E-Infrastructures
H2020- INFRAEDI-2018-2020

INFRAEDI-01-2018: Pan-European High Performance Computing infrastructure and services (PRACE)

PRACE-6IP

PRACE Sixth Implementation Phase Project

Grant Agreement Number: INFRAEDI-823767

D8.1

Formal funding decision on projects
Final

Version: 1.0
Author(s): Dr. Joost Vandevondele, ETH Zürich
Date: 31.05.2019

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: INFRAEDI-823767**	
	Project Title: PRACE Sixth Implementation Phase Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: < **D8.1**>	
	Deliverable Nature: < Report >	
	Dissemination Level: PU *	**Contractual Date of Delivery:** 31 / May / 2019
		Actual Date of Delivery: 31 / May / 2019
	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 2005/444/EC.

Document Control Sheet

Document	**Title: Formal funding decision on projects**	
	ID: D8.1	
	Version: <1.0>	**Status:** ***Final***
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2013	
	File(s): D8.1.docx	
Authorship	**Written by:**	Dr. Joost Vandevondele, ETH Zürich
	Reviewed by:	Dr. Oriol Pineda, BSC Dr. Florian Berberich, Jülich
	Approved by:	MB/TB

Document Status Sheet

Version	**Date**	**Status**	**Comments**
0.1	09/May/2019	Draft	
0.2	26/May/2019	Revised version	
1.0	31/May/2019	Final version	

Document Keywords

Keywords:	PRACE, HPC, Applications, Exascale, Software solutions

Table of Contents

List of Tables

References and Applicable Documents

[1] http://www.prace-project.eu

[2] https://ssl.linklings.net/applications/prace-6ip-wp8/

List of Acronyms and Abbreviations

ARM	ARM Limited (formerly Advanced RISC Machines Ltd.)
CoE	Centre of Excellence
EC	European Commission
HLST	High Level Support Team
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
SSC	Scientific Steering Committee
PRACE-6IP	PRACE Sixth Implementation Phase Project
WP	Work Package

List of Project Partner Acronyms

BADW-LRZ	Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften, Germany (3rd Party to GCS)
BILKENT	Bilkent University, Turkey (3rd Party to UHEM)
BSC	Barcelona Supercomputing Center - Centro Nacional de Supercomputacion, Spain
CaSToRC	The Computation-based Science and Technology Research Center (CaSToRC), The Cyprus Institute, Cyprus
CCSAS	Computing Centre of the Slovak Academy of Sciences, Slovakia
CEA	Commissariat à l'Energie Atomique et aux Energies Alternatives, France (3rd Party to GENCI)
CENAERO	Centre de Recherche en Aéronautique ASBL, Belgium (3rd Party to UANTWERPEN)
CESGA	Fundacion Publica Gallega Centro Tecnológico de Supercomputación de Galicia, Spain, (3rd Party to BSC)
CINECA	CINECA Consorzio Interuniversitario, Italy
CINES	Centre Informatique National de l'Enseignement Supérieur, France (3 rd Party to GENCI)
CNRS	Centre National de la Recherche Scientifique, France (3 rd Party to GENCI)
CSC	CSC Scientific Computing Ltd., Finland
CSIC	Spanish Council for Scientific Research (3rd Party to BSC)
CYFRONET	Academic Computing Centre CYFRONET AGH, Poland (3rd Party to PNSC)
DTU	Technical University of Denmark (3rd Party of UCPH)
EPCC	EPCC at The University of Edinburgh, UK
EUDAT	EUDAT OY

ETH Zurich (CSCS)	Eidgenössische Technische Hochschule Zürich – CSCS, Switzerland
GCS	Gauss Centre for Supercomputing e.V., Germany
GÉANT	GÉANT Vereniging
GENCI	Grand Equipement National de Calcul Intensiv, France
GRNET	Greek Research and Technology Network S.A., Greece
ICREA	Catalan Institution for Research and Advanced Studies (3rd Party to BSC)
INRIA	Institut National de Recherche en Informatique et Automatique, France (3rd Party to GENCI)
IST-ID	Instituto Superior Técnico for Research and Development, Portugal (3rd Party to UC-LCA)
IT4I	Vysoka Skola Banska - Technicka Univerzita Ostrava, Czech Republic
IUCC	Machba - Inter University Computation Centre, Israel
JUELICH	Forschungszentrum Juelich GmbH, Germany
KIFÜ (NIIFI)	Governmental Information Technology Development Agency, Hungary
KTH	Royal Institute of Technology, Sweden (3rd Party to SNIC-UU)
KULEUVEN	Katholieke Universiteit Leuven, Belgium (3rd Party to UANTWERPEN)
LiU	Linkoping University, Sweden (3rd Party to SNIC-UU)
MPCDF	Max Planck Gesellschaft zur Förderung der Wissenschaften e.V., Germany (3 rd Party to GCS)
NCSA	NATIONAL CENTRE FOR SUPERCOMPUTING APPLICATIONS, Bulgaria
NTNU	The Norwegian University of Science and Technology, Norway (3rd Party to SIGMA2)
NUI-Galway	National University of Ireland Galway, Ireland
PRACE	Partnership for Advanced Computing in Europe aisbl, Belgium
PSNC	Poznan Supercomputing and Networking Center, Poland
SDU	University of Southern Denmark (3rd Party to UCPH)
SIGMA2	UNINETT Sigma2 AS, Norway
SNIC-UU	Uppsala Universitet, Sweden
STFC	Science and Technology Facilities Council, UK (3rd Party to UEDIN)
SURFsara	Dutch national high-performance computing and e-Science support center, part of the SURF cooperative, Netherlands
TASK	Politechnika Gdańska (3rd Party to PNSC)
TU Wien	Technische Universität Wien, Austria
UANTWERPEN	Universiteit Antwerpen, Belgium
UC-LCA	Universidade de Coimbra, Labotatório de Computação Avançada, Portugal
UCPH	Københavns Universitet, Denmark
UEDIN	The University of Edinburgh
UHEM	Istanbul Technical University, Ayazaga Campus, Turkey
UIBK	Universität Innsbruck, Austria (3rd Party to TU Wien)
UiO	University of Oslo, Norway (3rd Party to SIGMA2)
UL	UNIVERZA V LJUBLJANI, Slovenia
ULIEGE	Université de Liège; Belgium (3rd Party to UANTWERPEN)
U Luxembourg	University of Luxembourg
UM	Universidade do Minho, Portugal, (3rd Party to UC-LCA)
UmU	Umea University, Sweden (3rd Party to SNIC-UU)
UnivEvora	Universidade de Évora, Portugal (3rd Party to UC-LCA)
UnivPorto	Universidade do Porto, Portugal (3rd Party to UC-LCA)

