle a ratio of burst to sustained I/O rates of at least an order of magnitude. One point for WP7 to keep in mind when looking to emerging techniques is that, since the work in PRACE-3IP, there has been an increasing interest in new tiered storage architectures in which solid state storage closely coupled to the HPC cluster fabric supports ever increasing performance requirements (the use of solid state storage is often leveraged as part of emerging burst buffer approaches, currently being evaluated on several European systems). On top of this, fault tolerance will be a dominant issue at all levels of the I/O stack, which is expected to impact the design of current and future European HPC applications.

In conclusion, it should be emphasised that all of the exploitation work proposed to be carried out during the next phase of Task 7.2a will continue to be informed and inspired by the on-going research across the various European and US (United States of America) exascale projects. 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, as we face into the exascale frontier.

5 Annex

In this annex we include the completed questionnaires from the eight CoEs

5.1 Blank Questionnaire

General questions

1. **What areas of science are the CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?**

2. **What are the expected implications and impact of the research conducted in the CoE?**

3. **What will the most important outcomes from your CoE?**
   - Scientific domain impact
   - Research papers
   - Software and tools
   - Software contributions to existing Open Source projects
   - Industrial or commercial applications
   - Other

4. **What kind of larger compute facility will be most relevant for the research conducted in your CoE?**
   - Supercomputers/High Performance Computing systems
   - Clusters for Big Data/ Data Analytics (Hadoop,spark clusters)
   - Large shared memory systems with or without visualisation equipment
   - Workflow, data sharing in a Cloud or Jyputer like framework
   - Combination of the above

5. **How would you describe the way in which your CoE need to use the systems to be most effective?**
   - Continuous
   - Sporadic
   - Short term goals (to achieve something within months)
   - Long term goals (to achieve something within years)
   - High intensive use of as much of the system resources as soon as possible
   - Low intensity use that is routine in terms of size or time to solution?

6. **Which of these modes of using the HPC-systems does your CoE deoend on?**
   - Capability
   - Capacity
   - Combination
   - Interactive
   - Batch

7. **What types of resources do will your CoE need?**
   - General processing, floating point
   - General processing, non-floating point (integer, text, image)
   - Use of accelerator hardware (GPUs, FPGAs, MIC, other)
   - Storage and I/O-requirements
   - Memory Size and bandwidth requirements

- Low latency interconnect
- Software or tools
- Local or remote access
- Do not know

8. What kind of support from PRACE will the CoE be relying on?

- General Assistance
- Training, workshops
- Online documentation
- Domain related expertise
- Computer expertise in code development, porting or tuning
- Do not foresee any need for support from PRACE

9. Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores):

Code Name	Comments

10. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

Code	Percentage of Peak	System	Comments

11. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments
Double precision (DP) only			
Single precision (SP) only			
Both SP and DP			
Other			

12. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments

Inter-node performance					
Intra-node performance					
Resilience					
Energy Efficiency					

Programming interfaces and standards

13. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI				
Fortran CoArrays				
Unified Parallel C (UPC)				
Chapel				
Global Arrays Toolkit				
X10				
OpenMP				
OpenCL				
CUDA				
OpenACC				
OmpSs				
Thread Building Blocks (TBB)				
Cilk Plus				
Spark				
Other				

14. How important are the following features of a programming tool to you?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Productivity					
Open Standard					
Sustainability					
Portability					

15. Which of the following platforms are your codes currently targeting or planning to target?

Platform	Yes	No	Comments
x86			

Power			
Xeon Phi			
NVIDIA GPGPU			
AMD GPGPU/APU			
Other			

Debuggers and Profilers

16. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	**Never**	**Seldom**	**Regularly**	**Often**
Scalasca				
TAU				
HPCToolkit				
…				
Manual (Console/Log-Output)				
Other tools:				
…				

17. For which analysis aspects do you use the above tools?

Tool	**Performance**				**MPI**		**Debugging**
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	
…							

18. At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes?

Tool	**Number of processes**		
	Typical usage	*Tool limitation*	*Targeted scale*
…			

19. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	**Ease of use**	**Reliability / Accuracy**	**Scalability**	**Accelerator support**	**Portability**	**Generality**
…						

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

20. Where do you see the strengths and most problematic limitations of the individual profiling tools?

Tool	**Pros**	**Cons**
…		

Scalable Libraries and Algorithms

21. Which categories of computational problems do your application codes deal with?

Code	**Linear algebra**		**FFTs**	**N-Body**	**Mesh**		**Other**
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
…							

22. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')?

Library	**Threading**	**Distributed**	**GPU**	**Xeon Phi**
ELPA				
Magma				
SuperLU				
MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW				
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

23. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years to each library?

Library	**Scalability (number of processes)**	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

24. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						
MUMPS						
DUNE						
FEAST						
MLD2P4/PSBLAS						
FFTW						
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

25. Where do you see the strengths and most important limitations of the individual libraries?

Library	Pros	Cons
ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

26. How can the IO behaviour of your application codes be characterized?

Considering your largest applications, please estimate for each of your application codes and for each of the categories “Input datasets”, “Output datasets” and “Checkpoints” (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
  - “once”: just one input cycle; all data is being read at the beginning.
  - “2/min”: two input cycles per minute.
- *Input strategy:*
  - S: “serial”: Master process reads and scatters all data.
  - P: “parallel”: Every process reads data.
  - H: “hybrid”: A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
  - “once”: just one output cycle; all data is being written at the end.
  - “2/min”: two output cycles per minute.
- *Output strategy:*
  - S: “serial”: Master process gathers and writes all data.
  - P: “parallel”: Every process writes output data.
  - H: “hybrid”: A subset of processes gathers and writes data.

Input Datasets:

Code	Number of Files	Avg. File size	Input frequency	Input strategy
…				

Output Datasets:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

Checkpoints:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

27. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

Library	Pros	Cons
HDF5		
PNetCDF		
XIOS		
SIONlib		
Other:		
…		

European Exascale Projects

28. Are the CoE's members acquainted with or have any of them contributed to any of the following projects?
- CRESTA
- DEEP or DEEP-er
- Mont Blanc or Mont Blanc2
- NUMEXAS
- EXA2CT
- EPIGRAM

29. Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?

5.2 BioExcel Questionnaire

Introduction

PRACE will provide computational services to the different Application Centres of Excellence (CoEs). The services will include compute time on PRACE supercomputers, training and access to competence. To better understand the requirements for the different CoEs, PRACE-4IP Work Package 4 and Work Package 7 would like to have technical discussion with different members of the CoE. As a guideline or agenda for the discussion, we would like to use the following questionnaire. The discussion can be conducted as a telecon or a Skype meeting.

The questionnaire covers question around training (from WP4) and application and technical questions (from WP), both sets relating to High Performance Computing.

General questions

30. What areas of science is your CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?

The expertise in BioExcel is in the area of biomolecular modelling and simulations. It includes usage of HPC/HTC applications for structural and functional studies of proteins, DNA, saccharides, membranes, solvents, small molecules, as applicable to fundamental research and also for industrial usage such as drug development, biotechnology, food industry, chemical industry.

31. What are the expected implications and impact of the research conducted in the CoE?

Improving the performance/efficiency/scalability of widely used software will greatly contribute towards the progress of biomolecular research. Improvements in the usability of workflow platforms for data handling and analysis will increase the productivity of researchers as well as uptake of advanced ICT technologies. Training and consultancy efforts will ensure the adoption of recommended tools and associated best practices, which will improve the efficiency of HPC resource usage. Successful implementation of sustainable operation will give opportunity for long-term application development and provisioning of support to the academic and industrial usage communities.

32. What will the most important expected outcomes from your CoE?

- Scientific domain impact
- Research papers
- Software and tools
- Software contributions to existing Open Source projects
- Industrial or commercial applications
- Other

All except for commercial applications

33. What kind of larger compute facility will be most relevant for the research conducted in your CoE?

- **Supercomputers/High Performance Computing systems**
- Clusters for Big Data/ Data Analytics (Hadoop,spark clusters)
- Large shared memory systems with or without visualisation equipment
- **Workflow, data sharing in a Cloud or Jyputer like framework**
- Combination of the above: please elaborate

34. How would you describe the way in which your CoE needs to use the systems to be most effective?

- **Continuous**
  - **This will be useful for production runs, but that's within the research group needs. The CoE could use the resource more for testing/benchmarking etc.**
- Sporadic
- **Short term goals (to achieve something within months)**
  - **This is more relevant, as described above**
- Long term goals (to achieve something within years)
- **High intensive use of as much of the system resources as soon as possible**
  - **That is needed for very-large-scale benchmarking**
- Low intensity use that is routine in terms of size or time to solution?

35. Which of these modes of using the HPC-systems does your CoE depend on?

- Capability
- Capacity
- **Combination**
- Interactive
- **Batch**

36. What types of resources from the following list will your CoE need?

- **General processing, floating point**
- General processing, non-floating point (integer, text, image)
- **Use of accelerator hardware (GPUs, FPGAs, MIC, other)**
- Storage and I/O-requirements
- **Memory Size and bandwidth requirements**
- **Low latency interconnect**
- **Software or tools**
- **Local or remote access**
- Do not know

37. What kind of support from PRACE will the CoE be relying on?

- **General Assistance**

  - o **When running benchmarks**
- **Training, workshops**
  - o **We should collaborate on aligning the traiing activities, specifically with PATC. BioExcel will be co-organizing a PRACE Spring School in 2017**
- **Online documentation**
  - o **Yes, link our knowledgebases**
- **Domain related expertise**
  - o **Yes, link our experts for advanced support/joint projects**
- **Computer expertise in code development, porting or tuning**
  - o **Yes, if PRACE could be helping with some specific platforms**
- Do not foresee any need for support from PRACE

38. Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores). If your CoE has not yet availed of resources at this scale, please list the codes that show best scalability:

Code Name	Comments
GROMACS	Depends on the system size but in principle it could use 100Ks cores
HADDOCK	This is a High-Throuput code, not really ready for 100K or beyond
CPMD	It could go to about 50K

39. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

Not sure

Code	Percentage of Peak	System	Comments

40. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments/Code
Double precision (DP) only	**sometimes**		
Single precision (SP) only	**Most of the time**		
Both SP and DP			
Other			

41. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Inter-node performance	X				
Intra-node performance	X				
Resilience			X		
Energy Efficiency				X	
Other (please specify)	Low-latency communication network is very important				

Programming interfaces and standards

42. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI	X			
Fortran CoArrays				
Unified Parallel C (UPC)				
Chapel				
Global Arrays Toolkit				
X10				
OpenMP	X			
OpenCL	X			
CUDA	X			
OpenACC				
OmpSs	X			
Thread Building Blocks (TBB)				Under consideration
Cilk Plus				
Spark				
Other				

43. How important are the following features of a programming tool to you?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Productivity	X				
Open Standard	X				

Sustainability	X				
Portability	X				

44. Which of the following platforms are your codes currently targeting or planning to target?

Platform	Yes	No	Comments
x86	X		
Power	X		
Xeon Phi	X		
NVIDIA GPGPU	X		
AMD GPGPU/APU	X		
Other	ARM		

Debuggers and Profilers

45. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	Never	Seldom	Regularly	Often
Scalasca				
TAU		Few tests		
HPCToolkit				
Extrae/Paraver			Would like to use regularly	
Manual (Console/Log-Output)				All the time
Other tools:				
…				

46. For which analysis aspects do you use the above tools (please list the tool)?

Tool	Performance				MPI		Debugging
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	
Extrae/paraver	X	X	X	X	X		
valgrind							X
DDT							X

47. At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes (please list tool)?

Tool	Number of processes		
	Typical usage	*Tool limitation*	*Targeted scale*
All tools	1-4 nodes	Don't know	Ideally until the scaling limit

48. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	Ease of use	Reliability / Accuracy	Scalability	Accelerator support	Portability	Generality
…						

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

49. Where do you see the strengths and most problematic limitations of the individual profiling tools (please list the tools)?

