SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

INFRA-2011-2.3.5 – Second Implementation Phase of the European High Performance Computing (HPC) service PRACE

PRACE-2IP

PRACE Second Implementation Project

Grant Agreement Number: RI-283493

D9.1.1
Support for Industrial Applications Year 1
Final

Version: 1.0
Author(s): Cinzia Zannoni, CINECA

Date: 28.08.2012

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: RI-283493**	
	Project Title: PRACE Second Implementation Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: **PRACE-2IP**	
	Deliverable Nature: Report	
	Deliverable Level: PU	**Contractual Date of Delivery:** 31 / 08 / 2012
		Actual Date of Delivery: 31 / 08 / 2012
	EC Project Officer: Mr. Leonardo Flores Añover	

* - The dissemination level are indicated as follows: **PU** – Public, **PP** – Restricted to other participants (including the Commission Services), **RE** – Restricted to a group specified by the consortium (including the Commission Services). **CO** – Confidential, only for members of the consortium (including the Commission Services).

Document Control Sheet

Document	**Title: Support for Industrial Applications Year 1**	
	ID: **D9.1.1**	
	Version: 1.0	**Status:** ***Final***
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2007	
	File(s): D9.1.1.docx	
Authorship	**Written by:**	Cinzia Zannoni, CINECA
	Contributors:	Paul Graham (EPCC), Thomas Boenish (HLRS), John Donners (SARA), Tomas Kozubek (VSB), Peter Raback (CSC), Marcin Ostasz (BSC)
	Reviewed by:	F. Berberich, JUELICH – PMO Mark Parsons, EPCC
	Approved by:	MB/EB

Document Status Sheet

Version	**Date**	**Status**	**Comments**
0.1	11/01/2012	Draft	Description of the procedure for the selection of the application

0.2	6/08/2012	Draft	Insert document describing the survey methodology
0.3	7/8/2012	Draft	Inserted contributions on section 3 and executive summary revision
0.4	8/8/2012	Draft	Inserted contribution in section 1 Introduction. Added contribution HLRS on OpenFOAM
0.5	9/08/2012	Draft	Added section 3 Selection of relevant applications. Reported description of the application in this paragraph
0.6	10/08/2012	Draft	Added revised section on Elmer, accepted revision by Marcin. Uploaded on BSCW.
0.7	22/08/2012	Draft	Revised according to reviewers comments. Figures still to be improved.
0.8	23/08/2012	Draft	Accepted comments on figures. Addressed comments on Elmer and Delft3D. Uploaded on BSCW.
1.0	28/08/2012	Final	Version for EC submitted.

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, Industrial applications

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[1] *PRACE Industrial Offer Assessment*, PRACE 1IP D5.2.2 - http://www.prace-ri.eu/IMG/pdf/d5.2.2-publicversion_1ip.pdf

[2] http://www.01consulting.net/02_2011_MCAE_GLOBAL_Market_ES.html

[3] http://www.openfoam.org

[4] http://www.csc.fi/elmer

[5] http://www.sci.utah.edu/software/40-mrl/39-febio

[6] http://oss.deltares.nl/web/opendelft3d/home

[7] http://www.extend-project.de

[8] http://www.openfoam.org/docs/user/applications.php

[9] http://www.opensourcecfd.com/conference2012

[10] http://www.openfoamworkshop.org/

[11] http://www.deltares.nl/en/about-deltares

[12] http://www.deltaressystems.com/

[13] Elmer community portal, http://www.elmerfem.org

[14] Elmer source code, http://www.sf.net/projects/elmerfem

[15] http://mrl.sci.utah.edu/home

[16] PRACE-1IP WP7 Deliverable 7.5

[17] http://www.openfoam.org/docs/user/fvSolution.php#x21-1240004.5.3

[18] http://www.openfoam.org/docs/user/fvSolution.php#x21-1170004.5

[19] Culpo, Massimiliano,PRACE 1IP-WP7 White paper 'Current bottlenecks in the scalability of OpenFOAM on massively parallel clusters"

[20] Manguoglu, Murat, PRACE 1IP-WP7 White paper, "A General Sparse Sparse Linear System Solver and its Application in OpenFOAM", http://www.praceri.eu/IMG/pdf/A_General_Sparse_Sparse_Linear_System_Solver_and_Its_Application_in_OpenFOAM.pdf

[21] Rivera, O. et al., PRACE 1IP-WP7 White paper, Parallel Aspect of OpenFOAM in the solution of Turbulent Flows"

[22] J. Ruokolainen, P. Råback, M. Lyly, T. Kozubek, V. Vondrak, V. Karakasis, G. Goumas: Improving the scalability of Elmer finite element software, http://www.prace-ri.eu/IMG/pdf/elmer.pdf

[23] C. Farhat, F-X. Roux, An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems, SIAM J. Sci. Stat. Comput. 13, 1992; 379–396.

[24] Z. Dostal, D. Horak, and R. Kucera: Total FETI – an easier implementable variant of the FETI method for numerical solution of elliptic PDE, Communication in Numerical Methods in Engineering 22, 2006, 1155-1162.

[25] T. Kozubek, D. Horak, V. Hapla: FETI Coarse Problem Parallelization Strategies and Their Comparison, PRACE 2IP WP12.2 whitepaper, http://www.prace-ri.eu/IMG/pdf.

[26] "MUMPS Web page", http://graal.ens-lyon.fr/MUMPS/

List of Acronyms and Abbreviations

BSC	Barcelona Supercomputing Center (Spain)
CAD	Conputer Aided Design
CAE	Conputer Aided Engineering
CFD	Computational Fluid Dynamics
CINECA	Consorzio Interuniversitario, the largest Italian computing centre (Italy)
CPU	Central Processing Unit
CSC	Finnish IT Centre for Science (Finland)
CSR	Compressed Sparse Row (for a sparse matrix)
EPCC	Edinburg Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
FETI	Finite Element Tearing and Interconnecting
GENCI	Grand Equipement National de Calcul Intensif (France)
GNU	GNU's not Unix, a free OS
GPL	GNU Public License
GPU	Graphic Processing Unit
HLRS	High Performance Computing Center Stuttgart
HPC	High Performance Computing; Computing at a high performance level
ISV	Independent Software Vendor
LGPL	Lesser general GNU Public License
MPI	Message Passing Interface
Open MP	Open Multi-Processing
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PRACE 1IP	PRACE First Implementation Phase
PRACE 2IP	PRACE Second Implementation Phase
SARA	Stichting Academisch Rekencentrum Amsterdam (Netherlands)
SME	Small Medion Enterprise
SW	Software
Tier-0	Denotes the apex of a conceptual pyramid of HPC systems. In this context the Supercomputing Research Infrastructure would host the Tier-0 systems;
Tier-1	Denotes the second layer of a conceptual pyramid of HPC systems. It refers to national or topical HPC centres would constitute
VSB	Technical University of Ostrava
WP5	Work Package 5 in PRACE 1IP: 'Relation with Industrial users'
WP9	Work Package 9 in PRACE 2IP: 'Industrial Application Support'
WP12	Work Package 12 in PRACE 1IP:'Novel programming technique'

Executive Summary

The PRACE 2IP-WP9 task is concerned with providing support for industrial partners' applications in order to better exploit HPC resources. This report describes the activity within task 9.1 on identifying Open Source codes which are both of relevance to industry and amenable to making efficient use of Tier-0 and Tier-1 systems, and the efforts underway in enhancing the performance of these codes.

The PRACE access program allows industrial partners to use Tier-0 systems. There are three principal software groups which can potentially take advantage of this access: proprietary codes owned and used by individual companies; independent software vendor (ISV) codes; and Open Source codes. Both large companies and SMEs are expressing a growing interest in Open Source simulation codes for their research and production activity. Many of these, however, can suffer from relatively low scalability.

The first aim of this work is to identify some of the relevant Open Source codes that are of interest to industry and could benefit from improved HPC exploitation. This was achieved by undertaking a survey of industrial partners. The survey results are presented and analysed in this report. A set of criteria were identified to choose which Open Source codes to investigate further. Based on this and the survey results, the process led to three codes being chosen as being of high interest to several partners: OpenFOAM, a CFD software package; Elmer, a multi-physics finite element code; and Delft3D, a modelling suite for investigating hydrodynamics, sediment transport and water quality for fluvial, estuarine and coastal environments.

The second aim of this task is to enable these codes to better exploit the PRACE Tier-0 and Tier-1 resources. The report describes the work undertaken on each code, including the identification of performance bottlenecks and the strategy for improving performance and scalability. The development work on the codes is an ongoing process not due to be completed for several months, so the report presents progress so far and preliminary results where available.

1 Introduction

The PRACE 2IP-WP9 task is concerned with providing support for industrial partners' applications in order to better exploit HPC resources. This report describes the activity within this task on identifying Open Source codes which are both of relevance to industry and (potentially) amenable to making efficient use of Tier-0 and Tier-1 systems, and the efforts underway in enhancing the parallel performance of these codes.

The objectives of the task are to:

- Survey industrial HPC users to examine their HPC usage
- Identify Open Source codes widely used by industry
- Enhance the parallel performance of a selection of these codes

With regards to the first objective, a detailed survey was conducted of industrial users of HPC facilities. From this survey several Open Source codes used by industry were identified, as discussed in Section 2 and 3 below. Also in this section the appropriateness of these codes for further investigation by PRACE is discussed.

Section 4 discusses in detail the enabling projects being undertaken with the relevant Open Source codes, and includes some preliminary results of the process.

Finally Section 5 summarises the progress so far and future work.