UPC	Universitat Politècnica de Catalunya, Spain (3rd Party to BSC)
USTUTT-HLRS	Universitaet Stuttgart – HLRS, Germany (3rd Party to GCS)
WCSS	Politechnika Wroclawska, Poland (3rd Party to PNSC)

Executive Summary

This deliverable reports on the successful selection of the projects that are funded in WP8. Following closely the process outlined in the proposal, the Scientific Steering Committee (SSC) of PRACE ranked proposals based on external expert review, that were received in an open call. The management board of PRACE-6IP followed the recommendation of the SSC and approved the top 7 ranked proposals immediately on 11.12.2018, and an 8th proposal after addressing review comments on 25.02.2019. The process was competitive, and thus reaching its goal of selecting high quality projects. Furthermore, swift communication of the decision has allowed partners to take the necessary steps to start the projects on time.

1 Introduction

Work Package 8 of PRACE-6IP focuses on 'Forward-looking Software Solutions' and has objective to deliver high quality, transversal software that addresses the challenge posed by the rapidly changing HPC pre-Exascale landscape. These challenges include the diversity of hardware and the software complexity. It will advance strategic and long-term projects, allowing for disruptive approaches to modernize HPC software. The main outcome is open source software in the form of libraries or significantly refactored codes.

To achieve the goal of selecting high quality projects, an open call with well-defined criteria has been published. This call allowed partners to elaborate project proposals that were ambitious in their aims, and detailed in their approach, in line with the latest technological roadmaps, and well supported by their user communities. For each project, a concise proposal has been received by the Scientific Steering Committee (SSC). These proposals were sent for external review by the SSC, in a process lead by the chair of the SSC. Reviewers motivated their grading, based on the criteria below, and concluded with a suggestion on funding the proposal (yes/maybe/no). Reviewers were known to the SSC, but anonymous otherwise. They were established scientists, experienced code developers, and compute centre staff, from a world-wide pool. This approach built on the established PRACE Peer Review Process. Based on their assessment of the external reviews the SSC autonomously ranked projects, with a summary motivating the ranking. This ranking and summary was provided to the MB who decided which projects to fund. As expected the funding decision (see below) followed the recommendations of the SSC.

These selection criteria included the scope, the approach, and the team and management as follows:

Scope (weight 40%):

- Provides 'forward looking' software solutions addressing: 1) The diversity of hardware (requires performance portability in codes) 2) The software complexity (requires separation of concerns in codes). Projects must be strategic (long term) and disruptive or 'high risk, high gain' approaches are encouraged.
- Delivers open source software (OSI approved license) that is adopted across major scientific software packages, in the form of libraries, runtimes, or significantly refactored codes (explicitly enabling sharing of components, or changing programming paradigm). Is

integrated in flagship codes of CoE or major application codes used across the PRACE Tier-0 infrastructure and future Exascale systems.

- Has a clear path to adoption, with stated interest from potential scientific users. Projects are well embedded in scientific communities, and with a significant impact on advancing capabilities for science in the area of extreme-scale computing.
- Is active in the area between HLST and CoE efforts. It does not overlap with the typical work of an HLST, which is shorter and focussed on helping (new) users to run on current machines, nor with the CoE efforts, which are even longer term, and likely to employ more scientific domain specific knowledge. WP8 projects are predominantly with a computer science or software engineering touch and transversal.

Approach (weight 35%):

- Takes into account that future Exascale hardware in Europe and worldwide will not be a single architecture, but a mix of multi-core (several vendors), many-core or GPU accelerated architectures, as well as possibly specialized (ASIC, FPGA) hardware. Memory solutions, IO solutions, and network topology between nodes will vary. Software should thus be adaptable, hide the underlying complexity of the hardware, and present a well-designed API to the scientific software developer.
- Addresses scalability of software across nodes and on complex nodes. Communication optimal (avoiding) strategies are recommended. It is expected that communication between nodes will be MPI (possibly asynchronous or one-sided), but task based programming for on-node parallelism is encouraged. Solutions that integrate well in the programming language adopted are to be preferred.
- Seizes the opportunity to modernize software leveraging software technologies employed, developed, and maintained by the big players in IT, avoids solutions specific to HPC. For example, a combination of userfacing Python with C/C++ backends receives the support from the data science market and thus has a lot of momentum.
- Takes software sustainability seriously, mostly uses industry standard tools, has issue tracking, continuous integration, validation and verification, documentation, public repository and website.