Tool	Pros	Cons
Most tools	Lost of functionality	Bugs prevent usage for complex scenarios, e.g. multi-node with mpi+openmp+cuda etc.

Scalable Libraries and Algorithms

50. Which categories of computational problems do your application codes deal with (please list the codes)?

Code	Linear algebra		FFTs	N-Body	Mesh		Other
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
…		X	X	X			

51. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')? If not listed, please add to the list.

Library	Threading	Distributed	GPU	Xeon Phi
ELPA				

Magma				
SuperLU				
MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW	X			
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

52. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years for each library? Please add to the list if a heavily used library does not appear:

Library	**Scalability (number of processes)**	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW	Adaquate	
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

53. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a)? Please add to the list if a heavily used library does not appear:

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						

MUMPS						
DUNE						
FEAST						
MLD2P4/PSBLAS						
FFTW	VG	VG	VG	G	VG	VG
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

54. Where do you see the strengths and most important limitations of the individual libraries? Please add to the list if a heavily used library does not appear:

Library	**Pros**	**Cons**
ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW	Adequate performance	
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

55. Considering your largest applications, please estimate for each of your application codes and for each of the categories “Input datasets”, “Output datasets” and “Checkpoints” (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
  - “once”: just one input cycle; all data is being read at the beginning.
  - “2/min”: two input cycles per minute.

- *Input strategy:*
  - S: “serial”: Master process reads and scatters all data.
  - P: “parallel”: Every process reads data.
  - H: “hybrid”: A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
  - “once”: just one output cycle; all data is being written at the end.
  - “2/min”: two output cycles per minute.
- *Output strategy:*
  - S: “serial”: Master process gathers and writes all data.
  - P: “parallel”: Every process writes output data.
  - H: “hybrid”: A subset of processes gathers and writes data.

Input Datasets:

Code	Number of Files	Avg. File size	Input frequency	Input strategy
…				

Output Datasets:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

Checkpoints:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

GROMACES uses a custom portable IO library called TNG. It comes with API with C/C++, FORTRAN support. The library is very lightweight but highly customisable and could be potentially useful to other projects. The CoE is willing to help with the adoption of the code, and if other projects find a need for parallel support in the library then PRACE will be in a very good position for providing this development as a service. How can the IO behaviour of your application codes be characterized?

56. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

Library	Pros	Cons
HDF5	Versatile	Too heavy (large for bundling in portable packages
PNetCDF		
XIOS		
SIONlib		
Other:		
…		

European Exascale Projects

57. **Are the CoE's members acquainted with or have any of them contributed to any of the following projects?**
    - **CRESTA**
    - **DEEP or DEEP-er**
    - Mont Blanc or Mont Blanc2
    - NUMEXAS
    - EXA2CT
    - **EPIGRAM**

58. **Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?**

Parts of the work in CRESTA are directly related to BioExcel objectivees

5.3 CoeGSS Questionnaire

Introduction

PRACE will provide computational services to the different Application Centres of Excellence (CoEs). The services will include compute time on PRACE supercomputers, training and access to competence. To better understand the requirements for the different CoEs, PRACE-4IP Work Package 4 and Work Package 7 would like to have technical discussion with different members of the CoE. As a guideline or agenda for the discussion, we would like to use the following questionnaire. The discussion can be conducted as a telecon or a Skype meeting.

The questionnaire covers question around training (from WP4) and application and technical questions (from WP), both sets relating to High Performance Computing.

General questions

59. **What areas of science is your CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?**

    All typical goals of Global Systems Sciences (GSS) are targeted, starting from life / social habits and pandemics up to green growth and urbanization. Furthermore, social media as well as financial market analysis are a very important topic as well.

60. **What are the expected implications and impact of the research conducted in the CoE?**

    The Centre of Excellence for Global Systems Science (CoeGSS) will build up an expertise centre to handle global problems in a more efficient way. For this purpose, fine-grained, precise and in-time simulations that rely on various unstructured data sources are mandatory to understand current or future conditions. Providing accurate data as well as multi-domain simulations will help to mitigate the decision risks and even further, will help to take effective actions that affect possibly the entire human mankind.

CoeGSS aims to drive the uptake of High Performance Data Analysis (HPDA) in conjunction with traditional HPC in order to provide large-scale accurate data sets, which will be used to build the basis for the simulation of GSS problem statements.

61. What will the most important expected outcomes from your CoE?

- Scientific domain impact
- Research papers
- Software and tools
- Other
  - Precise datasets to handle GSS problem statements
  - Pre-build and re-configurable synthetic populations for various kinds of problems
  - Marketplace to identify and handle problem statements (e.g. simulations, synthetic populations, datasets, etc.)
  - Consulting to solve GSS problem statements

62. What kind of larger compute facility will be most relevant for the research conducted in your CoE?

- Supercomputers / High Performance Computing systems
- Clusters for Big Data / Data Analytics (Hadoop, Spark clusters)
- (Large shared memory systems with or without visualisation equipment)

63. How would you describe the way in which your CoE needs to use the systems to be most effective?

- Continuous
- Short term goals (to achieve something within months)
- Long term goals (to achieve something within years)

64. Which of these modes of using the HPC-systems does your CoE depend on?

- Capability
- Interactive
- Batch

There is no dependency on architectures or operation modes, system capacity does not represent a key performance indicator.

65. What types of resources from the following list will your CoE need?

- General processing, floating point
- General processing, non-floating point (integer, text, image)
- Storage and I/O-requirements
- Memory Size and bandwidth requirements
- Low latency interconnect
- Software or tools
- Local or remote access

66. What kind of support from PRACE will the CoE be relying on?

- General Assistance
- Training, workshops
- Online documentation
- Domain related expertise
- Computer expertise in code development, porting or tuning

67. Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores).

If your CoE has not yet availed of resources at this scale, please list the codes that show best scalability:

Code Name	Comments
Open Source	Software to build up multi-agent-based simulations
PANDORA	
REPAST	
MASON	
Commercial / Not licensed	Software that implements GSS problem statements.
Simdemics	
Gleam	

The software packages are used to create multi-agent simulations to represent populations of a country, a continent or even the whole world.

68. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

Code	Percentage of Peak	System	Comments
- / -			

69. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments/Code
Double precision (DP) only	**X**		
Single precision (SP) only		**X**	
Both SP and DP		**X**	
Other		**X**	

70. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Inter-node performance		X			
Intra-node performance	X				
Resilience				X	
Energy Efficiency				X	

Other (please specify)				X	

Exascale does not only affect computational power, I / O as well as data analytics and data management are some of the key challenges that are of high importance.

Programming interfaces and standards

71. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI	X		All	
Fortran CoArrays		X		
Unified Parallel C (UPC)		X		
Chapel		X		
Global Arrays Toolkit		X		
X10		X		
OpenMP	X		All	
OpenCL		X		
CUDA	X	X		Data analytics with CUDA could be interesting.
OpenACC	X	X		Data analytics with OpenACC could be interesting.
OmpSs		X		
Thread Building Blocks (TBB)		X		
Cilk Plus		X		
Spark	X			
Other	X		Hadoop, Storm	Mechanisms and tools for data analytics.

CUDA and OpenACC do not represent a building block of this Centre of Excellence.

72. How important are the following features of a programming tool to you?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Productivity	X				
Open Standard	X				
Sustainability	X				

Portability	**X**				

73. Which of the following platforms are your codes currently targeting or planning to target?

Platform	**Yes**	**No**	**Comments**
x86	**X**		
Power	**X**		
Xeon Phi	**X**		
NVIDIA GPGPU	**X**		
AMD GPGPU/APU	**X**		
Other	**X**		**ARM**

Developments are not restricted, however most of the mentioned platforms are under investigation or planned. The different platforms will be addressed within the Co-Design endeavours.

Debuggers and Profilers

At the current point of time, the CoeGSS consortium is not able to answer this section in an appropriate manner. By coupling High Performance Computing with Global Systems Science, brand new applications that are able to cope with the different GSS requirements need to be developed from scratch. Although several applications are already scaling on thousands of cores, fundamental technologies are required to form a well-defined software architecture base.
So far, manual console outputs and their analysis seem to be suitable to tackle the ambitious development and performance optimization. However, as Python is widely used within the GSS community, High Performance Python with its cProfile will be important for early development phases.

74. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	**Never**	**Seldom**	**Regularly**	**Often**
Scalasca				
TAU				
HPCToolkit				
…				
Manual (Console/Log-Output)				
Other tools:				
…				

75. For which analysis aspects do you use the above tools (please list the tool)?

Tool	**Performance**				**MPI**		**Debugging**
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	

…							

76. At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes (please list tool)?

Tool	**Number of processes**		
	Typical usage	*Tool limitation*	*Targeted scale*
…			

77. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	**Ease of use**	**Reliability / Accuracy**	**Scalability**	**Accelerator support**	**Portability**	**Generality**
…						

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

78. Where do you see the strengths and most problematic limitations of the individual profiling tools (please list the tools)?

Tool	**Pros**	**Cons**
…		

Scalable Libraries and Algorithms

In Global Systems Sciences, traditional HPC problems are not highly relevant so that this section of the questionnaire is not filled in.
GSS applications widely use multi-agent-based systems in order to simulate the behaviour of objects, such as people, cars or power grids. Those agent-based systems do not rely on traditional mathematical algorithms or libraries, they model their behaviour based on empirical, statistical, spatial or even real-time data, just to name a few. Therefore, it is important to analyse various kinds of data sources, composed of different volumes and velocities. With the help of this data, a unique database can be created in order to build a suitable model (distribution and transition) for the agent based simulations.

In particular, there are agent-based simulations available that are built on coarse-grained databases and an insufficient amount of agents. As a consequence, only vague solutions for problem statements can be provided. Therefore, with the help of HPC and HPDA, data sources as well as the number of agents need to be increased in order to mitigate the risk for wrong decision-making.

79. Which categories of computational problems do your application codes deal with (please list the codes)?

Code	**Linear algebra**		**FFTs**	**N-Body**	**Mesh**		**Other**
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
…							

80. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')? If not listed, please add to the list.

Library	**Threading**	**Distributed**	**GPU**	**Xeon Phi**
ELPA				
Magma				
SuperLU				
MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW				
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

81. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years for each library? Please add to the list if a heavily used library does not appear:

Library	**Scalability (number of processes)**	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		

DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

82. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a)? Please add to the list if a heavily used library does not appear:

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						
MUMPS						
DUNE						
FEAST						
MLD2P4/PSBLAS						
FFTW						
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

83. Where do you see the strengths and most important limitations of the individual libraries? Please add to the list if a heavily used library does not appear:

Library	**Pros**	**Cons**
ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		

MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

IO Management will be the key to success for the CoeGSS project. In order to build the unique database and the models for the agent-based simulations, various unstructured data sources have to be analysed. However, in this early phase of the project, only rough estimations can be given for the volume, velocity and variety of data:

- Volume: up to several TB of data for input and output
- Velocity: up to 1 datum per second if real-time values are required
- Variety: up to 50 data sources forming a single agent-based model

Especially for the last bullet, the problem area as well as its size (population, localization, timeframe) regulate the amount of data. For initial developments, less data sources will be used to build the model.