2 Identification of relevant Open Source codes for industry

The input data used to select open source applications interesting for industry has been obtained through two surveys addressing 1) Industrial End Users (those industries that use parallel applications and HPC technologies in their R&D or production activities); and 2) Independent Software Vendors (ISVs). The surveys have been carried out in collaboration with PRACE-1IP WP5's team by adding specific questions about the applications used by industry to a more general survey targeting industrial user requirements [1]. With this approach, the risk of contacting the same industrial contact twice (once for WP5 and another time for WP9) to fill in similar surveys was avoided.

2.1 Survey methodology

The general survey targeting Industrial End Users, issued together with PRACE-1IP WP5, is composed of 24 questions profiling both the company responding and the requirements of Industrial End Users in terms of HPC resources and expertise. The general ISVs survey, issued together with PRACE-1IP WP5, is composed also of 24 questions profiling the HPC requirements of application developers. The following sections report the questions introduced in the surveys to collect information on the open source applications of interest to industry and discuss the results obtained. The surveys opened on November 16th, 2011. PRACE-1IP WP5 and PRACE-2IP WP9 teams advertised the survey among their industrial contacts and the participants of the PRACE Industrial Seminar. On January 30th, 2012, the surveys were closed with 36 participants in total and data analysis was initiated.

2.1.1 *Industrial End User Survey*

Five out of the 24 questions of the general survey, issued to identify the size of the company, the position of the participant in the organization, the industry sector, were relevant also for the present work. Four questions of the general survey, listed below, were specifically put in place to identify the applications currently used or that the organisation plans to use in the next 24 months or simply applications they considered interesting.

- Are you using Computer Aided Methodologies (CAE, CAD, or Data Analysis)?
  - Yes
  - No
- What are the applications that you are currently using?
  - Commercial ISVs' applications – Please list
  - Open Source applications – Please list
  - Internal applications – Please list
- What other applications are you planning to use in the next 24 months?
  - Commercial ISVs' applications – Please list
  - Open Source applications – Please list
- What other application(s) that you are not using but you consider interesting to use for your simulations?
  - Commercial ISVs' applications – Please list
  - Open Source applications – Please list

2.1.2 *ISVs Survey*

Five out of the 24 questions of the general survey, issued to identify the size of the company, the position of the participant in the organization, the industry sector, were relevant also for the present work. Moreover, seven questions of the general survey, listed below, were issued specifically to address ISVs licensing model and parallel performance of their applications:

- What are the principal codes your organization produces?
  - Please list
- Are any of these codes enabled to take advantage of multicore processing?
  - Yes
  - No
- For the codes that have been parallelised, what technology has been used?
  - MPI
  - openMP
  - Pthreads
  - CUDA
  - openCL
- What is the typical core count that your codes efficiently run on?
- Which is the current licensing model you use?
- Does your pricing model increase linearly with the number of processors/cores?
- If your pricing model does not increase linearly with the number of processors/cores, please describe your pricing mechanism

2.1.3 *Survey participants*

General questions on the company size and usage of HPC available in the survey allowed us to profile the companies that answered the survey and better evaluate the survey results. This section provides an overview of this data.

The two surveys registered 36 participants in total:

- Industrial End Users: 28 participants
- ISVs: 8 participants

24 participants left an indication of the country with the following distribution:

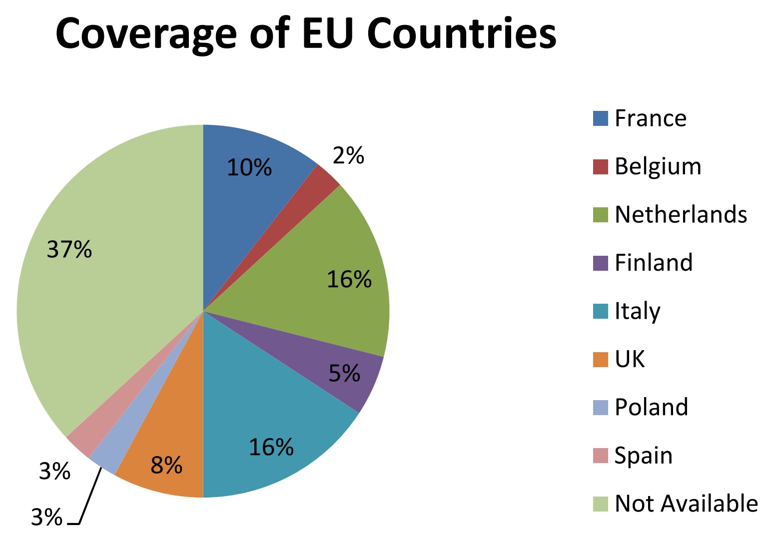

Figure 1: EU countries of the participants – both surveys considered

The companies participating in the ISV survey are significantly smaller than the End Users. Among others, one of the reasons is that the HPC centres that work with industry, in general have closer collaborations with companies that are HPC end users than ISVs. Relations with ISVs are more complicated due to the fact that most of the development teams are not located in Europe and the sales teams are more interested in directly working with End Users and generally see the HPC centres as a potential customer rather than a partner. However, when the company profile indicators are the same in both surveys, the results related to the company profile are aggregated. The number of the companies participating in the ISV Survey did not alter the trends of the results.

The information coming from the two surveys are relevant for both SMEs and large to very large companies. This holds true if we consider the 36 participants as a whole, as in the following pie charts, or the participants of the End User survey and of the ISV survey separately. If we observe both the revenue in the year 2010 and the number of employees presented in the following pie charts, we can see that about 50% of the respondent companies can be classified as small SMEs with less than 100 employees and less than 10 million Euros revenue, about 15% of them are large companies with more than 1000 employees and about 26% are very large companies with revenues higher than 1 billion Euros and more than 10.000 employees. Moreover most of the participants belong to both technical and top management providing significant answers concerning the importance of HPC and of the application used.

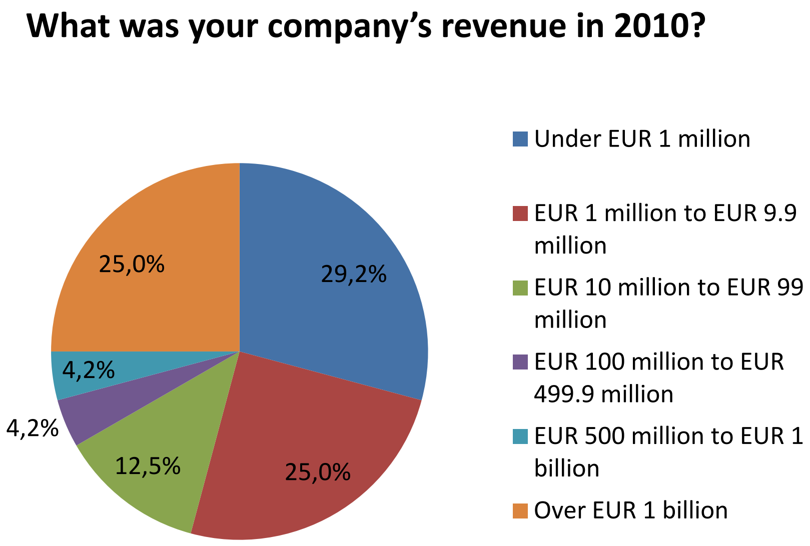

Figure 2: Revenue year 2010 - 24 answers – both surveys considered

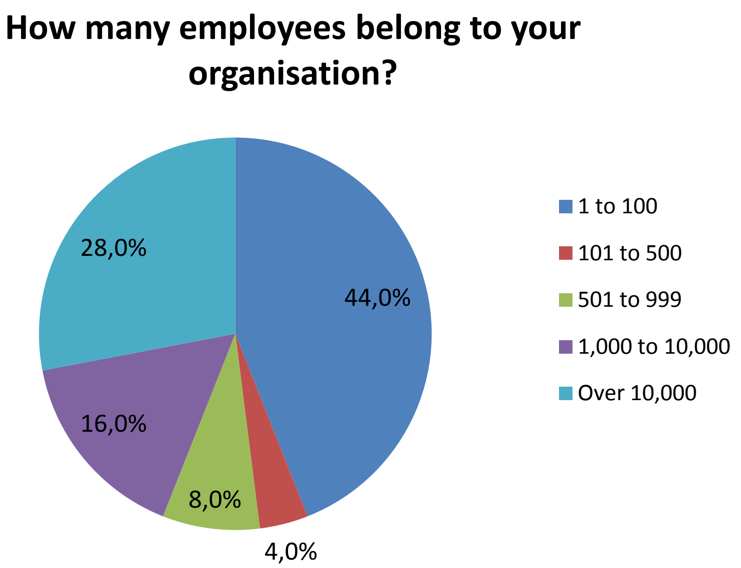

Figure 3: Company Employees - 25 answers – both surveys considered

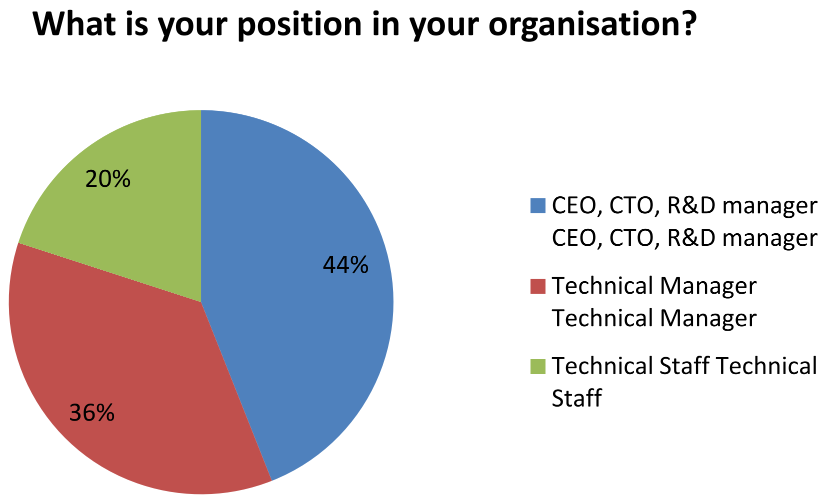

Figure 4: Position in the organization - 25 answers – both surveys considered

The companies that answered the End User survey cover a significant range of industries where HPC is known to be a key technology both in R&D and production activity. All of the respondent companies but one currently use HPC resources and state that they will continue to use it in the future. More than half of them state they are interested in extending HPC usage in the future.