Team and management (weight 25%):

- Represents a commitment from the partners that is significant, and matching (‘in-kind’) resources on the same project are considered a plus.
- The team is composed of experts that are likely to deliver on the project’s goals. A track record of delivering on software projects is a plus. International collaboration to build a stronger proposal, exploit synergies, and to increase expertise is considered a plus. The team must remain agile, and able to deliver on the software goals, which requires close collaboration and coordination.
- The timeline of the projects is ambitious, but realistic. Risks are present, but identified, and a strategy for mitigation is present. While tasks should aim to succeed, not all might reach their ambitious targets.

From the submitted proposals, a subset has been selected in a transparent and competitive process. This deliverable reports on the process of selection and the funding decision on projects.

2 Process

The SSC has autonomously performed the ranking of the WP8 proposals based on a committee of five SSC members. The committee was led by Sinéad Ryan and further included Marina Bécoulet, Claudia Filippi, Matej Praprotnik and Mike Payne. The full SSC was available for further consulting. The proposal submission and review were implemented on the Linklings system (See url [2], managed by Tim Robinson, ETHZ), Joost VandeVondele (ETHZ) organized the process.

The SSC received 15 proposals for review. This is consolidated from the 34 letters of intent received, and reflected in the fact that most proposals are collaborations between several partners (the two largest proposals have 12 partners). . A project requested funding (person months) for the work. This funding consists of a part specifically allocated for WP8, but not assigned to a partner prior to the call ('Central PRACE' funding), which had to be matched by the applicants. Matching could be done by resources that were committed to WP8 by the partner prior to the call (and thus funded by PRACE) or in-kind (i.e. not funded by PRACE, but dedicated to the PRACE project). In terms of 'Central PRACE' resources requested, the largest proposal requests 597k Euro funding, the smallest 94k Euro. These resources are (at least) matched by partner committed resources and in-kind resources. The total resources requested by all proposals is 4494k Euro (Central) + 2239k Euro (Committed).

The process followed closely the agreed protocol outlined in WP8 of the PRACE-6IP proposal. In particular, it was based on external review, with external reviewers selected by the SSC. The selected reviewers included the suggestions made by the applicant, but were mostly selected by the SSC members based on their own expertise. Over 90 reviewers (on average 6 per proposal) had to be contacted to have appropriate coverage of the proposals (41 completed reviews). For a few projects, the SSC based the ranking on just two reviews, that were deemed sufficiently consistent, both internally and with the SSC's own assessment of the proposals. Reviewers were asked to grade the scope (4 aspects), the approach (4 aspects), the team and management (3 aspects) on a scale from 1 to 5 (5 being the highest rating), as well as to make a funding recommendation (fund, maybe fund, do not fund). A final average grade was obtained using the agreed weighting. These results are summarized in Table 2. In general, there was a good agreement between the grades given, and the reviewer funding recommendation made. Nevertheless, the SSC was explicitly given the freedom to rank proposals autonomously according to their expertise, as only the committee had a overview of all proposals.

The submission and review process was conducted according to the timeline agreed to in the written consultation of the partners. In particular: the platform for proposal submission opened May 30th and closed August 30th (first submission August 28th), proposals were sent for review from Sept. 13th, review deadline (extended) Nov. 1st, SSC ranking meeting Nov 15th, management board meeting for the funding decision Dec. 11th. Decision communicated to proposal applicants (Dec. 17th). Two revised proposals received (Jan 18), SSC reviewed and decided (Feb 21st).

3 Ranking and funding recommendation

The SSC has discussed the proposals at the ranking meeting in depth, and has unanimously come to a decision in the ranking and funding recommendation. The SSC has:

- Identified 7 out of 15 proposals (47%) to be very strong, having ambitious plans with very clear ideas on the implementation, with relevance to the exascale roadmap and good connection to the user community. These proposals also have the highest rating in external review (4.0 and higher). These are proposals 108 (Louhivuori, Martii), 109 (Alexandrou, Constantia), 111 (Laure, Erwin), 114 (Bianco, Mauro), 107 (Solcà, Raffaele), 110 (Bull, Mark), 105 (Gibbon, Paul). The SSC recommends funding.
- Identified 2 out of 15 proposals (13%) that are weaker in some aspects, but might be funded if reviewer comments are taken appropriately into account. The SSC suggests to have a resubmission of these proposals, possibly as part of a later call. These are proposals 115 (Merta, Michael) and 113 (Meyer, Norbert). In particular, proposal 115 is ambitious but misses technical detail on the implementation and could improve its connection to the application domain; proposal 113 insufficiently references other work available and might duplicate efforts. Care needs to be taken that this addresses ARM architectures that are relevant for the exascale roadmap.
- Identified 6 out of 15 proposals (40%) that should not be funded in their current form. These are proposals 119 (Böhm, Stanislav), 118 (Hammer, Nicolay), 112 (Kristensen, Mads), 106 (Borrell Pol, Ricard), 116 (Tekin, Adem), 117 (Aykanat, Cevdet). These proposals were considered as having insufficient impact on exascale, to be limited to specific codes, to lack a supporting community, to lack coherency and sufficient detail, or generally considered insufficiently strong.

The SSC suggested that the 7 strongest proposals should be funded at the requested resource level. It is understood that this does not fully exhaust the available resources. The SSC therefore recommended that a second call for proposals is organized, where resubmissions and new proposals can compete for these remaining resources. Such a competitive second stage will yield additional high quality proposals and will lead to the most effective use of resources.