Due to its early phase of the project, the CoeGSS consortium can only provide base ideas for data management, accurate measures will be provided after the first 12 months of the project lifetime.

84. How can the IO behaviour of your application codes be characterized?

Considering your largest applications, please estimate for each of your application codes and for each of the categories “Input datasets”, “Output datasets” and “Checkpoints” (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
  - “once”: just one input cycle; all data is being read at the beginning.
  - “2/min”: two input cycles per minute.
- *Input strategy:*
  - S: “serial”: Master process reads and scatters all data.
  - P: “parallel”: Every process reads data.
  - H: “hybrid”: A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
  - “once”: just one output cycle; all data is being written at the end.
  - “2/min”: two output cycles per minute.
- *Output strategy:*
  - S: “serial”: Master process gathers and writes all data.
  - P: “parallel”: Every process writes output data.
  - H: “hybrid”: A subset of processes gathers and writes data.

Input Datasets:

Code	Number of Files	Avg. File size	Input frequency	Input strategy
…				

Output Datasets:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

Checkpoints:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

85. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

Library	Pros	Cons
HDF5		
PNetCDF		
XIOS		
SIONlib		
Other:		
…		

European Exascale Projects

86. Are the CoE's members acquainted with or have any of them contributed to any of the following projects?
- CRESTA
- Mont Blanc or Mont Blanc2

87. Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?

Although FORTISSIMO and FORTISSIMO 2 are no direct Exascale projects, the findings of those projects may be transferred into the CoeGSS project in order to create a well-balanced and sustainable marketplace. Furthermore, the findings of CRESTA and MontBlanc as well as MontBlanc 2 may form the basis for efficient co-design approaches and in addition, novel architectures evaluation.

5.4 E-CAM Questionnaire

Introduction

PRACE will provide computational services to the different Application Centres of Excellence (CoEs). The services will include compute time on PRACE supercomputers, training and access to competence. To better understand the requirements for the different CoEs, PRACE-4IP Work Package 4 and Work Package 7 would like to have technical discussion with different members of the CoE. As a guideline or agenda for the discussion, we would like to use the following questionnaire. The discussion can be conducted as a telecon or a Skype meeting.

The questionnaire covers question around training (from WP4) and application and technical questions (from WP), both sets relating to High Performance Computing.

General questions

88. What areas of science is your CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?

ECAM WP1 - Classical MD Classical simulations are increasingly applied in fields ranging from nanoscience and materials development to biomolecular engineering and drug design. This is facilitated by the existence of broad, consolidated software packages (e.g., LAMMPS, DL_POLY, Charmm, NAMD, GROMACS, OPENMM, AMBER) obviating the need to constantly develop new programs and thus drastically reducing the effort to solve a particular problem. However, many processes of academic and industrial interest, such as protein folding, structural transitions of solid materials or study of phase behaviour for tailored material design, are still beyond standard simulation tools. In fact, disparate time scales (femto- seconds to milliseconds or even longer), originating from rare barrier crossing events, characterize these processes and require the accurate reproduction of the dynamics both on short and long time scales, imposing excessively long simulation times. *In the last few years, modeling of rare events has made tremendous progress and several computational methods have been put forward to bridge the time scale gap. However, these new approaches have not been included yet, with adequate efficiency and scalability, in common simulation packages, mostly because their application requires some specialized expert knowledge. WP1 will fill this gap.*

ECAM WP2 Electronic structure Electronic structure calculations have become a major component of R&D in industries such as aerospace, nuclear, pharmaceutical, and electronics. Several methods have been developed, applying different levels of approximation to the description of the electron manifold. WP2 will focus on two approaches: Quantum Monte Carlo

(QMC) and Density Functional Theory (DFT). The first holds the promise of an accurate solution of the electronic structure problem that can be systematically controlled and improved. The second offers an excellent compromise between accuracy and computational efficiency and is the most widely applied approach even though approximations (most notably the form of the exchange correlation functionals) introduce an element of uncertainty in its accuracy. *ECAM WP2 will focus on selecting software functionalities that are common to many electronic structure implementations, important for the coding and efficiency of codes, and mature enough to allow for a good definition of standards and interfaces. In most cases software from pre-existing methods will be used, when allowed by licenses, adapted to a library form, and improved as necessary. The results will be validated by comparison between DFT and QMC performances.*

ECAM WP3 - Quantum Dynamics. Methods for quantum time propagation are at an earlier stage of development compared to their classical counterparts. The key difficulty for all exact calculations in this field is that they scale exponentially with the number of degrees freedom, currently limiting access to times and sizes several orders of magnitude smaller than classical MD. On the other hand, quantum dynamical effects are increasingly important in many industrial sectors including hardware design (coherence and interference effects), pharmaceutics, and energy production (when light is used to induce quantum physical or chemical transformations). These applications motivate the development of approximate methods including, for example, semi-classical and hybrid quantum classical descriptions, that are in principle capable of modeling systems relevant for industrial processes. *In this context it is crucial to accompany methodological progress with the creation of a software and algorithmic infrastructure allowing a variety of quantum dynamics methods to run efficiently on massively parallel platforms. In fact, while several groups have developed in house codes, no well-established, user friendly tool exists that can be easily adapted to run user-driven calculations and can be efficiently ported to different architectures. WP3 will fill this gap.*

ECAM WP4 - Meso and Multi-scale. The inclusion of atomistic or electronic detail and the short time-steps required in most quantum and classical MD calculations limit the system size and the total time accessible with these methods. For phenomena of relevance to academia and industry that occur on longer time and distance scales (such as protein folding, polymer and surfactant structuring, lubrication and blood cell flow) it is useful to integrate out some of the underlying degrees of freedom and to develop coarse-grained models. These mid-scale or meso-scale models can be studied using suitably adapted simulation techniques from classical simulations and by developing new techniques that go beyond the particle-based description, for which E-CAM will develop and apply a number of new methods. Equally important and challenging is the requirement to work across more than one length or timescale at the same time, using multi-scale simulation techniques targeted at the production of new materials with tailored macroscopic properties (for example dislocations, grain boundaries, active sites). While considerable theoretical work exists in this domain, there is no generally accepted code in the community. *One of the goals of this WP will thus be to produce the necessary software by combining software modules. It will also produce software to bridge different descriptions (quantum, classical, continuum) in a sequential coupling scheme in which input parameters are computed at the higher resolution and then used in the lower resolution model.*

Understanding the behaviour of soft matter (or complex fluids) on vastly different length and time scales.

89. What are the expected implications and impact of the research conducted in the CoE?

The objective of the ECAm CoE is to create a European infrastructure for computational science in the area of the simulation and modeling of atoms, molecules and condensed phases. This infrastructure, E- CAM, will support innovation and leadership in industry and academia with applications ranging from the design of new materials to drug development, from energy research to quantum computing. This will be achieved through three complementary instruments: development, testing, maintenance, and dissemination of software targeted at end-users needs; advanced training of current and future key players in this arena; multidisciplinary, coordinated, top level applied consultancy to industrial end-users.

90. What will the most important expected outcomes from your CoE?

- Scientific domain impact

- Research papers
- Software and tools
- Software contributions to existing Open Source projects
- Industrial or commercial applications
- Other

As detailed above, E-CAM will cover four scientific areas (classical Molecular Dynamics (MD), electronic structure, meso and multi-scale modelling, and quantum dynamics) and will range from the development of new scientific ideas to algorithm development, optimisation, and parallelization. A large number of research papers in each of the four scientific areas will be produced during the course of the CoE. All software produced will be open-source. Portability and scaling on different hardware architectures will be managed through collaboration with the PRACE and other HPC centres. Post-docs and scientific programmers will be employed to oversee and implement the different stages of software development. The scientific programmers will be based at the four PRACE centres (ICHEC Ireland, Juelich Germany, IDRIS France, and CSC Finland) associated to this proposal and at the Hartree centre at Daresbury (specialising in industrial software output) to guarantee close collaboration with the current HPC European infrastructure. The project will not cover new hardware development, but specific actions will be set in place to interact and collaborate with hardware vendors to gain maximum competitive advantage for our end-users. Collaborations with other initiatives focused on technical infrastructures for HPC at the European level such as ETP4HPC3 will also be pursued (see WP74). This multidisciplinary approach will make it possible to address challenging problems over a wide range of applications, from life and material science to sustainable energy research. The broad potential impact of this approach is confirmed by the range of industrial partners that have agreed to participate in the initial phase of the project.

E-CAM's activities will stimulate, support and enhance simulation-based research and development (R&D) in industries (both large multinationals and SMEs). The creation and development of this infrastructure will, however, also impact academic research by creating a stable platform for scientific software development and optimisation, by creating strong links with industry and by training young researchers in computational skills not normally covered in their training. E-CAM will also provide a structure for the optimisation and maintenance of codes developed by specific research communities to guarantee a long life and a wide use of their outputs.

91. What kind of larger compute facility will be most relevant for the research conducted in your CoE?

- Supercomputers/High Performance Computing systems

92. How would you describe the way in which your CoE needs to use the systems to be most effective?

- Continuous and sporadic (both kinds of demands exist)

93. Which of these modes of using the HPC-systems does your CoE depend on?

- Combination (probably more capacity than capability)

94. What types of resources from the following list will your CoE need?

- General processing, floating point
- Use of accelerator hardware (GPUs)
- Software or tools Debuggers and profilers for parallel environments

- Local or remote access

95. What kind of support from PRACE will the CoE be relying on?
- General Assistance
- Training, workshops
- Online documentation
- Computer expertise in code development, porting or tuning

96. Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores). If your CoE has not yet availed of resources at this scale, please list the codes that show best scalability:

Code Name	Comments
GROMACS,	
LAMMPS	
ESPRESSO++	

97. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

We do not currently know the answer to this question.

Code	Percentage of Peak	System	Comments

98. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments/Code
Double precision (DP) only	Mostly DP		
Single precision (SP) only			
Both SP and DP	Sometimes		GROMACS
Other			

99. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Inter-node performance	X				
Intra-node	X				

performance					
Resilience		X			
Energy Efficiency			X		
Other (please specify)					

Programming interfaces and standards

100. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI	x		GROMACS, LAMMPS	
Fortran CoArrays		x		
Unified Parallel C (UPC)		x		
Chapel		x		
Global Arrays Toolkit		x		
X10		x		
OpenMP	x			
OpenCL	(x)			
CUDA	(x)			
OpenACC	(x)			
OmpSs		x		
Thread Building Blocks (TBB)		x		
Cilk Plus		x		
Spark		x		
Other				

101. How important are the following features of a programming tool to you?

Here, we see a difference of opinion across workpackages within E-CAM, with some feeling that sustainability and portability are somewhat important

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Productivity		x			
Open Standard	x				
Sustainability	x				
Portability	x				

102. Which of the following platforms are your codes currently targeting or planning to target?

Platform	Yes	No	Comments
x86	x		

Power	x		
Xeon Phi		x	
NVIDIA GPGPU	(x)		
AMD GPGPU/APU		x	
Other		X	

Debuggers and Profilers

103. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	Never	Seldom	Regularly	Often
Scalasca		x		
TAU		x		
HPCToolkit		x		
…				
Manual (Console/Log-Output)				X
Other tools:				
…				

104. For which analysis aspects do you use the above tools (please list the tool)?

Tool	Performance				MPI		Debugging
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	
…		X	X	X	X	X	X

However, in general debuggers and profilers are seldom used within E-CAM.