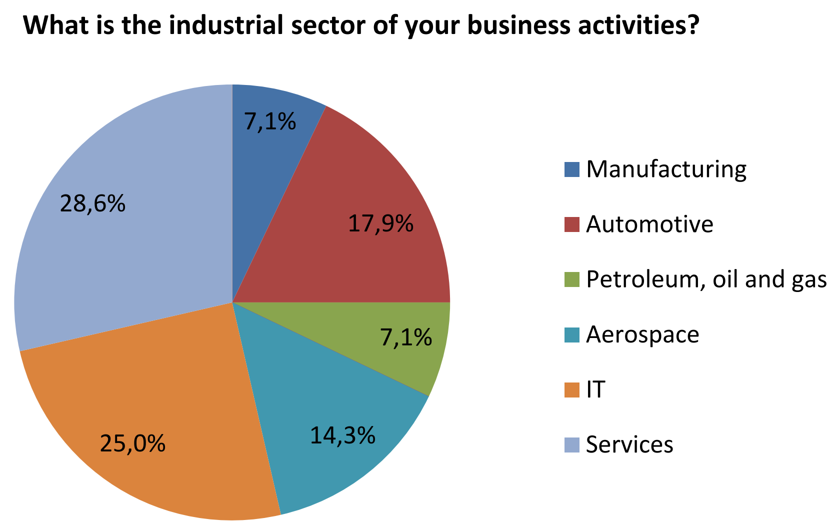

Figure 5: Business areas - End Users Survey

The companies answering the ISV survey relate to the following application areas:

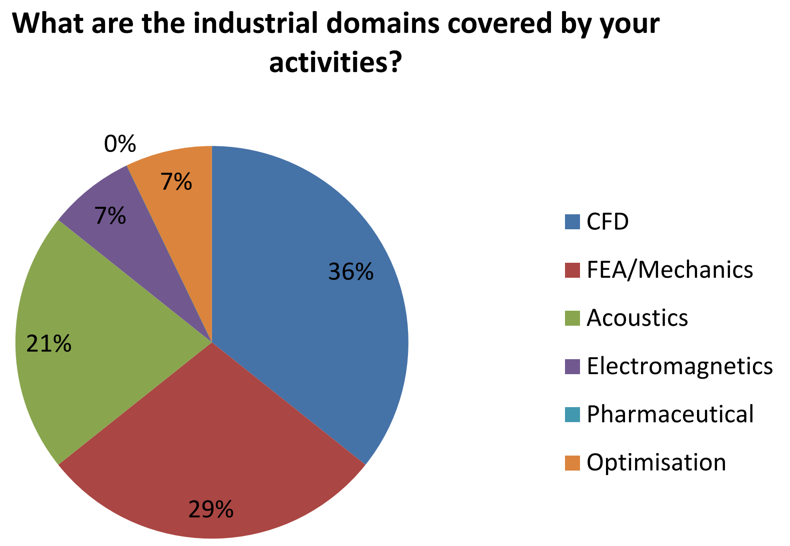

Figure 6: ISVs application domains - ISV survey

Concerning the typical usage of HPC, as shown in the following pie charts nearly 70% of them typically use a small number of cores, lower than 512, and only 10% use more than 2,048. Only one specified over 30,000 cores. About 18% of them have a capacity profile with a lot of small runs on a small number of cores, but over 80% tend to use a high number of cores. It has to be considered, however, that some of them perceive 100 cores as a high number of cores. Considering system architecture, all of them but one indicated an x86 based cluster system. The single exception to this indicated a BlueGene.

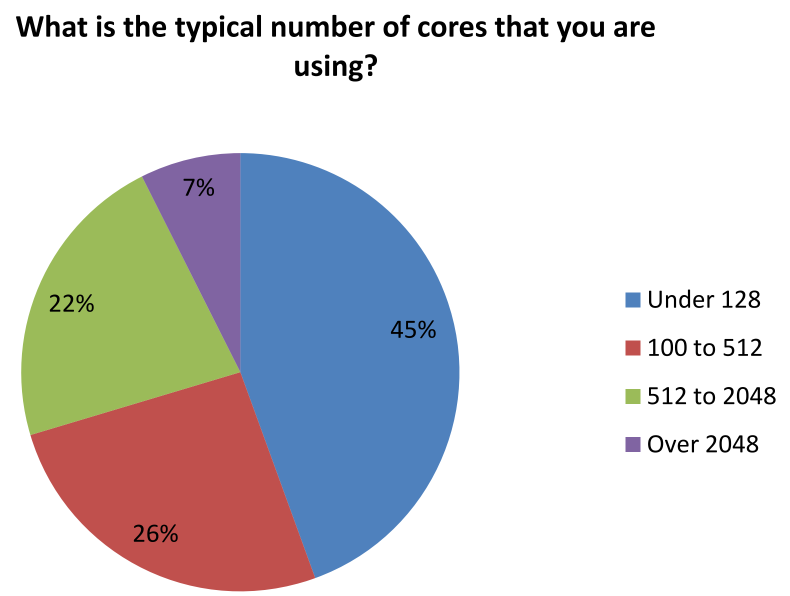

Figure 7: Typical number of cores used - both surveys considered

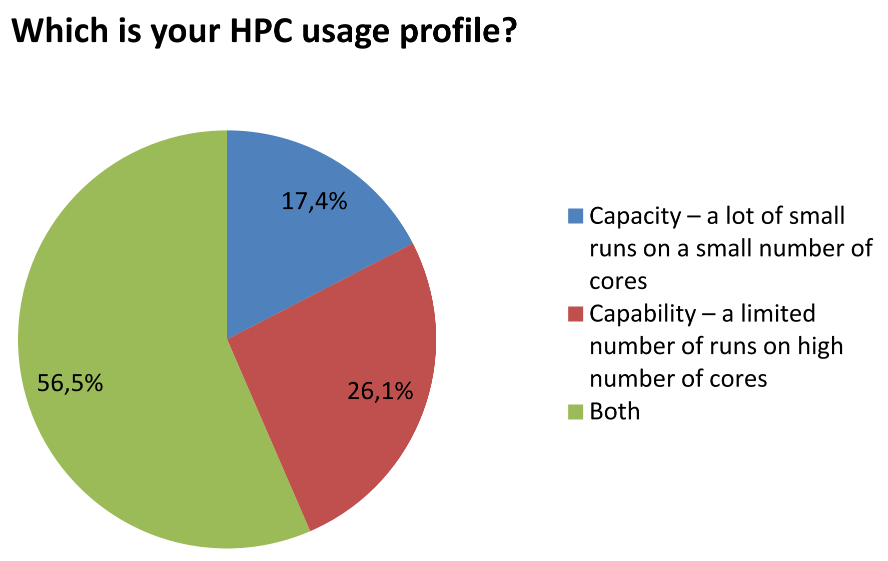

Figure 8: HPC profile - End User Survey

2.2 Surveys results

This section presents the survey results targeting specifically the applications used by industrial HPC users and developed by those ISVs that see the relation with HPC centres as promising. General indications on the parallel implementation of the applications and on the licensing models are provided even if the number of the participants to the ISV survey is very limited.

2.2.1 *Applications used by HPC Industrial Users*

It emerges from the survey that Industrial Users currently use either commercial ISVs' applications, open source codes and proprietary internally developed applications in the ratios shown in Figure 9.

End Users have been asked to list the most relevant applications and among the Commercial ISVs' codes, Ansys, and specifically its CFD code Fluent, is by far the most used software. More than 40% of the indications made by the participants were for Ansys and Fluent. This result is in line with our expectations. Although the CAE market is highly fragmented, Ansys is the market leader with between 20%-25% of the market (as stated by recent market research undertaken by 01Consulting [2] whose results are reported in Figure 11).

With regard to Open Source applications currently used by industrial users, OpenFOAM [3] is the most referenced one. Nine participants listed it among the open source applications currently used while two listed Elmer [4] and one listed FEBio [5].

It is worth noting that when asked about the applications they are not currently using but planning to use in the next 24 months or those that they consider interesting, although many of them skipped the question, those who answered reported OpenFOAM as the most interesting (4 users), followed by Ansys (2 users), Elmer and LS-Dyne (1 user).