4 Conclusion: Decision on resource allocation

The PRACE-6IP Management Board followed the ranking of the SSC and decided to fund the 7 strongest proposals, while the 2 next stronger proposals were asked to address the reviewer comments, and evaluated again by the SSC leading to one additional funded proposal (115, Merta Michael). As a result, 8 strong proposals can start immediately. The involvement of the awarded partners in the selected projects is shown in following table:

	Gibbon	Solcà	Louhivuori	Alexandrou	Bull	Laure	Bianco	Merta		Total:
JSC	60	30								90
LRZ				30						30
MPCDF			30							30
INRIA				30						30
EPCC					60					60
BSC					30					30
CSC			75							75
ETHZ-CSCS		75				45	75			195
KTH						60				60
CINECA		47								47
UiO							36			36
CASTORC				60						60
IT4I								62		62
UL	30	30	30							90
Total	90	182	135	120	90	105	111	62		

Table 1: Effort in Person Months by involved partners in the 8 funded projects

A small part of the budget for WP8 is not yet allocated, but is planned to be used for either 1-2 additional projects, based again on a review, following closely the procedure of the first call. This second call will be announced at the PRACE 6IP Kick-off meeting in Bratislava (May 2019), deadline for submission will be Aug 30 2019, ranking and decisions will be performed before Dec 2019, and project start will be Jan 2020.

5 Annex – Funded Project Summaries

5.1 Project Title: Particle kinetic codes for Exascale plasma simulation

Project Leader Paul Gibbon, Juelich Supercomputing Centre

Project Contributor Leon Kos, University of Ljubljana

Project Abstract

Plasma physics has long since been recognised as one of the potential drivers for exascale computing, for example through gyrokinetic modelling of ITER-scale magnetic fusion devices, or the study of laser-based particle beam and radiation sources with fully electromagnetic particle-in-cell (PIC) codes. The essentially unlimited degrees of freedom which characterise these domains leads to an unquenchable demand for high-fidelity, kinetic (non-fluid) descriptions of highly nonlinear phenomena in both of these domains. Particle simulation is a well-established technique which has spawned dozens of codes around the world, (eg BIT1, VPIC, VSIM, OSIRIS, REMP, EPOCH, SMILEI, FBPIC, GENE, WARP, PEPC), with varying degrees specialisation and accessibility. An example of a versatile, open-source tool is the UK-based EPOCH code, which with over 800 registered users worldwide, has rapidly become an indispensable workhorse for the laser-plasma community. Despite its popularity, the code is not yet capable of exploiting PRACE Tier-0 systems at scale, primarily due to its current pure MPI implementation, inflexible decomposition scheme, inefficien load-balancing and heavy I/O overhead. By leveraging advanced algorithmic features (scalable I/O, space-filling curves, task-based programming models) developed for our own PIC code (JUSPIC), we propose to implement key algorithmic kernels as a library to be used by codes like EPOCH. This work would be done in close cooperation with EPOCH's developer team in Warwick, thereby elevating this important community code to genuine pre-exascale capability. We also aim to add a complementary mesh-free algorithm within this library. Here we will make use of our considerable experience in applying scalable multipole algorithms to both electrostatic and recently 2D Darwin systems to develop a high-fidelity, 3D magnetoinductive code (ie: which includesself-generated magnetic fields but suppresses electromagnetic waves). While more expensive than conventional mesh-based particle simulation, this technique eliminates artificial grid-generated noise by design and offers a more direct method for treating collisions. Both approaches are complementary with different strengths in different regimes of collisionality and a combined library would make coupling of mesh-based and mesh-free methods easier. Strategically this work is expected to benefit not only future users of the Extreme Light Infrastructure (ELI), but will also provide a fully electromagnetic alternative approach for high-fidelity modelling of plasma- wall interactions in ITER, where gyrokinetics is no longer applicable – one of the major challenges acknowledged by the EuroFusion consortium and a key issue to be tackled by EoCoE, the Energy-oriented Centre of Excellence in the 2nd funding phase 2018-2020.

5.2 Project Title: Modernisation of Plasma Physics Simulation Codes for Heterogeneous Exascale Architectures

Project Leader Martti Louhivuori, CSC

Project Contributors Sebastian von Alfthan, CSC, Juhani Kataja, CSC Fredrik Robertsén, CSC Laurent Chôné, CSC / Aalto University Project Contributor Hermann Lederer, MPCDF Markus Rampp, MPCDF Tilman Dannert, MPCDF Leon Kos, University of Ljubljana Dejan Penko, University of Ljubljana Ivona Vasileska, University of Ljubljana Matic Brank, University of Ljubljana