105. At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes (please list tool)?

Tool	Number of processes		
	Typical usage	*Tool limitation*	*Targeted scale*
Console	8-32	None	None

However, debuggers and profilers are seldom used within E-CAM.

106. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	Ease of use	Reliability /	Scalability	Accelerator support	Portability	Generality

		Accuracy				
Console	Neutral	good	poor	poor	Very good	good

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

However, debuggers and profilers are seldom used within E-CAM

107. Where do you see the strengths and most problematic limitations of the individual profiling tools (please list the tools)?

Tool	Pros	Cons
…		

However, debuggers and profilers are seldom used within E-CAM.

Scalable Libraries and Algorithms

108. Which categories of computational problems do your application codes deal with (please list the codes)?

Code	Linear algebra		FFTs	N-Body	Mesh		Other
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
GROMACSs,			yes	yes		yes	
LAMMPS			yes	yes		yes	
NAMD			yes	yes		yes	
ESPRESSO++			yes	yes			

109. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')? If not listed, please add to the list.

None of the following libraries are used within E-CAM

Library	Threading	Distributed	GPU	Xeon Phi
ELPA				
Magma				
SuperLU				

MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW				
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

110. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years for each library? Please add to the list if a heavily used library does not appear:

N/A, as none of the following libraries are used within E-CAM

Library	**Scalability (number of processes)**	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

111. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a)? Please add to the list if a heavily used library does not appear:

N/A, as none of the following libraries are used within E-CAM

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						
MUMPS						
DUNE						

FEAST						
MLD2P 4/PSBL AS						
FFTW						
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

112. Where do you see the strengths and most important limitations of the individual libraries? Please add to the list if a heavily used library does not appear:

N/A, as none of the following libraries are used within E-CAM

Library	**Pros**	**Cons**
ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

113. How can the IO behaviour of your application codes be characterized?

Considering your largest applications, please estimate for each of your application codes and for each of the categories “Input datasets”, “Output datasets” and “Checkpoints” (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
    - “once”: just one input cycle; all data is being read at the beginning.

    - "2/min": two input cycles per minute.
- *Input strategy:*
    - S: "serial": Master process reads and scatters all data.
    - P: "parallel": Every process reads data.
    - H: "hybrid": A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
    - "once": just one output cycle; all data is being written at the end.
    - "2/min": two output cycles per minute.
- *Output strategy:*
    - S: "serial": Master process gathers and writes all data.
    - P: "parallel": Every process writes output data.
    - H: "hybrid": A subset of processes gathers and writes data.

Input Datasets:

Code	**Number of Files**	**Avg. File size**	**Input frequency**	**Input strategy**
ESPRESSO++	<10	10-1000MB	once	Serial

Output Datasets:

Code	**Number of Files**	**Avg. File size**	**Output frequency**	**Output strategy**
ESPRESSO++	<10	10-1000MB	1/4	Serial

Checkpoints:

Code	**Number of Files**	**Avg. File size**	**Output frequency**	**Output strategy**
ESPRESSO++	< 10	10-1000MB	1/244	Serial

114. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

To our knowledge, E-CAM does not use any of the following I/O libraries

Library	**Pros**	**Cons**
HDF5		
PNetCDF		
XIOS		
SIONlib		
Other:		
…		

European Exascale Projects

115. Are the CoE's members acquainted with or have any of them contributed to any of the following projects?

- CRESTA
- DEEP or DEEP-er
- Mont Blanc or Mont Blanc2
- NUMEXAS
- EXA2CT
- EPIGRAM

No

116. Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?

No

5.5 EoCoE Questionnaire

Introduction

PRACE will provide computational services to the different Application Centres of Excellence (CoEs). The services will include compute time on PRACE supercomputers, training and access to competence. To better understand the requirements for the different CoEs, PRACE-4IP Work Package 4 and Work Package 7 would like to have technical discussion with the different members of the CoEs. As a guideline or agenda for the discussion, we would like to use the following questionnaire. The discussion can be done as a telecon or a Skype meeting.

The questionnaire covers question around training (from WP4) and application and technical questions (from WP), both sets relating to High Performance Computing.

General questions

1. **What areas of science are the CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?**

   EoCoE is structured mainly in five work packages. Four thematic pillars (Meteo4Energy, Water4Energy, Materials4Energy and Fusion 4Energy) gather experts from four different scientific communities. A transversal basis gives support to the four pillars in 5 different areas: numerical methods, linear algebra, parallel IO, optimization and usage of HPC tools.

2. **What are the expected implications and impact of the research conducted in the CoE?**

   CoEs are part of the infrastructure and, as such, are not supposed to make research. The manpower available in both the transversal basis and the pillars is mainly used to make possible the collaboration between application developers and HPC experts of the transversal basis. However, each research group have their own scientific targets and some long term activities on numerical methods or new software packages are planned in the transversal basis.

3. **What will the most important outcomes from your CoE?**

- Scientific domain impact
- Research papers
- Software and tools
- **Software contributions to existing Open Source projects**
- Industrial or commercial applications
- Other

EoCoE will mainly improve software during its life time. Some papers in HPC or in the different pillars' thematic are expected but will not be the main outcome.

4. **What kind of larger compute facility will be most relevant for the research conducted in your CoE?**

- Supercomputers/High Performance Computing systems
- Clusters for Big Data/ Data Analytics (Hadoop,spark clusters)
- Large shared memory systems with or without visualisation equipment
- Workflow, data sharing in a Cloud or Jyputer like framework
- Combination of the above

EoCoE is targeting mainly HPC systems for now.

117.

5. **How would you describe the way in which your CoE need to use the systems to be most effective?**

- Continuous
- Sporadic
- Short term goals (to achieve something within months)
- Long term goals (to achieve something within years)
- High intensive use of as much of the system resources as soon as possible
- Low intensity use that is routine in terms of size or time to solution?

The resources needed in EoCoE will mainly target development activities in roder to evaluate the performance of the new versions developed by the members of the projects. These runs are expected to be short in time but might require a large amount of resources when a high scalability is targeted for a code.

6. **Which of these modes of using the HPC-systems does your CoE deoend on?**
- Capability
- Capacity
- Combination
- Interactive
- Batch

Interactive would be the best but it is not realistic. We are used to batch system so it is fine. However a higher priority would allow us a better productivity as the time spent to wait for the result could be used for further developments.

7. **What types of resources do will your CoE need?**
- General processing, floating point
- General processing, non-floating point (integer, text, image)
- Use of accelerator hardware (GPUs, FPGAs, MIC, other)
- Storage and I/O-requirements
- Memory Size and bandwidth requirements
- Low latency interconnect
- Software or tools
- Local or remote access
- Do not know

8. **What kind of support from PRACE will the CoE be relying on?**
- General Assistance
- Training, workshops
- Online documentation
- Domain related expertise
- Computer expertise in code development, porting or tuning
- Do not foresee any need for support from PRACE

As explained in another PRACE document on trainings, our consortium gathers HPC experts and application developers. The experts give already trainings on performance analysis in coordination with the POP CoE. If training needs for application developers emerge, we will forward them to PRACE trainings.

Questions 10 to 28 are really specific to one specific application. We have 23 applications in the consortium and gathering this information for all of them is part of our activity and it is not done yet.
The questions on training have been answered in another PRACE document.

9. **Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores):**

Code Name	**Comments**

10. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

Code	Percentage of Peak	System	Comments

11. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments
Double precision (DP) only			
Single precision (SP) only			
Both SP and DP			
Other			

12. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Inter-node performance					
Intra-node performance					
Resilience					
Energy Efficiency					

Programming interfaces and standards

13. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI				
Fortran CoArrays				
Unified Parallel C (UPC)				
Chapel				
Global Arrays Toolkit				
X10				
OpenMP				
OpenCL				
CUDA				
OpenACC				
OmpSs				
Thread Building Blocks (TBB)				
Cilk Plus				
Spark				
Other				

14. How important are the following features of a programming tool to you?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Productivity					
Open Standard					
Sustainability					
Portability					

15. Which of the following platforms are your codes currently targeting or planning to target?

Platform	Yes	No	Comments
x86			
Power			
Xeon Phi			
NVIDIA GPGPU			
AMD GPGPU/APU			
Other			

Debuggers and Profilers

16. **For analyzing the performance of your application codes, which of the following tools do you use and how often?**

Tool	**Never**	**Seldom**	**Regularly**	**Often**
Scalasca				
TAU				
HPCToolkit				
…				
Manual (Console/Log-Output)				
Other tools:				
Likwid				

17. **For which analysis aspects do you use the above tools?**

Tool	**Performance**				**MPI**		**Debugging**
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	

18. **At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes?**

Tool	**Number of processes**		
	Typical usage	*Tool limitation*	*Targeted scale*
…			

19. **How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):**

Tool	**Ease of use**	**Reliability / Accuracy**	**Scalability**	**Accelerator support**	**Portability**	**Generality**
…						

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?

- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

20. Where do you see the strengths and most problematic limitations of the individual profiling tools?

Tool	Pros	Cons
…		

Scalable Libraries and Algorithms

21. **Which categories of computational problems do your application codes deal with?**

Code	Linear algebra		FFTs	N-Body	Mesh		Other
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
…							

22. **Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')?**

Library	Threading	Distributed	GPU	Xeon Phi
ELPA				
Magma				
SuperLU				
MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW				
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

23. **In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years to each library?**

Library	Scalability (number of processes)	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		

PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

24. **How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):**

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						
MUMPS						
DUNE						
FEAST						
MLD2P4/PSBLAS						
FFTW						
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

25. **Where do you see the strengths and most important limitations of the individual libraries?**

Library	**Pros**	**Cons**
ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		

PT-Scotch		
NetGen		
Other		

IO Management Techniques

26. How can the IO behaviour of your application codes be characterized?

Considering your largest applications, please estimate for each of your application codes and for each of the categories "Input datasets", "Output datasets" and "Checkpoints" (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
  - "once": just one input cycle; all data is being read at the beginning.
  - "2/min": two input cycles per minute.
- *Input strategy:*
  - S: "serial": Master process reads and scatters all data.
  - P: "parallel": Every process reads data.
  - H: "hybrid": A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
  - "once": just one output cycle; all data is being written at the end.
  - "2/min": two output cycles per minute.
- *Output strategy:*
  - S: "serial": Master process gathers and writes all data.
  - P: "parallel": Every process writes output data.
  - H: "hybrid": A subset of processes gathers and writes data.

Input Datasets:

Code	Number of Files	Avg. File size	Input frequency	Input strategy
…				

Output Datasets:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

Checkpoints:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

27. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

Library	Pros	Cons
HDF5		
PNetCDF		

XIOS		
SIONlib		
Other:		
…		

European Exascale Projects

28. **Are the CoE's members acquainted with or have any of them contributed to any of the following projects?**

- CRESTA
- DEEP or DEEP-er
- Mont Blanc or Mont Blanc2
- NUMEXAS
- EXA2CT
- EPIGRAM

29. **Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?**

5.6 ESiWACE Questionnaire

Answers from:

MPI-M
DKRZ (missing, sep. Document)
UK-MetOffice
IPSL France
Norway
SMHI
BSC

General questions

118. What areas of science are the CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?

Climate and Weather Science. The biggest challenges in these areas are being able predict weather and climate with increased local detail and to represent more processes. This is challenging because it is hard to get the models to scale well enough to give answers in a useful timeframe.