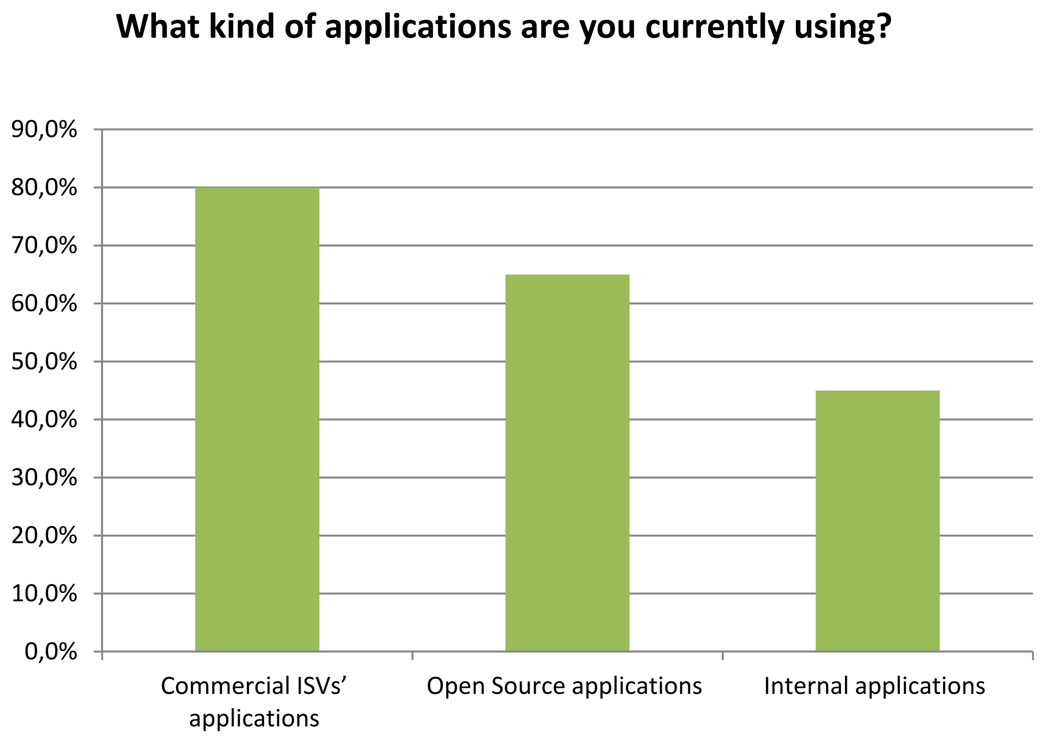

Figure 9: Different kinds of applications used by industrial users – Multiple choice allowed

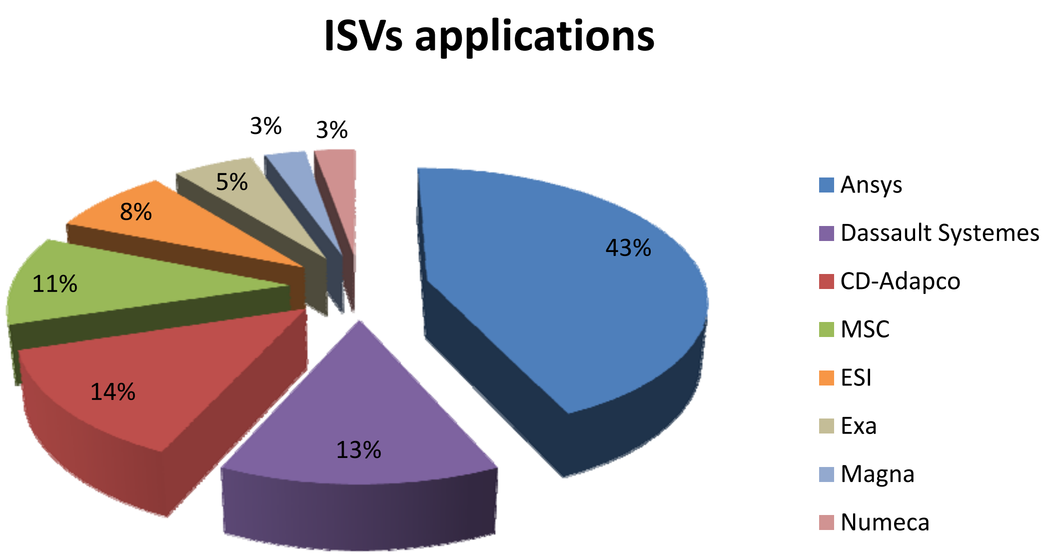

Figure 10: ISVs listed by Industrial End Users

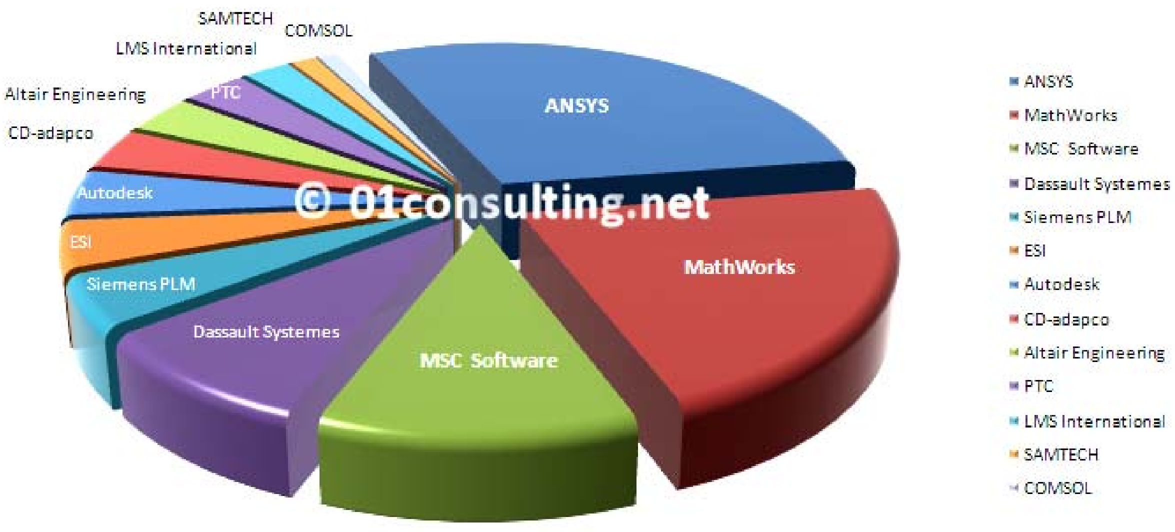

Figure 11: CAE market shares 2011
(Image publicly available on the web, see [2])

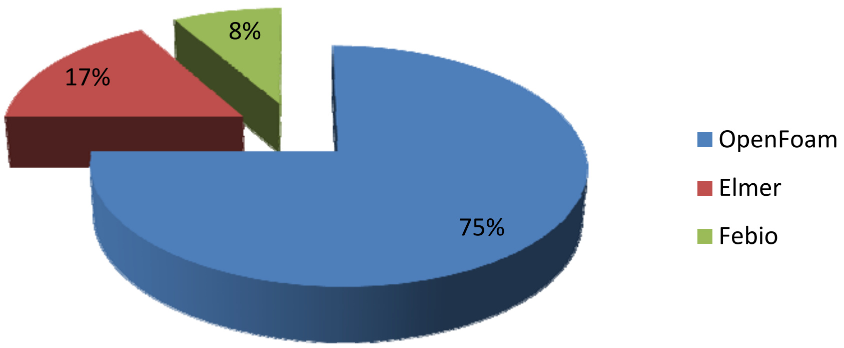

Figure 12: Open Source applications

2.2.2 *ISV codes*

The ISVs or SW companies that filled in the survey are involved in the development of the following codes:

- Open Source:
  - Elmer
  - OpenFOAM
  - Delft3D & SIMONA[6]
- Commercial ISVs codes:
  - Ansys CFX and Fluent

Although all the codes have a parallel implementation, as expected, their scalability is generally rather poor. Detailed analysis on the different codes needs to be done. From the survey appears that the major obstacle for ISVs to scale up the codes are the development costs.