Project Abstract

Numerical simulations are absolutely essential to address central open questions in plasma physics, from fusion energy to space weather. Of key importance is the understanding of the fundamental physical processes involved in plasma turbulence. Insight-providing simulations require enormous computational efforts. Prerequisite for adequate and efficient usability of next generation supercomputers in the pre-exascale and exascale era is to push respective codes in this field to the next structural level with respect to scalability, and portability to different HPC architectures. In this project, we address the software engineering challenge by focusing on three world-leading, and widely used European simulation codes in plasma physics. The codes are complementary concerning their functionalities, but they all have in common to rely on large grids of different dimensionality, ranging from 3D (ELMFIRE) to 5D (GENE) to 6D (Vlasiator). Essential tasks include the refactoring of the codes to make them more accessible and adaptable for modern programming methodologies, the implementation of task-based parallelism to achieve performance portability and scalability, and the knowledge transfer to share lessons learned and to encourage reuse of code parts in the plasma physics community and other science areas (e.g., computational fluid dynamics or astrophysics) with very similar numerical methods and related software engineering challenges. The refactoring task will lead to different types of restructuring in the three codes increasing the pool of new code parts for transversal activities and reuse options. As part of the refactoring task of GENE, the computationally intensive parts shall be reduced to calling a small number of kernels that can be ported to different architectures. For taking the step to exascale supercomputers, Vlasiator requires refactoring to support more scalable grids and task based parallelism to address solvers with vastly differing time- and length scales, leading to synchronisation and load imbalances. ELMFIRE requires modernization and modularization as enabling step for the exploitation of next generation supercomputers. The planned introduction of task-based parallelism will provide an efficient approach for heterogeneous parallel computing, both in terms of algorithms and computing tasks but also in terms of diverse hardware environments. The planned transfer of the knowledge gained through the code modernization experiences includes sharing lessons learned with developers of similar codes encouraging the use of external libraries and code reuse. The standard of plasma physics codes will be improved overall by sharing example implementations of common algorithms of plasma physics codes. To encourage this code reuse and the adoption of external libraries in the plasma physics community at large, beside the three development repositories of the target codes, a common git repository shall be provided hosting the development work related to all three codes.

5.3 Project Title Linear Algebra, Krylov-subspace methods, and multi-grid solvers for the discovery of New Physics (LyNcs)

Project Leader Constantia Alexandrou, CaSToRC, The Cyprus Institute

Project Contributors Emmanuel Agullo, Inria Simone Bacchio, CaSToRC, The Cyprus Institute/University of Cyprus Olivier Coulaud, Inria Juan J. Durrillo, Leibniz Supercomputing Center Jacob Finkenrath, CaSToRC, The Cyprus Institute Kyriakos Hadjiyiannakou, CaSToRC, The Cyprus Institute Luc Giraud, Inria Giannis Koutsou, CaSToRC, The Cyprus Institute Stéphane Lanteri, Inria Gilles Marait, Inria Michele Martone, Leibniz Supercomputing Center

1. Project Abstract

Sparse linear systems arise naturally in a diverse range of applications in science and engineering. Partial differential equations involving local operators arise in numerous physical systems described by e.g. quantum mechanics and electrodynamics. Discretely representing local operators as sparse matrices allows for iterative methods for their solution. This sparsity is well suited for distributed computing architectures, via domain decomposition of the problem space, which is why such sparse solvers have traditionally served as prototype High Performance Computing (HPC) applications, exhibiting good scaling properties on the world's largest supercomputers. However, on modern and upcoming architectures, especially at Exascale, the efficient solution of such problems will require disruptive approaches to parallelism, beyond simply domain decomposing the problem space. Our project, Linear Algebra, Krylov methods, and multi-grid API and library support for the discovery of New Physics (LyNcs), will address this challenge, pooling together software development efforts across Europe to provide communities with the next generation of parallel libraries for solving sparse linear systems at the Exascale. LyNcs is led by the Computation-based Science and Technology Research Centre (CaSToRC) of The Cyprus Institute which will join forces with partners from the French Institute for Research in Computer Science and Automation (Inria) and the Leibniz Supercomputing Center (LRZ). Within LyNcs, we will implement cutting-edge sparse linear solver algorithms, develop and prototype new Krylov and block Krylov solvers, and optimize existing parallel codes that implement a range of preconditioners for these solvers, such as multi-grid. The improvements proposed span all levels of the scientific application software stack, from the basic Sparse BLAS library to fully-fledged simulation codes. In particular, we target the Fast Accurate Block Linear krylOv Solver 1 (Fabulous) and its dependencies Starpu, Chameleon, Maphys, and Pastix 2 , which we will enable for lattice Quantum Chromodynamics (QCD), for Computational Chemistry, and Computational Electromagnetics. Lattice QCD community solver libraries QUDA and DDalphaAMG 3 will be further developed to enable optimized Krylov and block Krylov solvers and new preconditioners. At the lowest level, we will further develop the efficient sparse matrix support software librsb, its APIs and adapter libraries 4 , pursuing a tighter integration with the aforementioned packages. This library development and optimization activity will be accompanied by a coherent effort in designing, implementing, documenting, and maintaining an Application Programmer Interface (API), enabling various scientific user communities to build full-fledged scientific applications on top of these libraries. In particular, within LyNcs, we will directly support communities by linking to major application codes running on PRACE Tier-0 systems, including tmLQCD and openQCD 5 for lattice QCD, HORSE 6 for Computational Electromagnetics, and A-VCI 7 for Computational

Chemistry. To this end, developers and users of these community applications have expressed support for LyNcs, stating their willingness to employ our developments in Letters of Support attached to this application.