In terms of Scientific Computing and computational science it is
Scalability
Usability
Exploitability

Global climate and regional climate modelling and simulation

119. What are the expected implications and impact of the research conducted in the CoE?

ESiWACE pursues the following objectives: ESiWACE will substantially improve the efficiency and productivity of numerical weather and climate simulation on high-performance computing platforms. ESiWACE will support the end-to-end workflow of global Earth system modelling for weather and climate simulation in high performance computing environments. The European weather and climate science community will drive the governance structure that defines the services to be provided by ESiWACE. ESiWACE will foster the interaction between industry and the weather and climate community on the exploitation of high-end computing systems, application codes and services. ESiWACE will increase competitiveness and growth of the European HPC industry.

The CoE is doing work to support and improve research that will improve our ability to protect people from weather events and climate change either directly or by better planning for resilience.

120. What will the most important outcomes from your CoE?

- X, X, X Scientific domain impact
- X, X Research papers
- X, X Software and tools
- X Software contributions to existing Open Source projects
- Industrial or commercial applications
- Other

Directly, the CoE will improve software and tools leading to better scientific domain impact.

121. What kind of larger compute facility will be most relevant for the research conducted in your CoE?

- X, X Supercomputers/High Performance Computing systems
- X Clusters for Big Data/ Data Analytics (Hadoop,spark clusters)
- X, X Large shared memory systems with or without visualisation equipment
- (X) Workflow, data sharing in a Cloud or Jyputer like framework | *not so much Cloud based!*
- X, X Combination of the above

The community needs HPC and high performance storage system. The tasks in the CoE …. I do not know because the Met office work package does not directly need PRACE. But we do want to develop work-flow solutions that work in the PRACE environment.

122. How would you describe the way in which your CoE need to use the systems to be most effective?

- X, X, X Continuous
- X Sporadic
- X Short term goals (to achieve something within months)
- X Long term goals (to achieve something within years)
- X High intensive use of as much of the system resources as soon as possible
- X Low intensity use that is routine in terms of size or time to solution?

For our work-package. Low intensity and as continuous as possible to develop workflow solutions. But the people using the workflows will need high intensity resources at various times and there are demands now.

123. Which of these modes of using the HPC-systems does your CoE deoend on?

- X, X, X Capability
- X, X Capacity
- (X) , X Combination | *Logically! Follows from answers 1 & 2*
- X Interactive
- X, X Batch

Mostly capacity, but the community needs continuous access as time to solution is difficult.

124. What types of resources do will your CoE need?

- X, X, X General processing, floating point
- , X General processing, non-floating point (integer, text, image)
- (X) Use of accelerator hardware (GPUs, FPGAs, MIC, other)
- XX, X, X Storage and I/O-requirements
- X, X Memory Size and bandwidth requirements
- X, X Low latency interconnect
- X, X Software or tools
- X Local or remote access
- Do not know

High performance floating point. High IO. High memory and high memory performance.

125. What kind of support from PRACE will the CoE be relying on?

- X, X, X General Assistance
- X Training, workshops
- X, X Online documentation
- X, X Domain related expertise
- X, X, X Computer expertise in code development, porting or tuning
  - We need a demonstration of an efficient execution of benchmark with a coupled model. This MPMD/MPI/OpenMP benchmark is based on 3 different executables running simultaneously, each executable could be sequential or parallel (with MPI or an hybrid MPI/OpenMP approach). This type of execution is difficult to run without close cooperation with system administrators.
- Do not foresee any need for support from PRACE

General assistance and documentation

126. Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores):

Code Name	Comments
MPI-M	None that are used operationally
Python	And lots of sub-libraries
Cylc	http://cylc.github.io/cylc/
Rose	https://github.com/metomi/rose
DYNAMCO	On 60k CPU cores
EC-EARTH	Not yet tested at > 200k cores
NMMB/BSC-CTM	Not yet tested at > 200k cores
NEMO	Not yet tested at > 200k cores

I put these because workflow tools can allow us to use lots of resources and we do include these tools in the CoE, but it is probably not the point

127. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

Code	Percentage of Peak	System	Comments
MPI-M			Not known, but even for smaller scale typically below 5 %
Met Office Unified Model	~4 %	Cray XC 40	This varies massively depending on resolution and scale.

128. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments

Double precision (DP) only	X		
Single precision (SP) only			
Both SP and DP	X, X, X		Unified model solvers more often mixed precision these days.
Other			

129. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Inter-node performance	X, X	X, X			
Intra-node performance	X, X, X, X				
Resilience	X	X	X		Operational Centre
Energy Efficiency	X	X, X			

Programming interfaces and standards

130. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI	X, X, X, X, X, X, X		Models, EC-Earth ICON, MPI-ESM1, MPI-ESM2, and most codes running on mistral[1]	
Fortran CoArrays	(X)	X, X, X	EC-Earth	In pipeline
Unified Parallel C (UPC)		X, X, X, X		
Chapel		X, X, X		
Global Arrays Toolkit		X, X, X, X		
X10		X, X, X, X		
OpenMP	X, X, X, X, X, X, X		Models, EC-Earth	

[1] See https://www.dkrz.de/Nutzerportal-en/doku/mistral

OpenCL		X, X, X, X		May come with future ESM-components
CUDA	, X	X, X, X	Some satellite data processing	May come with future ESM-components
OpenACC	X	X, X, X	Gungho (dev. Phase)	May come with future ESM-components
OmpSs	X	X, X, X, X		
Thread Building Blocks (TBB)		X, X, X, X		
Cilk Plus		X, X, X, X		
Spark		X, X, X, X		
Other: MPMD/MPI/OpenMP Multi-Executable	3 executables X	x, X, X, X		

131. How important are the following features of a programming tool to you?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Productivity	X, X, X, X, X, X, X				
Open Standard	X, X, X	X, X, X, X			
Sustainability	X, X, X, X	X, X, X			
Portability	X, X, X, X, X, X, X				

132. Which of the following platforms are your codes currently targeting or planning to target?

Platform	Yes	No	Comments
x86	X, X, X, X, X, X, X		
Power	X, X, X, X, X	, X	
Xeon Phi	X, X, X, X, X	X	Planning, very soon, would like to perform some tests
NVIDIA GPGPU	X, X	X, X	Long term yes, May come with future ESM-components
AMD GPGPU/APU		X, X	Long term yes, May come with future ESM-components

Other	NEC SX, X	X	Long term - ARM

Debuggers and Profilers

133. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	Never	Seldom	Regularly	Often
Scalasca	X, X	X	X	Although re-investigating now.
TAU	X, X, X, X			
HPCToolkit	X, X, X, X			
…				
Manual (Console/Log-Output)			X, X	X, X, X
				Cray Pat
Other tools:	*Allinea MAF*	*Cray Pat Vtune*	*Dr. Hook*	*MAP will be trialled this year*
Vampir Trace			X	Allinea
Totalview			X	X
Alinnea DDT			X	
Alinea Perf report			X	
Vtune			X	
MPI perf snapshot		X		
Extrae/Paraver				X
Score-p/ VAMPIR		I have just started using it		

134. For which analysis aspects do you use the above tools?

Tool	Performance				MPI		Debugging
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	
Manual		X, X, X	X, X, X	X, X X, X	X, X	X, X	X, X, X
Vampir		X	X	X	X		
Allinea		X		X			
Alinnea DDT						X	X, X
Alinnea Perf report		X	X	X	X		
Vtune		X	X	X			

MPI Perf snapshot					X		
Totalview							X, X, X
Extrae/Paraver		X	X	X	X	X	X
Score-p (including Scalasca& VAMPIR)		x	x	x	x		
Allinea-MAP		x	x	x	x		

135. At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes?

Tool	Number of processes		
	Typical usage	*Tool limitation*	*Targeted scale*
…	200*36		400*36
Manual	100-5 000 / 1800	Full machine	50000 / 2048
Vampir	100-5 000		50000
Totalview	60, 1 < n < 256	60 (license)	1000
Performance			Full model
Extrae/Paraver	1 < n < 5k		
Score-p (w. Scalasca&VAMPIR)	64		1024
Allinea-MAP	60	? have not tried out	2048
Allinea DDT	24	? have not tried out	128

136. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	Ease of use	Reliability / Accuracy	Scalability	Accelerator support	Portability	Generality
Dr. Hook	Good	Good	Good	No	n/a	n/a
Scalasca	Ok	Good	Good	?	Fairly good, bit limited on AIX	
Vtune	Ok	Very good	Ok	Xeonphi	X86 only	
Manual	Good / Very good	Very good	Very good	n/a	good / Very good	good / Very good
Vampir	good		good		good	good
General debugging	Very good	n/a	Very	Not using	Issues on Cray	
Alinnea Perf reports	Very	n/a	Very	Not using	Issues on Cray	
MPI Perf snapshot	n/a	n/a	n/a	n/a	n/a	Intel only
Extrae / Paraver	Poor	Very Good	Very Good	Very Good	Very Good	Very Good
Gprof + gprof2dot	Good	Good	Very Poor	NA	NA	Poor
Score-p (w.	Very good	good	neutral	n/a	Very good	Very good

Scalasca &VAMPIR)						
Allinea-MAP	Very good	neutral	neutral	n/a	good	good
Allinea DDT	Very good	Very good	good	n/a	Very good	Very good

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

137. Where do you see the strengths and most problematic limitations of the individual profiling tools?

Tool	Pros	Cons
DrHook	Code is setup to use so very easy to use – typically not even a re-compile Good thread awareness at routine level Lots of scripts available to analyse output	No details of OpenMP within a subroutine No links to MPI timers No links to I/O metrics
Scalasca	Great display for Load balance and threading coverage	A bit awkward to use, and doesn't cover everything (e.g. issues with nested threading).
Manual		Very difficult (if possible) to use in MPMD/MPI/OpenMP mode
Vampir		Very difficult (if possible) to use in MPMD/MPI/OpenMP mode
Extrae / Paraver	Very powerful tool. Many different analysis views over very different metrics and counters. Supports multi executable simulations.	Difficult at the beginning.
Gprof + gprof2dot	Simple to get very useful and graphical outputs.	Difficult to analyse executions with large number of cores.
Score-p (which includes Scalasca and VAMPIR)	Very detailed profiling and tracing information generated, which is vital for performance analysis	The overhead introduced by the tool is limiting the number of processes which can be used during profiling
Allinea-MAP	Promises lower overhead, and can be used for large	The profiles produced are not so detailed. More light

	number of processes	weight information about run time distribution, due the sampling method used. It is very crucial to set the right sampling rate so as to get a proper profiling.

Scalable Libraries and Algorithms

138. Which categories of computational problems do your application codes deal with?

Code	Linear algebra		FFTs	N-Body	Mesh		Other
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
Unified Model / ICON	X	X					
4DVar	X						Minimisation
IPSLCM	yes		yes		yes	(yes)	

139. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')?

Library	Threading	Distributed	GPU	Xeon Phi
ELPA				
Magma				
SuperLU				
MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW	yes	yes		yes
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

140. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years to each library?