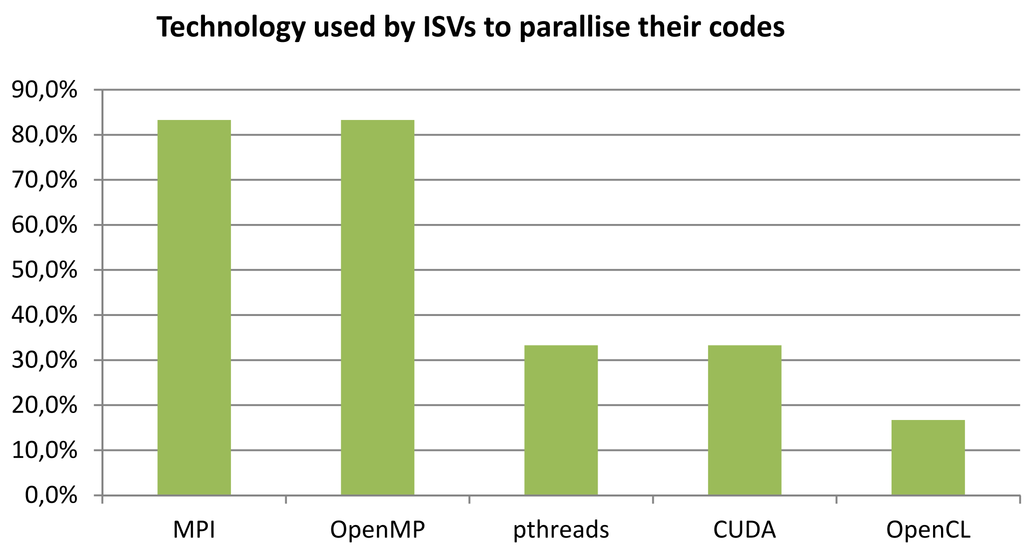

Figure 13: Different technologies used by ISVs to parallelise their codes

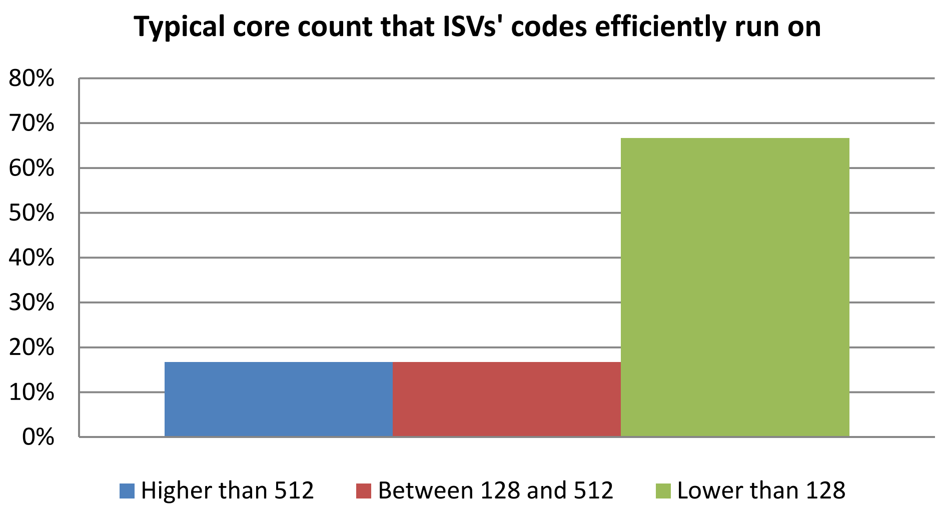

Figure 14: Typical core count that ISVs codes efficiently run on

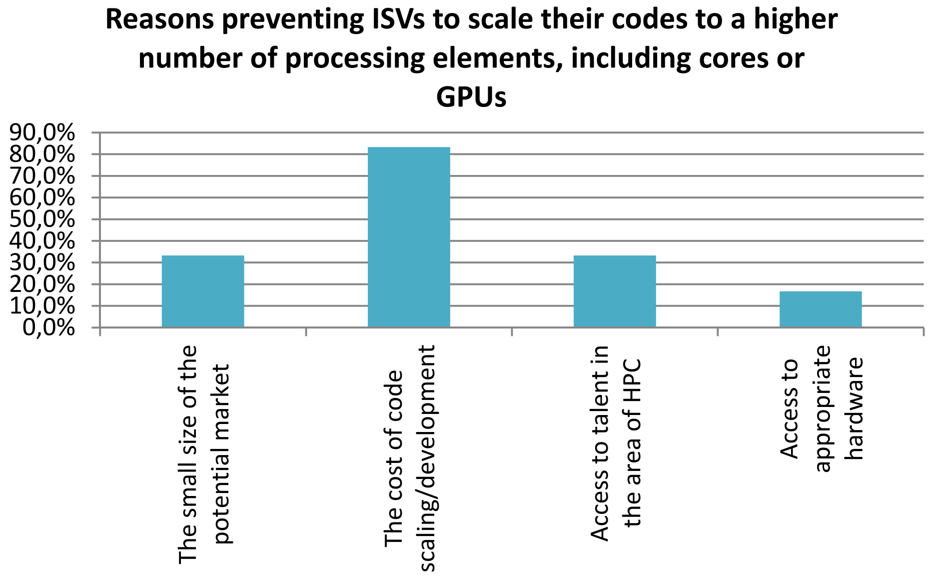

Figure 15: Main reasons preventing ISVs to scale up their codes

3 Selection of the applications relevant for enabling activity

The survey has been used as the primary instrument to identify applications of interest to industry. The applications listed were further analysed and beside the indication from the survey the actual usage by industry was assessed (analysing by a deep activity the information publicly available on the codes) and the necessary requirements to start an enabling project verified. The following section provide a general overview of the application and for each one the following parameters are reported:

- Actual usage by industry
- License terms and possibility to contribute code to the development project
- Available contacts between PRACE team and the developers community

3.1 OpenFOAM

3.1.1 *General description*

OpenFOAM is an open source CFD software package used throughout the world in academic as well as in industrial environments. There are currently two development directions: The OpenFOAM Foundation [3] with the latest version being 2.1.1 and the OpenFOAM Extend Project [7] with version 1.6-ext. OpenFOAM is a registered trademark of SGI Corp.

OpenFOAM is an application that can also be used as a C++ library. Besides utilities for pre-, post-processing and mesh generation, it provides a large number of solvers for incompressible, compressible and multiphase flows. Turbulence models, chemical reactions (especially combustion), heat transfer and electromagnetics can be included [8].

3.1.2 *Assesment of the requirement*

The results of the survey are confirmed. In addition to large industries such as Audi or Bombardier who use OpenFOAM in production modelling and simulation, engineering consulting SMEs such as Icon or Wikki use OpenFOAM for their activities, provide training courses and support contracts. The *Open Source CFD International Conference* [9] at its 6th

edition, and the *OpenFOAM workshop* [10] at its 7th edition, present many examples from industry.

Contacts between the PRACE team and the developers' community, which is based in the UK with contributors from many EU countries, are in place. CINECA and HLRS have established contacts with OpenFOAM developers and are currently cooperating with them.

OpenFOAM is managed and distributed by the OpenFOAM Foundation and is freely available as open source, licensed under the GNU General Public Licence.

This prevents open source software being exploited by its inclusion within non-free, closed sourced software products. Apart from this, the licence is designed to offer freedom, in particular it does not force users of the software to make modifications or developments publicly available. OpenFOAM can be used as the basis of in-house, proprietary software provided the resulting software is not sold or made available to any external parties.

Actual usage by industry has been confirmed and the necessary requirements to start a profitable enabling project have been verified by the research presented here.

3.2 Delft3D

3.2.1 *General description*

Delft3D [6] is a world leading 3D modelling suite to investigate hydrodynamics, sediment transport and morphology and water quality for fluvial, estuarine and coastal environments. As per January 1st, 2011, the Delft3D flow (FLOW), morphology (MOR) and waves (WAVE) modules are available in open source. Delft3D has over 350,000 lines of code and is developed by Deltares [11] an independent, institute for applied research in the field of water, subsurface and infrastructure.

3.2.2 *Assessment of the requirements*

The results of the survey are confirmed. The software is used and has proven its capabilities in many places around the world, such as the Netherlands, USA, Hong Kong, Singapore, Australia, Venice, etc. The software is continuously improved and developed with innovative advanced modelling techniques thanks to the research work of Deltares. Training courses and SW packages are distributed under the Deltares Systems [12] brand and services.

Deltares Systems distributes its products both using a closed source model and open source model. Delft3D is distributed under the GNU Public License (GPLv3). In the same was as OpenFOAM this prevents open source software being exploited by their inclusion within non-free, closed sourced software products. The license allows modification of the source code to develop proprietary applications.

Contact between Deltares, which is located in the Netherlands, and PRACE partners has been established. Deltares expressed interest in actively supporting the enabling project and confirmed that participation with a speech to the 4th PRACE Industrial Seminar held in Bologna, 16th-17th April 2012.

Actual usage by industry has been confirmed and the necessary requirements to start a profitable enabling project have been verified by the research presented here.

3.3 Elmer

3.3.1 *General description*

Elmer is, to the authors' knowledge, the most popular multiphysical simulation software published under open source [4][13][14]. It consists of several components: ElmerGUI is the graphical user interface, ElmerPost is the post-processing tool, ElmerGrid is the mesh manipulation and partitioning tool, and most importantly, ElmerSolver is the finite element solver.

Elmer includes physical models of fluid dynamics, structural mechanics, electromagnetics, heat transfer and acoustics etc. The physical models may be weakly coupled without any *a priori* defined method. Due to the modular structure new equations may be added to Elmer without touching the main library. The equations are implemented in a way that hides the space dimension, element order and also the parallel solution process. All-in-all there are around 50 different physical or auxiliary modules in Elmer.

3.3.2 *Assessment of the requirements*

The internationally most visible user community of Elmer is in computational glaciology. In industry it has been used to model crystal growth (e.g. Okmetic, DC Wafers), acoustics (e.g. Nokia), and MEMS devices (e.g. Innoluce, VTI Technologies), for example. In the period from January - June 2012 the Windows binary from sourceforge.net [14] was downloaded over 9000 times. Naturally most of these users use Elmer in serial mode only, but with improved parallel performance there is a natural route for massively parallel computing.

The relation with the Elmer developers and PRACE is very strong since CSC – IT Center for Science (CSC) is one of the initial developers. Elmer development was started in 1995 in collaboration with several Finnish Universities, research institutes and industry. After its open source publication in 2005, the use and development of Elmer has become very international.

In May 2012 the library functionalities of Elmer were published under Lesser GNU Public License (LGPL) which makes the use of Elmer code in industry even more flexible than before.

Actual usage by industry has been confirmed and the necessary requirements to start a profitable enabling project have been verified by the research presented here.

3.4 FEBio

3.4.1 *General description*

FEBio is a nonlinear finite element solver that is specifically designed for biomechanical applications. It offers modelling scenarios, constitutive models and boundary conditions that are relevant to many research areas in biomechanics. All of the features can be used together seamlessly, giving the user a powerful tool for solving 3D problems in computational biomechanics. The software is open-source, and pre-compiled executables for Windows, OS-X and Linux platforms are available. It is an academic code and its development was initiated and is currently supported by the Staff of the Musculoskeletal Research Laboratories [15] of the University of Utah (US).

3.4.2 *Assessment of the requirements*

The FEBio software is an academic project focused on modelling and simulation of biomechanics problems and specifically on modelling materials representing soft tissue behaviour in FE models. To the authors' knowledge, and according to the FEBio web site

[15], including articles and publications, there is no evidence of any collaboration with industry nor any industrial application.

The developer team located in the US allows collaboration among developers through forums. However, there is no clear explanation on how or even if the development team accept contributions from external developers. There isn't any software repository available and the source code can only be downloaded through zip files from the web site.

The source code is distributed under a three years License agreement requesting the recipient not to use the software for any commercial purpose, and limit use of the software for the purpose of research only. In contrast to the LGPL license, the agreements ask the recipient not to modify the software or create derivative works based upon the software.