5.4 Project Title Performance portable linear algebra

Project Leader ETH Zurich

Project Contributor University of Ljubljana, Slovenia Jülich Supercomputing Centre (JSC) Cineca Consorzio Interuniversitario

1. Project Abstract

The time to solution of many scientific applications depends highly on dense linear algebra operations. A typical example is to be found in Materials Science where the computational time is dominated by the execution of complex linear algebra tasks such as the Hermitian eigenvalue problem solution. Similarly, in applications requiring non-linear optimization, a pivotal role is played by the solution of dense linear systems. In both cases the size of the operands involved is so large that the solver of choice has to necessarily operate over multiple computing nodes. The de-facto standard library for distributed linear algebra is ScaLAPACK, a library that has been developed in 1995, when supercomputers were based on nodes which had a single CPU core. Since then, the node architecture has evolved; nowadays, supercomputers are built upon multi-socked nodes, multi-core CPUs, and accelerators. The parallelism approach of the ScaLAPACK library does not perform well on these new architectures and the corresponding algorithms have not evolved to keep up with developments in hardware or modern solving strategies. Studies conducted on prototype implementations using alternative task-based runtime libraries show that significant performance improvements are possible; for example, using such task based system decreases the time to solution of the Cholesky factorization by a factor between 20% and 50%. Additionally, modern algorithms adopting communication minimization strategies show evidence of substantial performance improvements even in the case of already well-tuned kernels. A classic example is the matrix-matrix multiplication implemented in CARMA, which shows a factor of 2 improvement in performance compared to an optimized ScaLAPACK implementation. In light of the ongoing hardware evolution, the developers of ScaLAPACK have started a project to implement a new distributed linear algebra library, based on MPI and OpenMP tasks, called SLATE. It is our aim to work in close collaboration with the SLATE team on two main goals: 1) to develop dense eigensolvers to be integrated in SLATE, and 2) to implement and release an advanced version of the SLATE library based on the HPX library. Instrumental to the goals above is the need to simplify the transition of application software to the new modern libraries. This will be achieved by continuing the development of the Distributed Linear Algebra (DLA) interface started during the PRACE 5IP WP7 program, and writing a detailed guide containing information and instructions on how to optimize the performance of the routines contained in the DLA interface. On the other hand, new C++ software applications can fully profit from the HPX-SLATE interface based on the C++ std::future object, which will reduce the amount of synchronization necessary between distinct routines: introducing fine-grained task dependencies with futures allows a second routine to start even if the first routine is not completely finished. We will contribute with the integration of the libraries in scientific applications, such that they can exploit the performance of modern architecture supercomputers. We will focus on Quantum ESPRESSO, SIRIUS and CP2K, applications used by the material science community in scientific publications.

5.5 Project title Performance Portable Communication Layer for Grid Applications

Project lead Mauro Bianco Swiss National Supercomputing Centre (CSCS)

Project Contributor Marcin Krotkiewski University of Oslo, USIT, SIGMA2

1. Project Abstract

The objective of this project is to provide high-level, performance-portable communication primitives to perform halo-exchanges for massively parallel numerical applications that use underlying grids and meshes. Halo exchange is one of the fundamental components of parallel solvers of Partial Differential Equations (PDEs) based on Finite Difference, Finite Volume, and Finite Element methods. In short, the modeled spatial domain is discretized using a grid (or a mesh) and distributed among the Processing Elements (PEs) using some domain decomposition scheme. To satisfy the PDEs across the domain borders, the individual PEs have to exchange the border data with their spatial neighbors. Grid-based PDE solvers are amongst the largest global consumers of HPC compute resources, and amongst the most widely used numerical methods in scientific HPC, e.g., in atmospheric sciences, astrophysics, structural and mechanical engineering, automotive industries, geology, etc. In all these applications the halo exchange is performed often, impacting the scalability of the applications at large scales. Consequently, the halo exchange communication primitive is crucial for the efficiency and scalability of the entire application. Given the multitude of today's (and expected future) HPC architectures, and their heterogeneity, requiring performance portability at exascale is particularly important, yet notoriously difficult to attain. Different Pes not only have different compute capabilities and require different optimization techniques, but also exchange data using a variety of communication channels. It is hence not enough to deliver a performance-portable, high-level numerics library. A communication interface that can handle specialized data exchange procedures between the different components is necessary, in order to make the whole parallel application performance portable and scalable. To address these challenges we aim to develop GHEX - a next generation exascale-ready library to exchange the halo information in a generic, portable, efficient, and scalable manner. From the memory point of view it is now a standard that many PEs have access to the same, cache-coherent shared memory. For performance reasons it is often best that threads within a shared memory node access each others' data directly, without any explicit data copy. On the other hand, off-node communication, or in general communication between peers that work in different address spaces has to be implemented explicitly, whether through MPI calls, or some RDMA primitives. Implementing these contradicting requirements in a clean, hardware-oblivious, and future-proof API, while delivering an adequate performance on the multitude of today's architectures is a complex task. Moreover, for a wide adoption it has to be assured that the library addresses the actual needs of the users. To achieve these goals, the semantics of both the user-facing and the hardware-facing interfaces need to be carefully designed and tested in practice. To this end, GHEX will be developed in a close collaboration with strategic scientific communities, especially with developers of the COSMO weather and climate model, and the DISPATCH astrophysics task-based modeling framework. The developed software will be included in the open-source GridTools package, which is already used by several projects around the world.