Library	**Scalability (number of processes)**	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW	OK for 100-5 000	50 000
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

141. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						
MUMPS						
DUNE						
FEAST						
MLD2P4/PSBLAS						
FFTW	good	good		good	good	
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

142. Where do you see the strengths and most important limitations of the individual libraries?

Library	**Pros**	**Cons**

ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW	OK	
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

143. How can the IO behaviour of your application codes be characterized?

Considering your largest applications, please estimate for each of your application codes and for each of the categories "Input datasets", "Output datasets" and "Checkpoints" (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
    - "once": just one input cycle; all data is being read at the beginning.
    - "2/min": two input cycles per minute.
- *Input strategy:*
    - S: "serial": Master process reads and scatters all data.
    - P: "parallel": Every process reads data.
    - H: "hybrid": A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
    - "once": just one output cycle; all data is being written at the end.
    - "2/min": two output cycles per minute.
- *Output strategy:*
    - S: "serial": Master process gathers and writes all data.
    - P: "parallel": Every process writes output data.
    - H: "hybrid": A subset of processes gathers and writes data.

Input Datasets:

Code	Number of Files	Avg. File size	Input frequency	Input strategy
Unified Model	1-20	50 MB – 50 GB depending on resolution of	Once in the main, but some configurations use periodic updates	Serial mostly. Some starting to use parallel.

		model run	that can be chosen by user.	
IPSLCM6[2]	50	1 GB	once	S and P
NorESM	*20-50*	*1 -10 GB*	*Once*	*S and P*
IFS	5	1.4GB	once	S
NEMO	3	8.2GB	once	S
NEMO-ORCA12	5	30Gb	once	hybrid

Output Datasets:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
Unified Model	1-50 (user can choose)	50 MB – 50 GB as above. User choice as to which variables.	User choice. Can be as often as once/timestep	Serial or Hybrid. User choice.
IPSLCM6	50	7 GB	1 each 2h, 1 each 10 mn and 1 each 1 mn	H
NorESM	*8-10*	*10 -200 GB*	*1/min*	*P*
IFS	2	8GB/simulated month	once per simulated month	S
NEMO-ORCA12	2	5Gb	1/15 min	hybrid
NEMO	6	4GB/simulated month	once per simulated month	H

Checkpoints:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
Unified Model	1-10	50 MB – 50 GB – as above	User choice.	Serial or Hybrid. User choice
IPSLCM6	20 * number of MPI procs	A total of 57 GB	1 each 2h	S and P
NorESM	*5-7*	*10-100 GB*	*1 per 3-12 hours*	*P*
IFS	#cores	10GB/restart	once per simulated month	P
NEMO	#cores	19GB/restart	once per simulated month	P
NEMO-ORCA12	1200	111Mb	once	parallel

144. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

Library	Pros	Cons
HDF5	Compression, In use,	

[2] 10 simulated years per day, 1 year run

	no comments, compression	
PNetCDF	Good scalability and implementation	Getting performance
XIOS	efficiency, easy to use, modular; In use, no comments: Good performance	Lead to some bugs
SIONlib		
Other:		
NetCDF4 (yes this is HDF5 underneath)	Standard in community	
GROB API	In use, no comments	

We need a lot of data management tool like cdos, nco, …
More generally, we need a clear data life cycle :

- scratch
- buffer
- archive (local to the Prace machine or remote with a large buffer space, efficient network links with our centre and useful tools)

European Exascale Projects

145. Are the CoE's members acquainted with or have any of them contributed to any of the following projects?

- CRESTA – acquainted with
- DEEP or DEEP-er – acquainted with
- Mont Blanc or Mont Blanc2 – acquainted with ; X
- NUMEXAS
- EXA2CT– acquainted with
- EPIGRAM

146. Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?

We are interested to follow these projects.

Some Earth models (NMMB/BSC-CTM or WRF) have been deployed and executed in Montblanc prototypes.

5.7 MaX Questionnaire

Introduction

PRACE will provide computational services to the different Application Centres of Excellence (CoEs). The services will include compute time on PRACE supercomputers, training and access to competence. To better understand the requirements for the different CoEs, PRACE-4IP Work Package 4 and Work Package 7 would like to have technical discussion with the different members of the CoEs. As a guideline or agenda for the discussion, we would like to use the following questionnaire. The discussion can be done as a telecon or a Skype meeting.

The questionnaire covers question around training (from WP4) and application and technical questions (from WP), both sets relating to High Performance Computing.

General questions

147. What areas of science are the CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?

Material Science, Physics, Chemistry and related areas of interest.

148. What are the expected implications and impact of the research conducted in the CoE?

The MaX CoE is a pan-European hub for developers and end users of advanced HPC applications for materials design and discovery. It will catalyse transnational research cooperation and technology transfer, boosting scientific knowledge and industrial competitiveness while providing a business model for the delivery of services to the community.

MaX aims at enabling the exascale transition, by evaluating and putting in practice advanced programming models, novel algorithms, domain-specific libraries, in-memory data management, software/hardware co-design and technology transfer actions.

149. What will the most important outcomes from your CoE?

- Scientific domain impact
- Research papers
- Software and tools
- Software contributions to existing Open Source projects
- Industrial or commercial applications

A combination of the above (20% research papers, 50% software and tools distributed in Open Source projects, 30% industrial and commercial applications)

150. What kind of larger compute facility will be most relevant for the research conducted in your CoE?

- Supercomputers/High Performance Computing systems
- Clusters for Big Data/ Data Analytics (Hadoop,spark clusters)
- Large shared memory systems with or without visualisation equipment
- Workflow, data sharing in a Cloud or Jupiter like framework
- Combination of the above

A combination of the above (more or less evenly distributed)

151. How would you describe the way in which your CoE needs to use the systems to be most effective?

- Continuous
- Sporadic
- Short term goals (to achieve something within months)
- Long term goals (to achieve something within years)
- High intensive use of as much of the system resources as soon as possible
- Low intensity use that is routine in terms of size or time to solution?

Long term goals

152. Which of these modes of using the HPC-systems does your CoE depend on?

- Capability
- Capacity
- Combination
- Interactive
- Batch

Combination of capability and capacity

153. What types of resources will your CoE need?

- General processing, floating point
- General processing, non-floating point (integer, text, image)
- Use of accelerator hardware (GPUs, FPGAs, MIC, other)

- Storage and I/O-requirements
- Memory Size and bandwidth requirements
- Low latency interconnect
- Software or tools
- Local or remote access
- Do not know

Due to the diversity of the actions planned within the MaX CoE, all the above described requirements will impact the achievement of our CoE goals (with special emphasis on floating point processing, memory and bandwidth, and low latency interconnects).
(AF: do you agree on the last detail? If not I am also happy with a general statement)

154. What kind of support from PRACE will the CoE be relying on?

- General Assistance
- Training, workshops
- Online documentation
- Domain related expertise
- Computer expertise in code development, porting or tuning
- Do not foresee any need for support from PRACE

The MaX CoE will mostly need PRACE support through Training and Workshops, Domain-related expertise and porting & tuning of Materials science codes.

155. Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores):

Code Name	Comments
Quantum ESPRESSO	http://www.quantum-espresso.org
Fleur	http://www.flapw.de/
Siesta	http://www.icmab.es/siesta/
Yambo	http://www.yambo-code.org/
Aiida	http://www.aiida.net; This is a tool aimed at building a software ecosystem in materials science, helping with automation, data storage and sharing

156. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

Code	Percentage of Peak	System	Comments

It is very difficult to properly answer this question. There are a plenty of different architectures, a lot of very different test-cases and 4 flagship codes. Any kind of provided value in this form would be disputable. Typically DFT-based codes can exploit between 40% and 60% of the PP.

157. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments
Double precision (DP) only			
Single precision (SP) only			
Both SP and DP	X		All the involved applications can run both in single and double precision. The latter is the most significant for realistic simulations
Other			

158. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Inter-node performance	X				
Intra-node performance	X				
Resilience	X				
Energy Efficiency	X				

Programming interfaces and standards

159. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI	X		ALL	
Fortran CoArrays		X		
Unified Parallel C (UPC)		X		

Chapel		X		
Global Arrays Toolkit		X		
X10		X		
OpenMP	X		ALL	
OpenCL		X		
CUDA	X		Some of the codes (QE, for example) have branches developed with CUDA.	
OpenACC	X		ALL	Some development is foreseen with OpenACC
OmpSs	X		ALL	Some development is foreseen with OmpSs
Thread Building Blocks (TBB)		X		
Cilk Plus		X		
Spark		X		
Other				

160. How important are the following features of a programming tool to you?

Metric	**Very Important**	**Somewhat Important**	**Of Little Importance**	**N/A**	**Comments**
Productivity	X				
Open Standard	X				
Sustainability	X				
Portability	X				

161. Which of the following platforms are your codes currently targeting or planning to target?

Platform	**Yes**	**No**	**Comments**
x86	X		
Power	X		
Xeon Phi	X		Preserving the portability of the codes, we don't restrict the development to any platform in particular
NVIDIA GPGPU	X		

AMD GPGPU/APU	X		
Other			

Debuggers and Profilers

162. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	Never	Seldom	Regularly	Often
Scalasca			X	
TAU		X		
HPCToolkit		X		
…				
Manual (Console/Log-Output)		X		
Other tools:				
Intel VTUNE			X	

163. For which analysis aspects do you use the above tools?

Tool	Performance				MPI		Debugging
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	
Scalasca	X		X		X		
TAU		X			X	X	X
Intel VTune	X	X				X	

164. At which scale (number of parallel processes) do you typically use the above tools?If you hit the tool's limitations, what would be the targeted scale for your analysis purposes?

Tool	Number of processes		
	Typical usage	*Tool limitation*	*Targeted scale*
…			

Typically, the number of parallel processes is limited by the amount of available resources rather than by the tool limitation.

165. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	Ease of use	Reliability / Accuracy	Scalability	Accelerator support	Portability	Generality
SCALASCA	good	Very good	neutral	n/a	Very good	Very good

Intel VTune	poor	Very good	n/a	Very good for MIC	Very poor	Very good
TAU	good	Very good	good	neutral	good	good

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

166. Where do you see the strengths and most problematic limitations of the individual profiling tools?

Tool	Pros	Cons
SCALASCA	Reliability, Ease of use	Sometimes too I/O
TAU	Reliability, Ease of use	Some lack of documentation
Intel VTUNE	Very accurate	Only for Intel archs

Scalable Libraries and Algorithms

167. Which categories of computational problems do your application codes deal with?

Code	Linear algebra		FFTs	N-Body	Mesh		Other
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
Quantum Espresso	X		X				
Yambo	X		X				
SIESTA		X	X				
Fleur	X		X				
AiiDA							Database search

168. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')?

Library	Threading	Distributed	GPU	Xeon Phi
ELPA		X		
Magma	X			
SuperLU				
MUMPS				
DUNE				

FEAST				
MLD2P4/PSBLAS				
FFTW	X	X		X
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

169. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years to each library?

Library	**Scalability (number of processes)**	
	Currently Achieved	*Targeted*
ELPA	O(10000)	
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

ELPA typically permits the performance to be improved by a factor of 2, letting the WHOLE APPLICATION, reach a scalability of the order of 10^4 cores (this also depends on the core architecture, btw).
For FFTW the problem is more arguable. Typically in DFT calculation the scalability regarding ONLY the FFT doesn't scale much over a single node.

170. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA	good	good	n/a	good	good	n/a
Magma						
SuperLU						
MUMPS						
DUNE						

FEAST						
MLD2P4/PSBLAS						
FFTW	neutral	good	n/a	good	good	n/a
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

171. Where do you see the strengths and most important limitations of the individual libraries?

Library	Pros	Cons
ELPA1/ELPA2	Enhances the scalability by a factor of 2	Not clear license, not clear release plan, not clear future.
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW	It works well	FFTW, FFTW3, MKL are compatible only using some tricks
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

172. How can the IO behaviour of your application codes be characterized?

Considering your largest applications, please estimate for each of your application codes and for each of the categories “Input datasets”, “Output datasets” and “Checkpoints” (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
  - "once": just one input cycle; all data is being read at the beginning.
  - "2/min": two input cycles per minute.
- *Input strategy:*
  - S: "serial": Master process reads and scatters all data.
  - P: "parallel": Every process reads data.
  - H: "hybrid": A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
  - "once": just one output cycle; all data is being written at the end.
  - "2/min": two output cycles per minute.
- *Output strategy:*
  - S: "serial": Master process gathers and writes all data.
  - P: "parallel": Every process writes output data.
  - H: "hybrid": A subset of processes gathers and writes data.

Input Datasets:

Code	**Number of Files**	**Avg. File size**	**Input frequency**	**Input strategy**
QE	10-10000	10-1000 kB	once/run	P
Yambo	10-1000	10-1000 kB	once/run	P

Output Datasets:

Code	**Number of Files**	**Avg. File size**	**Output frequency**	**Output strategy**
QE	10-10000	10-1000 kB	Once/run	P
Yambo	10-1000	10-1000 kB	Few times per run	P

Checkpoints:

Code	**Number of Files**	**Avg. File size**	**Output frequency**	**Output strategy**
QE	n/a	n/a	Few tens of times per run	n/a
Yambo	n/a	n/a	Few tens of times per run	n/a

173. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

Library	**Pros**	**Cons**
HDF5		
PNetCDF		
XIOS		
SIONlib		
Other:		
…		

Some codes (i.e. Yambo) use NetCDF, some others (i.e. QE) use ad-hoc libraries (iotk). The general aim is to achieve a common standard I/O library, that would probably be HDF5. In general, in DFT and material science codes the I/O is not a strong issue (about performances). What is more desirable is to have a common standard that would be permit to exchange data between different codes. HDF5 would be the natural candidate because of its popularity.

European Exascale Projects

174. Are the CoE's members acquainted with or have any of them contributed to any of the following projects?

- CRESTA
- DEEP or DEEP-er
- Mont Blanc or Mont Blanc2
- NUMEXAS
- EXA2CT
- EPIGRAM

The CoE's members have contributed to DEEP and DEEP-ER, MontBlanc and MontBlanc2 and EPIGRAM.

175. Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?

Knowledge acquired during the Exascale projects will be used during the MaX CoE. The experiences gathered in the other exascale projects permitted us to learn different approaches towards the exascale, in particular for what concerns the exploitation of programming paradigms. In our CoE we will try to use tasking approaches and multiple levels of parallelization that showed their importance in the achievement of the MontBlanc and DEEP results.

5.8 NoMaD Questionnaire

Introduction

PRACE will provide computational services to the different Application Centres of Excellence (CoEs). The services will include compute time on PRACE supercomputers, training and access to competence. To better understand the requirements for the different CoEs, PRACE-4IP Work Package 4 and Work Package 7 would like to have technical discussion with different members of the CoE. As a guideline or agenda for the discussion, we would like to use the following questionnaire. The discussion can be conducted as a telecon or a Skype meeting.

The questionnaire covers question around training (from WP4) and application and technical questions (from WP), both sets relating to High Performance Computing.

General questions

176. What areas of science is your CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?

Answer: NOMAD will bring together physicists, chemists, materials, and computer scientists, and industry to develop a Materials Encyclopaedia and advanced tools for Big-Data Analytics to facilitate the discovery, creation and utilisation of new materials.

177. What are the expected implications and impact of the research conducted in the CoE?

Answer: Materials science and engineering is the exploration of how materials behave and how they may be utilized in technological systems. New materials influence all aspects of our society, as *they are important in the development of essentially every new commercial product, be it for better or novel solar panels, harder surfaces, lighter metals, and countless other applications.*

178. What will the most important expected outcomes from your CoE?
- Scientific domain impact
- Research papers
- Software and tools
- Software contributions to existing Open Source projects
- Industrial or commercial applications
- Other

Answer: Support for design and development of new materials

179. What kind of larger compute facility will be most relevant for the research conducted in your CoE?
- Supercomputers/High Performance Computing systems
- Clusters for Big Data/ Data Analytics (Hadoop,spark clusters)
- Large shared memory systems with or without visualisation equipment
- Workflow, data sharing in a Cloud or Jyputer like framework
- Combination of the above: please elaborate

Answer: Clusters, large shared memory systems with visualisation equipment, and potentially also supercomputers for Big Data Analytics, long-term supercomputers for simulation runs to close information gaps in the Materials Encyclopaedia

180. How would you describe the way in which your CoE needs to use the systems to be most effective?
- Continuous
- Sporadic
- Short term goals (to achieve something within months)
- Long term goals (to achieve something within years)
- High intensive use of as much of the system resources as soon as possible
- Low intensity use that is routine in terms of size or time to solution?

Answer: Continuous, long-term, in several locations in Europe

181. Which of these modes of using the HPC-systems does your CoE depend on?
- Capability
- Capacity
- Combination
- Interactive
- Batch

Answer: All

182. What types of resources from the following list will your CoE need?
- General processing, floating point
- General processing, non-floating point (integer, text, image)
- Use of accelerator hardware (GPUs, FPGAs, MIC, other)
- Storage and I/O-requirements
- Memory Size and bandwidth requirements
- Low latency interconnect

- Software or tools
- Local or remote access
- Do not know

Answer: Nearly all

183. What kind of support from PRACE will the CoE be relying on?
- General Assistance
- Training, workshops
- Online documentation
- Domain related expertise
- Computer expertise in code development, porting or tuning
- Do not foresee any need for support from PRACE

Answer: Computer expertise in code development, porting or tuning

184. Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores). If your CoE has not yet availed of resources at this scale, please list the codes that show best scalability:

Code Name	Comments

Answer: a) Simulation codes for computational materials science for info gap closure in the Materials Encyclopaedia (e.g. QuantumEspresso, CP2K, FHI-aims, GROMACS, …)
b) Big Data Analytics codes under development

185. If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?

Code	Percentage of Peak	System	Comments

Remark:
Questions 9 to 26 can currently not be answered for NOMAD. The focus of NOMAD is Big Data Analytics the methods, for which, first have to be developed in the course of the project. Only after that, they can be implemented, and optimized also with the aim for usage on supercomputers. Only then will data mining and knowledge extraction be possible. Relevant gaps in the existing data sets - missing pieces of information for certain materials questions - will become evident. This information to fill these gaps shall then be obtained through

dedicated simulation runs with the well known highly scalable simulation codes on Tier-0 or Tier-1 machines. But support for these will only be needed later.

186. What level of floating point precision do you typically use for your simulations?

	Yes	No	Comments/Code
Double precision (DP) only			
Single precision (SP) only			
Both SP and DP			
Other			

187. How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Inter-node performance					
Intra-node performance					
Resilience					
Energy Efficiency					
Other (please specify)					

Remark: Questions 13 to 26 not so relevant for core activity of NOMAD CoE

Programming interfaces and standards

188. Which of the following programming tools do you exploit in the development of your codes?

Programming Tool	Yes	No	Codes	Comments
MPI				
Fortran CoArrays				
Unified Parallel C (UPC)				
Chapel				
Global Arrays Toolkit				
X10				
OpenMP				
OpenCL				
CUDA				
OpenACC				
OmpSs				

Thread Building Blocks (TBB)				
Cilk Plus				
Spark				
Other				

189. How important are the following features of a programming tool to you?

Metric	**Very Important**	**Somewhat Important**	**Of Little Importance**	**N/A**	**Comments**
Productivity					
Open Standard					
Sustainability					
Portability					

190. Which of the following platforms are your codes currently targeting or planning to target?

Platform	**Yes**	**No**	**Comments**
x86			
Power			
Xeon Phi			
NVIDIA GPGPU			
AMD GPGPU/APU			
Other			

Debuggers and Profilers

191. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	**Never**	**Seldom**	**Regularly**	**Often**
Scalasca				
TAU				
HPCToolkit				
…				
Manual (Console/Log-Output)				
Other tools:				
…				

192. For which analysis aspects do you use the above tools (please list the tool)?

Tool	**Performance**				**MPI**		**Debugging**
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	

…							

193. At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes (please list tool)?

Tool	**Number of processes**		
	Typical usage	*Tool limitation*	*Targeted scale*
…			

194. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	**Ease of use**	**Reliability / Accuracy**	**Scalability**	**Accelerator support**	**Portability**	**Generality**
…						

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

195. Where do you see the strengths and most problematic limitations of the individual profiling tools (please list the tools)?

Tool	**Pros**	**Cons**
…		

Scalable Libraries and Algorithms

196. Which categories of computational problems do your application codes deal with (please list the codes)?

Code	**Linear algebra**		**FFTs**	**N-Body**	**Mesh**		**Other**
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
…							

197. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')? If not listed, please add to the list.

Library	Threading	Distributed	GPU	Xeon Phi
ELPA				
Magma				
SuperLU				
MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW				
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

198. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years for each library? Please add to the list if a heavily used library does not appear:

Library	Scalability (number of processes)	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

199. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a)? Please add to the list if a heavily used library does not appear:

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						
MUMPS						
DUNE						
FEAST						
MLD2P4/PSBLAS						
FFTW						
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

200. Where do you see the strengths and most important limitations of the individual libraries? Please add to the list if a heavily used library does not appear:

Library	**Pros**	**Cons**
ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

201. How can the IO behaviour of your application codes be characterized?

Considering your largest applications, please estimate for each of your application codes and for each of the categories "Input datasets", "Output datasets" and "Checkpoints" (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
  - "once": just one input cycle; all data is being read at the beginning.
  - "2/min": two input cycles per minute.
- *Input strategy:*
  - S: "serial": Master process reads and scatters all data.
  - P: "parallel": Every process reads data.
  - H: "hybrid": A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:
  - "once": just one output cycle; all data is being written at the end.
  - "2/min": two output cycles per minute.
- *Output strategy:*
  - S: "serial": Master process gathers and writes all data.
  - P: "parallel": Every process writes output data.
  - H: "hybrid": A subset of processes gathers and writes data.

Answer: Biggest challenge is a global search in large data sets

Input Datasets:

Code	Number of Files	Avg. File size	Input frequency	Input strategy
…				

Output Datasets:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

Checkpoints:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

202. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?

Library	Pros	Cons
HDF5		

PNetCDF		
XIOS		
SIONlib		
Other:		
…		

European Exascale Projects

203. Are the CoE's members acquainted with or have any of them contributed to any of the following projects?

- CRESTA
- DEEP or DEEP-er
- Mont Blanc or Mont Blanc2
- NUMEXAS
- EXA2CT
- EPIGRAM

Answer: Some partners are involved

204. Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?

Answer: Not on short term, maybe long-term

5.9 POP Questionnaire

Introduction

PRACE will provide computational services to the different Application Centres of Excellence (CoEs). The services will include compute time on PRACE supercomputers, training and access to competence. To better understand the requirements for the different CoEs, PRACE-4IP Work Package 4 and Work Package 7 would like to have technical discussion with different members of the CoE. As a guideline or agenda for the discussion, we would like to use the following questionnaire. The discussion can be conducted as a telecon or a Skype meeting.

The questionnaire covers question around training (from WP4) and application and technical questions (from WP), both sets relating to High Performance Computing.