The licensing mechanism is more strict than the other codes making the usage from industry less profitable. We also found no compelling evidence that this code is widely used by industry. Given these factors, it was felt that FEBio was not an appropriate code to examine further within this task.

4 Enabling projects

This section considers the work being undertaken to enable the chosen Open Source codes for Tier-0 and Tier-1 systems.

The work that will be done on OpenFOAM and Elmer is based on and continues the work that was done in PRACE-1IP WP7 [16]. Concerning OpenFOAM the focus will be on further improving scalability and testing on significant industrial cases. The strategy to remove the scalability bottleneck tested in PRACE-1IP WP7 [16] will be further developed, fine tuned and tested on industrial cases. With regard to Elmer, the aim is to extend the implementation of multithreading to improve scalability, implement features important for industrial applications like rotating boundary conditions and test the achievements obtained in the implementation of different parallelization strategies on test cases relevant to industry.

The next sections describe the workplan for PRACE-2IP WP9 and preliminary results in more detail.

4.1 OpenFAOM for Industrial Applications

According to the analysis of the survey results performed in section 2.2.1 OpenFOAM is of high interest in industry. In nearly every response, it has been mentioned as being used in the company or at least as being of high interest. Compared to the academic side, where mainly the versions from the extended project are used, industry very often relies on the versions provided by the OpenFOAM Foundation. In addition, industry typically relies on proven methods for simulations whereas academics are often interested in the development of new methods and a corresponding extension of OpenFOAM. As a result, the solvers and modules used within OpenFOAM by industry are often different from those used by the academic side.

Therefore, it has been decided in this effort to work explicitly on an OpenFOAM version and especially on the specific solvers and modules of high interest for industry.

4.1.1 *Workplan*

The goal of the work on OpenFOAM in this task is the enhancement of the software for industrial usage on PRACE Tier-0 and Tier-1 systems. The focus is on OpenFOAM versions, solvers and modules used by industry. Current industrial cases are in the range of 20M to 50M mesh cells with partially complex (multiphase) flows. These cases scale to 10s to 100s of cores on today's PC-cluster like systems. On the other hand modern Tier-0 systems provide in excess of 100,000 cores which can all be used in parallel. One user for example has mentioned a case with 20M cells that does not scale beyond 32 cores but with a run time of several days. Nevertheless, this user would be interested in using meshes with up to 1 billion cells for the same setup if possible. Memory limitations require greater numbers of cores are deployed but the code will almost certainly not scale, and hence perform, well. Therefore, the main goal is to improve scalability of OpenFOAM for industrial relevant cases.

Within PRACE-1IP WP7 some of the OpenFOAM solvers have been thoroughly benchmarked and profiled on Tier-0 and Tier-1 Intel hybrid clusters (TGCC Curie and CINECA PLX). As is typical of CFD applications the scalability bottleneck has been identified as being in the MPI communication pattern of the linear algebra core libraries. To boost parallelism further the addition of multi-threading capabilities to the code has been proposed. A first prototype implementation, managing fine-grained parallelism at the socket or node level, has been provided and has shown promising results [19]

In Task 9.1 of PRACE 2IP WP9 the plan is to extend this work, porting OpenFOAM to a BG/Q architecture to assess the effectiveness of a massively parallel architecture on industrial test cases. To this aim the multi-threaded implementation started in PRACE-1IP will be further developed and fine-tuned. Moreover a careful study of OpenFOAM I/O performances will be conducted on the same machine, and appropriate strategies to overcome architectural-dependent bottlenecks will be devised and implemented.

Contacts to relevant industrial users will be established to acquire significant (test) cases from industry. In this way the analysis of OpenFOAM on HPC-Systems done in PRACE 1IP WP7[16] [19] will be further extended on all the relevant solvers using significant industrial cases. The attempt to improve scalability of the relevant solvers and modules by taking into account the whole simulation cycle will be done.

Finally the industrial cases will be used to validate the changes and as a benchmark to assess the benefits and limits of running realistic CFD simulations on a Tier-0 supercomputer.

4.1.2 *Preliminary Results*

Contact with industrial users has been established. First discussions and phone conferences have been performed. AirLiquide is about to provide two of their production-like test cases which will be used for the planned OpenFOAM enhancement. The relevant software versions have been installed on the PRACE Tier-0 system Hermit.

With regard to the scalability issue mentioned earlier, the outcome of the analysis performed by PRACE-1IP WP7 has been investigated for relevance to the industrial case. In general, the convergence of CFD simulations depends on the underlying velocity-pressure coupling algorithm. There are two standard algorithms: the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm and the PISO (Pressure-Implicit Split-Operator) algorithm. Most fluid dynamics solvers in OpenFOAM use one of these approaches. PISO is used for transient problems and SIMPLE for steady-state problems [17][18] The application solvers themselves call the linear solvers: PCG or GAMG are used for solving the pressure equation and PBiCG or some "smooth solver" solve the velocity equation. OpenFOAM also provides some pre-conditioners (for example DIC, DILU, GAMG) and smoothers (Gauss-Seidel, DIC) [18]. These linear solvers have been identified as bottlenecks in PRACE-1IP [19][20][20].

In addition, contact with the OpenFOAM community has been established and PRACE staff are actively participating in the special interest group (SIG) "HPC, Scalability and large test cases" (http://www.extend-project.de/user-groups/groups/viewgroup/75) which was founded at the 7th OpenFOAM Workshop in Darmstadt (in June 2012).

The next steps will be the detailed bottleneck analysis on industrial cases together with the instrumentation of specific parts of OpenFOAM. In addition, linear solver scalability will be further investigated.

4.2 Delft3D

The FLOW module is the heart of Delft3D and this is a multi-dimensional (2D or 3D) hydrodynamic (and transport) simulation programme which calculates non-steady flow and transport phenomena resulting from tidal and meteorological forcing on a curvilinear, boundary fitted grid or spherical coordinates. In 3D simulations, the vertical grid is defined following the so-called sigma coordinate approach or Z-layer approach. The MOR module computes sediment transport (both suspended and bed total load) and morphological changes for an arbitrary number of cohesive and non-cohesive fractions.

The application consists of mainly Fortran 90, with some routines in C and C++ and some features from Fortran 2003. It uses MPI with domain decomposition as its parallelisation strategy. Although the application should scale well, as is shown by similar models with the same input data set, this is not the case. The developers did not provide us information on code profiling so a profile analysis must be done to show the main bottleneck that prevents the scalability of the code.

4.2.1 *Workplan*

The first steps will be to port the Delft3D model to different compilers and MPI libraries. This should give an indication of portability issues and how to remove them. Some points are:

- Configuration
- Makefiles
- Non-portable language elements

The next step is to create a benchmark and discuss that with the developers. Most important aspects are:

- Scalasca benchmark for parallel profile, communication time and hardware counters.
- Gperftools for benchmark at source-line level.
- What are the most important routines in terms of cpu time?
- How does the computation compare to the peak performance?
- What routines are the least scalable?
- Is scalability limited by communication time or something else?

Based on this initial work an informed, prioritised plan of improvements will be created.

4.2.2 *Configuration and setup*

The program uses automake, autoconf and libtool to create a configure script, which then configures the whole package for a particular system. An svn check out is the only method to obtain the source code. It is expected that a packaged version of the source of Delft3D would come with only a configure script to target a specific platform. At the moment, there are still a few platform-specific options in the configure.ac-file that prevent this.

4.2.3 *Porting*

The library stdc++ was thought to be necessary for each C++-compiler, but this is not true. E.g., the IBM XL C++ compiler uses a different set of libraries, which are included automatically when dynamically linking your application or library with the C++ compiler. This has therefore been removed from configure.ac. As a result, the compilation fails because some C++-parts of the Delft3D-code are compiled as static libraries (libstream.a and libesm_c.a) and linked with Fortran-parts, which then miss the C++-libraries. The automake-files (*/Makefile.am) are changed to build shared libraries of the C++-parts through libtool, which then automatically links to the C++-libraries, even if it is combined with Fortran-code.

A special PRACE svn branch has been setup at Deltares. The MPI-implementation in Delft3D checks the environment variable PMI_RANK, which is only used in the MPICH2-library and is therefore not yet portable either. Delft3D only checks if its length is at least one character, so this can easily be fixed.

4.2.4 *Benchmarking*

The simulation is a schematic representation of the Waal, one of the main rivers in the Netherlands, with groynes and part of the floodplain. This model is used to estimate the effect of lowering the groynes on flood level.

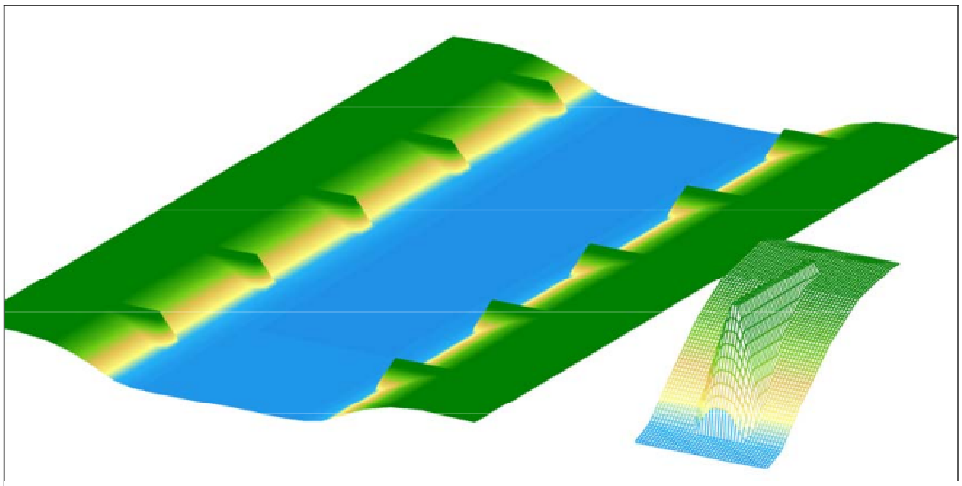

Figure 16: Image of the floodplain

The resolution is high enough to reach a good scaling up to 80 processors using a similar, but commercial, software package that is also developed by Deltares. The total domain is 30x2 km and uses a resolution of 2x2m in the main channel and 4x2m on the floodplain. The total number of grid cells is more than 9 million.