5.6 Project Title LoSync - Synchronisation reducing programming techniques and runtime support

Project Leader Mark Bull, UEDIN

Project Contributor Vicenç Beltran Querol, BSC

1. Project Abstract

The LoSync project will focus on improving the scalability of applications by removing unnecessary synchronisation and serialisation, and full realising opportunities to overlap calculation and communication. To do this, we will make use of modern features of well-standardised APIs, to ensure portability and relevance. These techniques will include: Using OpenMP/OmpSs tasks with data dependency clauses. This will include not only expressing computation as tasks, but also communication, by wrapping MPI or GASPI library calls inside tasks. We will utilise the Task-Aware MPI (TAMPI) and Task-Aware GASPI (TAGASPI) interoperability libraries developed by the INTERTWinE project to make this as efficient as possible. MPI single-sided communication. Recent developments in MPI libraries have significantly improved the performance of single-sided communication to the point where its benefits can be realised in real applications. GASPI single-sided (put-notify) communication. This is a lightweight alternative to MPI single-sided communication which interoperates well with MPI, and offers different synchronisation semantics which can help remove serialisation constraints. A significant part of the work of this task will consist of implementing these techniques in key kernels of important applications such as Gysela, Flucs, IFS, Quantum Espresso, CP2K, iPIC3D, CASTEP, LULESH, NTCHEM, as well as, some of the applications involved in the DEEP-EST project. The project will also make use of smaller kernels and mini-apps, sourced from the PRACE CodeVault, the INTERTWinE project resource packs, and elsewhere as appropriate. In addition, work will be carried out on runtime library implementation to support this work, building on development in the INTERTWinE FET-HPC and POP CoE projects. This will include: Continuing the development of the INTERTWinE TAMPI and TAGASPI interoperability libraries to support interaction of MPI and GASPI with OpenMP/OmpSs tasks with dependencies. Continuing the development of the INTERTWinE Resource Management interface and its implementation. Exploring extensions to the OpenMP tasking model to support task dependencies on external events, task-nesting, fine-grained dependencies, weak dependencies and early release of dependencies, to avoid artificial synchronization and serialization effects. Making use of performance analysis tools and techniques developed in POP to identify optimisation targets in real applications where these techniques can be most beneficially applied.The project will engage with the relevant standards bodies for MPI, OpenMP and GASPI, to track relevant upcoming features, propose new features where necessary, and to produce prototype implementation to test these features. The project will engage with the application developer community through training courses, hands-on developer workshops, and targeted website materials.

5.7 Project Title NB-LIB: Performance portable library for N-body force calculations at the Exascale

Project Leader Prof. Erwin Laure, KTH

Project Contributor Prof. Erik Lindahl, KTH Prof. Berk Hess, KTH Dr. Mark Abraham, KTH

Dr. Victor Holanda Rusu, CSCS Prashanth Kanduri, MSc, CSCS

1. Project Abstract

A large number of scientific applications use particle interactions (e.g. Molecular Dynamics, Monte Carlo or multiscale simulations in life sciences or materials), and several smaller codes or combinations of codes have unique features. However, as computers have become more specialized, many codes have not been accelerated e.g. for GPUs and it is increasingly hard to maintain parallelization – which will make them increasingly difficult to use on next-generation PRACE systems. The goal of the NB-LIB project is to address this with a library of cutting-edge performance nonbonded interactions as well as a parallelization framework that can be used by all these applications, and where future acceleration, porting or library features will benefit all applications – which will make it easier to convince vendors to contribute such efforts. We will achieve this by separating the lower-level parts of the GROMACS code into an API of non-bonded force routines to enable re-use of widely used, highly portable, and performant HPC code instead of reinventing the wheel. This will benefit applications in multiple domains and leverage Exascale optimization underway in co-funded and collaborative projects. The library will be available via APIs in industry standard languages, including both python3 and C++11. This will permit domain scientist to use the API to prototype and deploy solutions for new N-body simulations rapidly, leveraging existing knowledge and best practice, rather than learning how to modify existing code. This will enable innovation across disciplines as we approach the Exascale era, while still providing familiar tools to HPC developers, rather than require knowledge of niche runtimes or languages. The project will work alongside co-funded PRACE IP6 and Swedish HPC projects, as well as distinct API efforts funded in the USA. It will provide efforts that will advantage both users of the NB- LIB library and API, and the underlying GROMACS simulation back end. The disruptive innovation within GROMACS will stimulate existing modernization and optimization efforts. It will evolve testable software seams to permit non-bonded force-calculation schedules to be tailored for correctness and performance for particular Exascale node structures and/or user workloads in a maintainable way. GROMACS is a widely used and highly targeted code, within PRACE and around the world, which will ensure sustainability of the library and its API. Once deployed the API will provide the leverage for future innovation, for example to deploy Exascale-suitable FMM implementations as drop-in replacements, so that user-level Python workflows will require no changes. The resulting library and API will be available under a highly permissive license such as the LGPL v2 used currently for GROMACS (no restrictions whatsoever on linking to the library). Sustainability will be assured through integration with existing GROMACS development workflows and roadmaps, and tests of the API functionality will be developed and deployed the same way. Core GROMACS developers are closely involved in the project to facilitate this. The KTH-CSCS collaboration has been ongoing for over a year, has already led to significant code changes, and would make a strong impact for N-body HPC simulation users through the NB-LIB project.