POP is a transversal CoE on Performance, Optimization and Productivity. POP consortium is a small team of experts in performance and optimization offering services to the community. Users and codes became customers of POP and do not participate in the project consortium.

Under this scenario, most of the requested data targeting the codes characterization is not applicable to POP because we cannot know in advance which codes we would like to run on PRACE systems.

That may not be the usual framework of other CoEs. We would like to request PRACE a dynamic and flexible use of the CPU time that can be allocated to POP. Our goal is to be able to offer to some selected POP users the possibility to run on PRACE systems. POP targets to work with 150 different codes during the 30 months of the project. A rough estimator of the codes we may like to offer the PRACE allocation may be around 10-20% (15-30 codes).

Some of the PRACE executions may be done by POP partners, but we would like to be able to offer the possibility to give a temporal account in the PRACE system to the POP users.

As the POP services last from one month to few months, we need to request PRACE to implement a fast evaluation procedure to POP requests that enables us to maintain the planned scheduling.

PRACE would define the data required to evaluate POP requests. We would appreciate the acceptance criteria is adapted to POP requirements where the goal may not be very large scale runs but detailed performance analysis of codes for SMEs that do not have resources, or for codes with poor scalability. Knowing the acceptance criteria we will do our best to apply only for studies that have a high score to be accepted so we work efficiently.

General questions

1. **What areas of science is your CoE covering and what are the most challenging scientific questions or problems that the CoE will examine?**

   We are a transversal CoE on Performance, Optimization and Productivity, Any area of science can benefit from POP CoE.

2. **What are the expected implications and impact of the research conducted in the CoE?**

   There is no research on POP, we offer service.

3. What will the most important expected outcomes from your CoE?

- Scientific domain impact
- Research papers
- Software and tools
- Software contributions to existing Open Source projects
- Industrial or commercial applications
- Other
-

Improvement on applications performance and increase on productivity

4. What kind of larger compute facility will be most relevant for the research conducted in your CoE?

- Supercomputers/High Performance Computing systems
- Clusters for Big Data/ Data Analytics (Hadoop,spark clusters)
- Large shared memory systems with or without visualisation equipment
- Workflow, data sharing in a Cloud or Jyputer like framework
- Combination of the above: please elaborate

That would depend on the users/customers that are not part of the project, so we do not know in advance.

5. How would you describe the way in which your CoE needs to use the systems to be most effective?

- Continuous
- ***Sporadic***
- ***Short term goals (to achieve something within months)***
- Long term goals (to achieve something within years)
- High intensive use of as much of the system resources as soon as possible
- Low intensity use that is routine in terms of size or time to solution?

6. Which of these modes of using the HPC-systems does your CoE depend on?

- Capability
- Capacity
- Combination
- Interactive
- Batch

That would depend on the users/customers that are not part of the project, so we do not know in advance.

7. What types of resources from the following list will your CoE need?

- General processing, floating point
- General processing, non-floating point (integer, text, image)
- Use of accelerator hardware (GPUs, FPGAs, MIC, other)
- Storage and I/O-requirements
- Memory Size and bandwidth requirements
- Low latency interconnect
- Software or tools
- Local or remote access
- ***Do not know***

8. **What kind of support from PRACE will the CoE be relying on?**
   - General Assistance
   - Training, workshops
   - Online documentation
   - Domain related expertise
   - Computer expertise in code development, porting or tuning
   - ***Do not foresee any need for support from PRACE***

9. **Please list the names of the most widely used open source codes in your CoE that can already exploit large-scale HPC systems (where large scale here means > 200k CPU cores). If your CoE has not yet availed of resources at this scale, please list the codes that show best scalability:**

   N/a as the codes are external to the project.

10. **If known, what percentage of peak performance (approximately) do your best performing simulations achieve on large-scale systems?**

    N/a as the codes are external to the project.

11. **What level of floating point precision do you typically use for your simulations?**

    N/a as the codes are external to the project.

12. **How important is the improvement of the following to the most widely used codes in your CoE as you prepare for exascale?**

    N/a as the codes are external to the project.

Programming interfaces and standards

13. **Which of the following programming tools do you exploit in the development of your codes?**

    I answer what we do support with the performance tools we develop. The codes that are not part of the project, so we do not know in advance.

Programming Tool	Yes	No	Codes	Comments
MPI	x			
Fortran CoArrays		x		
Unified Parallel C (UPC)		x		
Chapel		x		
Global Arrays Toolkit		x		
X10		x		
OpenMP	x			
OpenCL	x			
CUDA	x			
OpenACC		x		
OmpSs	x			
Thread Building Blocks (TBB)		x		
Cilk Plus		x		
Spark		x		
Other	SHMEM ptthreads			

14. How important are the following features of a programming tool to you?

Metric	Very Important	Somewhat Important	Of Little Importance	N/A	Comments
Productivity					
Open Standard					
Sustainability					
Portability					

That would depend on the codes that are not part of the project, so we do not know in advance.

15. Which of the following platforms are your codes currently targeting or planning to target?

I reply what the tools support. The codes that are not part of the project, so we do not know in advance.

Platform	Yes	No	Comments
x86	X		
Power	X		
Xeon Phi	X		
NVIDIA GPGPU	X		
AMD GPGPU/APU	X		
Other	Fujitsu Cray		

Debuggers and Profilers

POP will be interested on having access to the replies of the other CoEs to this section as we develop performance tools.

16. For analyzing the performance of your application codes, which of the following tools do you use and how often?

Tool	Never	Seldom	Regularly	Often
Scalasca			X	
TAU		X		
HPCToolkit		X		
…				
Manual (Console/Log-Output)		X		
Other tools:				
Intel VTUNE			X	

Within the consortium we develop Scalasca and Paraver/Dimemas tools. Questions 17 to 20 are not applicable to POP because being the tools developers our usage of the tools is not the one from a typical user and we are not the right persons to rate our tools or to point strengths and weaknesses.

17. For which analysis aspects do you use the above tools (please list the tool)?

Tool	Performance	MPI	Debugging				
	XeonPhi / GPU	*Intra-node*	*Inter-node*	*IO*	*Communication patterns*	*Correctness*	
Scalasca							
Paraver/Dimemas							

18. At which scale (number of parallel processes) do you typically use the above tools? If you hit the tool's limitations, what would be the targeted scale for your analysis purposes (please list tool)?

Tool	Number of processes		
	Typical usage	*Tool limitation*	*Targeted scale*
Scalasca			
Paraver/Dimemas			

19. How do you rate the individual profiling tools with respect to the following criteria (very poor, poor, neutral, good, very good, n/a):

Tool	Ease of use	Reliability / Accuracy	Scalability	Accelerator support	Portability	Generality
Scalasca						
Paraver/Dimemas						

- *Reliability / Accuracy:* How reliable and accurate are the tool's reported analysis results (according to your estimation).
- *Accelerator support:* Is the analysis of accelerated (Xeon Phi / GPU) kernels possible?
- *Portability:* Does the tool have strong prerequisites regarding system architecture and system environment?
- *Generality:* Does the tool cover only a very special analysis purpose? Is it bound to special programming languages (e.g. tools based on source-code-instrumentation)?

20. Where do you see the strengths and most problematic limitations of the individual profiling tools (please list the tools)?

Tool	Pros	Cons
Scalasca		
Paraver/Dimemas		

Scalable Libraries and Algorithms

This information has to be related to the codes we would like to run on PRACE systems. As the codes are not part of the project, we do not know in advance.

21. Which categories of computational problems do your application codes deal with (please list the codes)?

Code	Linear algebra	FFTs	N-Body	Mesh	Other		
	Dense	*Sparse*			*Generation / Partitioning*	*Adaptation / Repartitioning*	*<enter here>*
…							

22. Which libraries do your application codes use with respect to the above-mentioned categories? Which kind(s) of parallelism do you exploit when using these libraries (indicate with 'X')? If not listed, please add to the list.

Library	**Threading**	**Distributed**	**GPU**	**Xeon Phi**
ELPA				
Magma				
SuperLU				
MUMPS				
DUNE				
FEAST				
MLD2P4/PSBLAS				
FFTW				
FFTE				
PetSc				
Trilinos				
Zoltan				
ParMetis				
PT-Scotch				
NetGen				
Other				

23. In your applications, what scalability do these libraries currently achieve? What are the scalability requirements for the next two years for each library? Please add to the list if a heavily used library does not appear:

Library	**Scalability (number of processes)**	
	Currently Achieved	*Targeted*
ELPA		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

24. How do you rate the above libraries with respect to the following criteria (very poor, poor, neutral, good, very good, n/a)? Please add to the list if a heavily used library does not appear:

Library	Scalability	Performance	Resilience	Portability	Productivity	Sustainability
ELPA						
Magma						
SuperLU						

MUMPS						
DUNE						
FEAST						
MLD2P4/PSBLAS						
FFTW						
FFTE						
PetSc						
Trilinos						
Zoltan						
ParMetis						
PT-Scotch						
NetGen						
Other						

25. Where do you see the strengths and most important limitations of the individual libraries? Please add to the list if a heavily used library does not appear:

Library	Pros	Cons
ELPA1/ELPA2		
Magma		
SuperLU		
MUMPS		
DUNE		
FEAST		
MLD2P4/PSBLAS		
FFTW		
FFTE		
PetSc		
Trilinos		
Zoltan		
ParMetis		
PT-Scotch		
NetGen		
Other		

IO Management Techniques

26. How can the IO behaviour of your application codes be characterized?

Answers gerneral to performance tools

Considering your largest applications, please estimate for each of your application codes and for each of the categories “Input datasets”, “Output datasets” and “Checkpoints” (if applicable) the following IO characteristics:

- *Number of files:* The number of files being read or written per IO-cycle
- *Avg. file size:* Average file size
- *Input frequency:* The frequency of input cycles; e.g.:
  - “once”: just one input cycle; all data is being read at the beginning.
  - “2/min”: two input cycles per minute.
- *Input strategy:*
  - S: “serial”: Master process reads and scatters all data.
  - P: “parallel”: Every process reads data.
  - H: “hybrid”: A subset of processes reads and scatters data.
- *Output frequency:* The frequency of output cycles; e.g.:

- "once": just one output cycle; all data is being written at the end.
- "2/min": two output cycles per minute.

- *Output strategy:*
  - S: "serial": Master process gathers and writes all data.
  - P: "parallel": Every process writes output data.
  - H: "hybrid": A subset of processes gathers and writes data.

Input Datasets:

Code	Number of Files	Avg. File size	Input frequency	Input strategy
…	few	small	once	Serial /parallel

Output Datasets:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…	large	large	Buffer size vs. granularity	parallel

Checkpoints:

Code	Number of Files	Avg. File size	Output frequency	Output strategy
…				

27. Which of the following techniques or libraries do your application codes use for IO? Where do you see the strengths and weaknesses of these libraries?This information has to be related to the codes we would like to run on PRACE systems. As the codes are not part of the project, we do not know in advance

Library	Pros	Cons
HDF5		
PNetCDF		
XIOS		
SIONlib		
Other:		

European Exascale Projects

28. Are the CoE's members acquainted with or have any of them contributed to any of the following projects?

- *CRESTA*
- *DEEP or DEEP-er*
- *Mont Blanc or Mont Blanc2*
- *EXA2CT*

29. Will results from any of the above projects be used in the CoE, or any expected impact dependent upon reuse of results from a previous or ongoing Exascale project?

All the previous experience related with performance analysis and optimization may impact on our work within POP. Above that, it is not foreseen any special impact from the European Exascale Projects