After the first, non-performance-related, bugs were resolved an initial scaling benchmark could be produced.

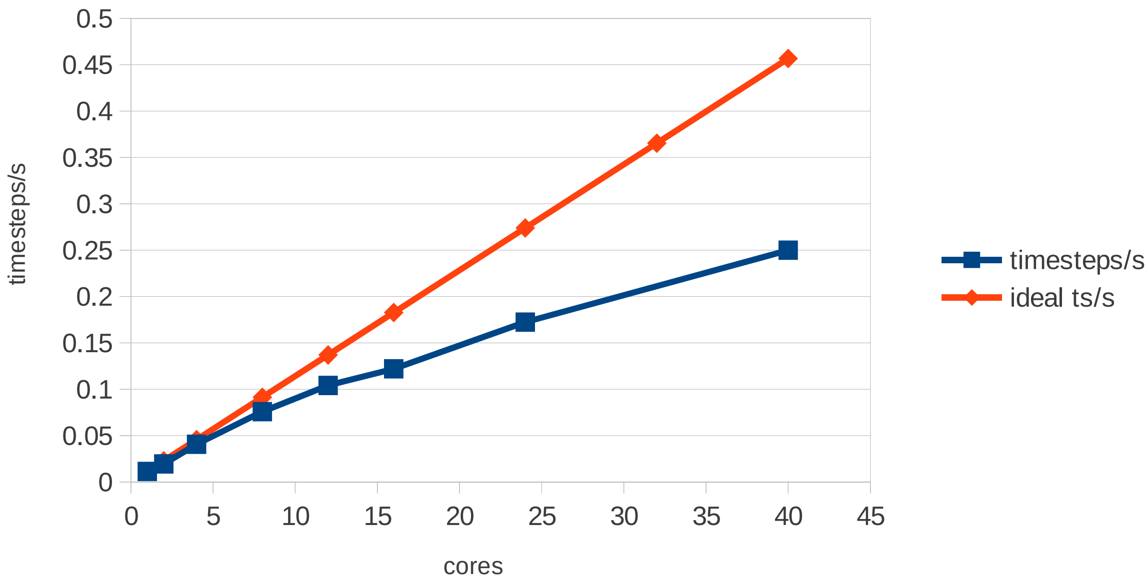

Figure 17: Initial scaling of the simulation

4.2.5 *Performance analysis*

The scalasca tool was used to generate a performance profile of the application. Version 1.4.1 was specially compiled with position-independent code, so the scalasca libraries can be linked with the shared libraries in Delft3D.

The platform used was the lisa cluster at SARA, with:

- Intel Nehalem processors
- 32 GB of memory per node
- OpenMPI 1.4.5
- Intel compilers 11.1.072

The Delft3D code uses, by default, dynamic loading of shared libraries, which is not supported by the scalasca tool. Therefore, we used the option to compile Delft3D as a monolithic executable, with the shared libraries all added at link-time.

The MPI-implementation would not call MPI_Finalize in the case where there was only one MPI-task, so scalasca would not write its final report. This is now fixed in the svn branch.

Furthermore, for three MPI tasks or more, there would be an integer overflow when multiplying the total nr. of grid points (9M) with the running sum of the cpu weights (in this case 300). This was fixed by using INTEGER*8 variables.

The most important routines are called UZD (solving the continuity equation) and SUD (solving the momentum equation).

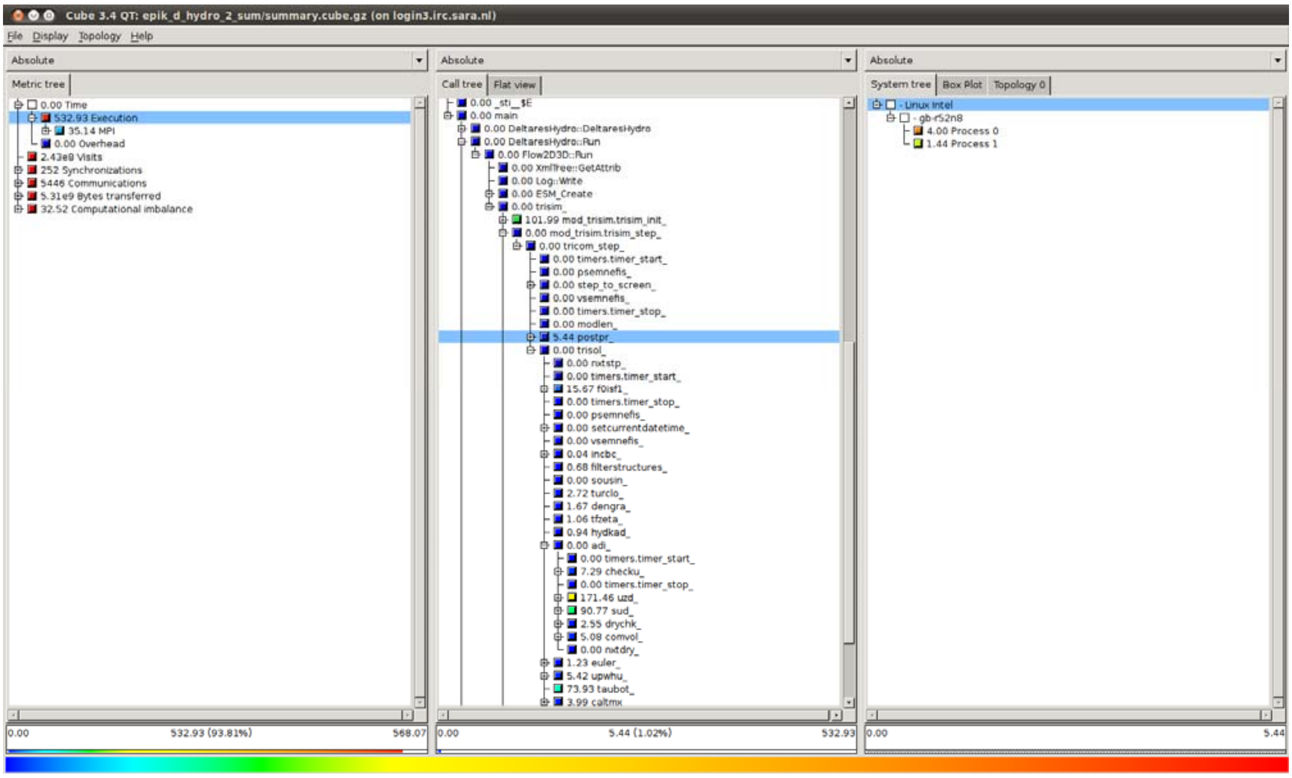

Figure 18: Profiling of Delft3D main loop using Scalasca

Gperftools was used to create a profile at the source-line level. It was not possible to get a profile of the full program, but it had to be restricted to the main loop with the routines ProfilerStart and ProfilerStop. A Fortran interface was written that uses the intrinsic module ISO_C_BINDING. Additionally, the profiler-library had to be linked with the Delft3D-executable, instead of being preloaded at runtime.

4.3 Elmer

ElmerSolver builds on generic finite element library functionalities that may be called by dynamically linked modules describing the particular physical phenomena. The library provides standard nodal elements up to 3rd degree, edge and face elements, and also p-elements up to 10th degree. The Elmer library also includes its own developments of standard preconditioned iterative methods and multilevel method. More recently a first version of a scalable FETI (finite element tearing and interconnecting) solution has been implemented within the PRACE project [22]. In addition interfaces to a number of external linear solver packages are provided.

The workplan regarding Elmer is divided into three subtasks described in the following sections. These are partially related for both improving the parallel performance and scalability of the code, but also in opening avenues for new classes of applications by including rotating boundary conditions.

4.3.1 *Hybrid parallelization*

As the number of processing units increases, the size of individual task often decreases. This means that the relative cost of communication in standard programming models based on the Message Passing Interface (MPI) increases. A partial solution for the problem is hybridization of the code so that within a shared memory hierarchy OpenMP pragmas are used to achieve a secondary layer of parallelism.

Elmer is built to be a MPI code and this form of parallelism is used at every level. In starting to use OpenMP for multithreading the starting point is rather modest. A few years ago some preliminary tests were carried out but at that time there was no benefit from using OpenMP compared to only using MPI With the current multicore processors the situation is changed and the number of cores addressing a node's shared memory will continue to increase.

In utilizing OpenMP some things can be achieved quite easily. If the bottle-neck is the matrix-vector multiplication (standard iterative methods) then the OpenMP pragmas take just a few lines of code. However, when multithreading is used to resolve the most time-consuming part new bottlenecks appear making the code revision more complex. In typical simulations the assembly process takes about 5-20% of the total effort. Therefore this needs to be addressed as well. In Elmer each Solver includes the loop over elements. Therefore each solver must be threaded separately.