5.8 Project Title FEM/BEM based domain decomposition solvers

Project Leader IT4Innovations

1. Project Abstract

The aim of this project is to extend the domain decomposition library ESPRESO to support highly scalable solution of problems in complex domain using finite/boundary element tearing and interconnecting (FETI/BETI) nonoverlapping domain decomposition method, thus enabling solution of large scale sound scattering and harmonic analysis problems. The current implementation in ESPRESO (similarly as in most other solvers) focus on the elasticity or the heat transfer problems for which it features excellent parallel and numerical scalabilities. The implementation for complex problems (such as solution of the Helmholtz equation) will be based on the FETI-H/FETI-DPH (FETI-Helmholtz/FETI-Helmholtz dual-primal) approach where the regularization is done using the complex interface mass matrix and the preconditioning is based on the plane wave deflation. To overcome the main bottleneck of the FETI method, solution of the coarse problem, the hybrid version of the method (HFETI/HTFETI) will be used which reduces the size of the coarse problem by grouping subdomains into clusters. The goal is to provide a modern, modular and portable code written in C++, parallelized on all possible levels, and capable of utilizing the most powerful supercomputers. Basic refactoring of the original ESPRESO code providing wrappers to external libraries such as BLAS or parallel direct solvers, which should enable faster development, will be part of the project. The development will take into account heterogeneous nature of current and future supercomputers. The code will be parallelized in a hybrid manner in shared and distributed memory and will benefit from the SIMD (Single Instruction Multiple Data) vector capabilities of modern CPUs. The parallelization in shared memory and vectorization will be done using the OpenMP pragmas. The computationally most demanding parts of the code will be accelerated using state of the art GP-GPUs. MPI will be used for a distributed memory parallelization. The implementation will be made available to developers of external libraries using a public API with extensive documentation and, moreover, the solver will be incorporated into a “Solver as a service” platform at IT4Innovations, thus enabling scientists and engineers with zero experience in HPC to leverage the power of supercomputers.

	Num		Review Averages													Decision Status				
	C	Revs	SCP1	SCP2	SCP3	SCP4	APP1	APP2	APP3	APP4	TAM1	TAM2	TAM3	REC	WAvg	A	U	UE	R	AR
prop108 Louhivuori, Martti (CSC); von Alfthan, Sebastian (CSC); K...	1	2	5.0	5.0	5.0	4.5	5.0	4.5	3.5	5.0	4.5	4.5	4.5	3.0	4.7	●	○	○	○	○
prop109 Alexandrou, Constantia (The Cyprus Institute, University...		3	4.0	3.7	4.3	5.0	4.3	4.7	4.7	5.0	5.0	4.7	4.3	2.7	4.5	●	○	○	○	○
prop111 Laure, Erwin (KTH); Holanda Rusu, Victor (CSCS)		3	4.3	5.0	3.3	4.3	4.3	4.0	4.7	5.0	4.7	5.0	4.7	3.0	4.5	●	○	○	○	○
prop114 Bianco, Mauro (Swiss National Supercomputing Centre (CSCS...	1	4	4.5	4.8	4.8	4.5	4.0	4.8	4.0	4.5	4.5	4.8	4.0	3.0	4.5	●	○	○	○	○
prop107 Solcà, Raffaele (ETH Zurich); Di Napoli, Edoardo (Forschu...	3	3	4.7	5.0	3.7	4.3	4.7	4.7	4.7	2.7	4.3	5.0	3.7	3.0	4.3	●	○	○	○	○
prop110 Bull, Mark (University of Edinburgh); Beltran, Vicenç (Ba...		4	4.0	4.5	4.0	4.8	4.5	4.8	3.5	4.0	4.8	4.8	3.8	2.8	4.3	●	○	○	○	○
prop105 Gibbon, Paul (Forschungszentrum Juelich GmbH); Kos, Leon...	1	2	4.0	4.5	4.5	4.5	3.5	4.0	3.0	4.5	4.0	3.5	3.5	3.0	4.0	●	○	○	○	○
prop115 Merta, Michal (IT4Innovations); Zapletal, Jan (IT4Innovat...		2	4.5	4.5	1.0	4.5	5.0	4.0	4.5	4.5	4.5	1.5	4.5	3.0	3.9	○	●	○	○	○
prop113 Meyer, Norbert (Institute of Bioorganic Chemistry PAS); K...	1	2	4.0	4.5	4.0	4.5	3.5	3.0	4.0	3.0	4.5	3.0	1.5	2.0	3.6	○	●	○	○	○
prop119 Böhm, Stanislav (It4Innovations)		4	2.5	3.8	2.5	3.5	2.5	2.5	4.0	4.5	4.3	4.3	3.5	2.0	3.4	○	○	○	●	○
	C		SCP1	SCP2	SCP3	SCP4	APP1	APP2	APP3	APP4	TAM1	TAM2	TAM3	REC	WAvg	A	U	UE	R	AR
prop118 Hammer, Nicolay J. (Leibniz Supercomputing Centre (LRZ));...	1	2	2.0	3.5	3.5	5.0	2.5	2.0	4.0	4.0	3.5	4.0	2.5	2.0	3.3	○	○	○	●	○
prop112 Kristensen, Mads R. B. (University of Copenhagen)	3	2	3.0	3.0	3.5	3.5	2.5	2.5	4.0	4.0	4.0	4.0	2.0	2.0	3.3	○	○	○	●	○
prop106 Borrell Pol, Ricard (Barcelona Supercomputing Center); Ho...		3	2.7	3.3	3.0	3.7	3.0	2.7	3.3	3.3	3.0	3.7	2.7	1.7	3.1	○	○	○	●	○
prop116 Tekin, Adem (Istanbul Technical University, UHEM)		3	1.7	2.7	1.3	3.7	2.3	1.7	2.3	3.3	4.7	2.7	2.3	1.0	2.6	○	○	○	●	○
prop117 Aykanat, Cevdet (Bilkent University); Ozdal, Mustafa (Bil...		2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	○	○	○	●	○

Table 2: Summary of external review grading and SSC funding recommendations, review average on scope (SCP1-SCP4), approach (APP1-APP4), and team and management (TAM1-TAM3), together with a funding recommendation by the reviewers (REC) and average grade (WAvg) are shown in the columns, as well as the final status (Accept, Undecided, Reject). Not used C (Conflicted), UE (Under Evaluation), AR (Administrative Reject).