4.3.2 *Rotating Boundary Conditions*

A generic feature that would be useful for industries is rotating boundary conditions. This would enable the use of Elmer in many fields where rotating machines are involved. Most importantly rotating problems involving the Navier-Stokes equation and magnetic fields are to be expected (possible applications could therefore include pumps and electrical machines, for example). It is hoped that the implementation could be such that it could be utilized directly by all currently existing solvers.

Elmer has a rather generic periodic boundary condition which involves a projection matrix P that maps fields, say u, between boundaries, say a and b, i.e. $u_b=Pu_a$. The machinery does not assume that the mesh is conforming. Unfortunately in this implementation the mapping matrix P is static, and even more importantly, the CRS matrix structure is assumed to be static. This machinery could be further developed to allow dynamically moving interfaces i.e. rotating boundaries.

There are two ways to utilize the projection matrix. One is to eliminate the variable on boundary b with the help of the projector. This will maintain the size of the matrix equation, albeit adding additional connections. The other possibility is to add additional equations as an additional constraint to the existing equation.

4.3.3 *Evaluation of FETI developments*

Within the PRACE-1IP project the FETI-1 and TFETI domain decomposition methods were implemented into Elmer [22]. This was done in collaboration between CSC and VSB. FETI based methods are very efficient for the parallel solution of large linearized problems in engineering and it has demonstrated (e.g. for linear elasticity) rather reasonable scalability up to ~3000 cores on Curie with 80 million degrees of freedom. There are however, some remaining bottlenecks requiring attention to achieve even better scalability.

The usual approach using massively parallel computers is to maximize the number of subdomains so that the sizes of subdomain stiffness matrices are reduced, which accelerates their factorization and subsequent pseudoinverse application, which are typically the most time consuming actions. On the other hand, a negative effect of that is an increase of the null space dimension and the number of Lagrange multipliers on subdomains interfaces, i.e. the dual dimension, so that the bottleneck of FETI [23] or TFETI [24] methods is the application of the projector $P=I-GT(GGT)-1$G onto the natural coarse space. The action time and level of communication depend first of all on the form of the natural coarse space matrix $G =$

$R^T B^T$, its distribution, and implementation of the coarse problem solution $GG^T\text{x} = \text{y}$. R denotes the matrix whose columns span the kernel of the stiffness matrix K and B denotes matrix with gluing or Dirichlet boundary conditions. Until now this coarse problem has been solved by one master process. Splitting this problem into an intermediate level would further improve the scalability of the FETI and simultaneously reduce the required memory per core. Consequently it would enable Elmer to solve large problems up to billions of unknowns.

In Elmer the matrix G is kept unassembled, i.e. in the form $G = R^T B^T$ which has an advantage concerning the sparsity pattern and fill-in. The number of nonzeros in G matrix is significantly larger than the sum of nonzeros in R and B matrices and therefore the matrix-vector and matrixtranspose-vector multiplications are faster using this unassembled form as compared to the case with assembled G matrix. The original implementation of the coarse problem solution contained the preprocessing of GG^T. More precisely, the cores owning the neighboring subdomains interchanged the corresponding data and sent the computed products to the master core to build sequential GG^T, to factorize it, and to solve sequentially all coarse problems. It is necessary to mention that all remaining cores have to wait and that the size of the solved coarse problem is limited by the master core's memory.

Within WP12 of the PRACE-2IP project different parallelization strategies of the coarse problem solution were suggested and tested [25] with regard to the improvement of the FETI massively parallel implementation. The best strategy based on MUMPS [26] was implemented into Elmer and the improvement was demonstrated on an engineering benchmark within WP9. Thus the method in the current Elmer FETI implementation is to assign the coarse problem solution to the group of cores selected from the communicator. Firstly the distributed matrix $\mathrm{GG^T}$ is formed by the chosen group of cores and then using MUMPS it is factorized in parallel to be prepared for the parallel coarse problem solutions.

4.3.4 *Results*

The hybridization with OpenMP process has been started. So far the sparse matrix-vector multiplication and the assembly process in some of the standard solvers has been threaded. The effect can best be seen in problems where these parts dominate. Practically this means the use of an iterative solution with Krylov subspace methods using either the diagonal or Vanka preconditioner.

As an example, a test case of a small Navier-Stokes (25000 nodes) simulation with BiCGStabl [22] and Vanka preconditioning was tested on a HP ProLiant DL580 G7 with quad-core Intel Xeon processors. The execution times are given in the following table. Up to 4 processors the speedup is superlinear and at 8 processors still almost linear. There are some operations that have not been threaded and therefore the performance eventually sacks. With the hybrid approach the number of cores may be increased by a factor of four maintaining the scalability. In the future more extensive tests will be performed and also the OpenMP pragmas will be extended to cover more parts of the code.

Number of cores	Time for solution (s)	Speed-up
1	341	-
2	130	2.6
4	69	4.9
8	47	7.3
16	38	9.0
32	37	9.2

Table 1: Effects of threading matrix-vector multiplication and the assembly process

In rotating boundary conditions some preliminary work has been carried out. The current methodology has been extended to vector valued fields where the periodicity of the components may require also rotation. This is relevant also for rotating boundary conditions in the case that only a part of the system (e.g. 60 degrees) is modelled.

Also another system to enforce continuity was added. This includes additional constraints by using Lagrange multipliers to the system rather than manipulating the existing matrix rows to achieve continuity. Which of the methods will be chosen with regard to the dynamic case is not yet clear. The work on the dynamic case will continue in autumn.

In the development of FETI significant advances were taken in conjunction with experts from VSB. Particularly noteworthy was the utilization of MUMPS for solving the coarse space problem in a given number of cores. This eliminates memory bottlenecks and slightly improves also performance. Until now the MUMPS library might be called for the local subdomain problems. MUMPS has the advantage that singular problems may be treated without the need to eliminate the null-space solution i.e. the rigid body motion of the local problem.

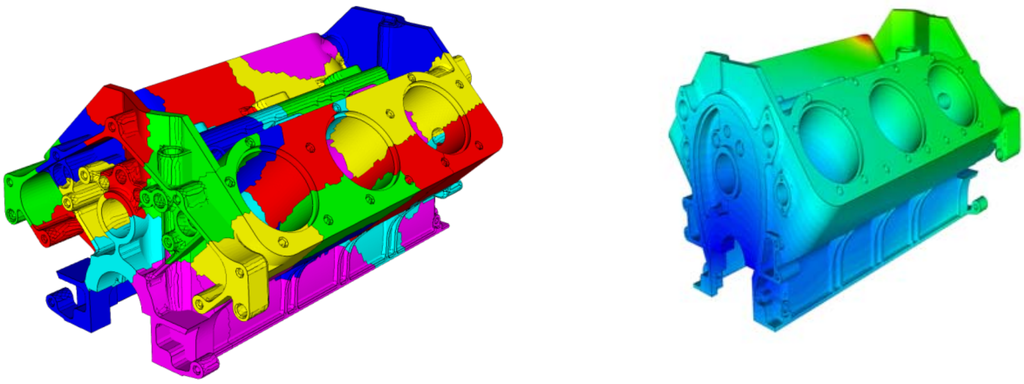

Figure 19: Block of car engine: left - domain decomposition, right - total displacement

Numerical experiments were run on matrices and vectors obtained from the decomposition and discretization of the 3D linear elasticity benchmark (see Figure 19). To illustrate the behaviour of this new strategy based on MUMPS for the coarse problem solution we first used the decomposition into 1,000 subdomains, primal dimension approx. 2.9 million DOFs, dual dimension approx. 500,000 , and null-space dimension 6 thousand. In Table 2 we report the speedup of the parallel coarse problem solution on HECToR at EPCC on the phase 2b (Cray XT6) machine. Although the speedup is negligible for this case, the big advantage is that there is no limitation on the coarse problem size in comparison with the coarse problem solved using only the master core which is obviously limited by its memory. Then we used the decomposition into 5,000 subdomains, primal dimension approx. 98.2 million DOFs, dual dimension approx. 13 million, and null-space dimension 30,000. In Table 3 we report the corresponding speedup of the parallel coarse problem solution. We see immediately that for this larger case the speedup is more relevant. Conclusion: this new strategy improves the FETI performance at least for very large problems and enables to solve larger problems than the originally implemented strategy due to distribution of the memory requirements, an important result for many usage scenarios.

#cores	4	10	13	16	33	67	100
speedup	0.66	0.92	1.05	0.92	1.03	0.95	0.83

Table 2: Speedup for the parallel coarse problem solution related to its solution by master core (2.9 million DOFs)

#cores	3	10	78	627	1253
speedup	0.98	1.23	1.25	1.13	0.99

Table 3: Speedup for the parallel coarse problem solution related to its solution by master core (98.2 million DOFs)

5 Conclusions

Some of the Open Source applications relevant for industry were identified through a survey. Further information was collected on the four applications that received at least an expression of interest in the survey. The aim of this work was to confirm the actual usage from industry of the two applications which received many expressions of interest in the survey, OpenFOAM and Elmer, and to assess the actual usage by industry for those that received only one expression of interest in the survey Delft3D and FEBio. Moreover the licensing terms and conditions and the existence of contacts with the development community were verified, these being considered a necessary requirement in order to establish a profitable enabling project. FEBio, it was discarded as it appears to be an academic project actually not widely used by industry, the developer group is located in the US and direct contacts could not be established, no clear indication could be found about contributions from external developers being accepted in the main development trunk, thus limiting the possibility of maintaining and making available to the user community the results of the enabling activity.

The selected applications were analysed from a technical point of view and preliminary results are reported. For applications like Elmer and OpenFOAM performance analysis and some of the bottlenecks are already known, Delft3D is an application where the PRACE team did not have any experience and so both profiling and benchmarking activities were necessary.

Activities will continue on the selected application to improve scalability and to test the results on industrial test cases.