SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

INFRA-2010-2.3.1 – First Implementation Phase of the European High Performance Computing (HPC) service PRACE

PRACE-1IP

PRACE First Implementation Project

Grant Agreement Number: RI-261557

D7.1.1
Applications enabling for capability science

Final

Version:	1.0
Author(s):	Jussi Enkovaara (CSC), Cedric Cocquebert (CEA), Alexander Schnurpfeil (FZJ)
Date:	17.06.2011

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: RI-261557**	
	Project Title: PRACE First Implementation Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: < **D7.1.1**>	
	Deliverable Nature: <Report >	
	Deliverable Level: PU *	**Contractual Date of Delivery:** 30 / June / 2011
		Actual Date of Delivery: DD / June / 2011
	EC Project Officer: Bernhard Fabianek	

* - 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:** <**Applications enabling for capability science**>	
	ID: <**D7.1.1**>	
	Version: <1.0 >	**Status:** Final
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2007	
	File(s): D7.1.1 - final.docx	
Authorship	**Written by:**	Jussi Enkovaara (CSC), Cedric Cocquebert (CEA), Alexander Schnurpfeil (FZJ)
	Contributors:	Peter Råback (CSC), Thierry Deutsch (CEA), Charles Moulinec (STFC), Andrew Sunderland (STFC), Iain Bethune (EPCC), Claudio Gheller (CINECA), Alan Simpson (EPCC), Peter Michielse (SARA), Iris, Christadler LRZ
	Reviewed by:	Daniel Ahlin (KTH), Thomas Eickermann (FZJ)
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	27/April/2011	Skeleton Draft	Initial version JE
0.2	23/May/2011	Draft	Internal call and preparatory access process described
0.3	29/May/2011	Draft	Collaboration with other tasks described

0.4	30/May/2011	Draft	Internal project reports included
0.5	04/June/2011	Draft	Feedback from F2F meeting included
0.6	09/June/2011	Draft	Feedback from ADS, PM, IC included
0.7	10/June/2011	Draft for internal review	TELEMAC report updated
1.0	17/June/2011	Final version	Modifications from internal review included

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, Applications enabling, Capability science

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[1] PRACE-1IP Project deliverable D7.4.1 - Applications and user requirements for Tier-0 systems.

List of Acronyms and Abbreviations

BLAS	Basic Linear Algebra Subprograms
BSC	Barcelona Supercomputing Center (Spain)
CCE	Cray Compiler Environment
CEA	Commissariat à l'Energie Atomique (represented in PRACE by GENCI, France)
CINECA	Consorzio Interuniversitario, the largest Italian computing centre (Italy)
CINES	Centre Informatique National de l'Enseignement Supérieur (represented in PRACE by GENCI, France)
CPU	Central Processing Unit
CSC	Finnish IT Centre for Science (Finland)
CSCS	The Swiss National Supercomputing Centre (represented in PRACE by ETHZ, Switzerland)
CSR	Compressed Sparse Row (for a sparse matrix)
CUDA	Compute Unified Device Architecture (NVIDIA)
DEISA	Distributed European Infrastructure for Supercomputing Applications. EU project by leading national HPC centres.
DGEMM	Double precision General Matrix Multiply
DMA	Direct Memory Access
DNA	DeoxyriboNucleic Acid
DP	Double Precision, usually 64-bit floating point numbers
EC	European Community
EPCC	Edinburg Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
EPSRC	The Engineering and Physical Sciences Research Council (United Kingdom)
ETHZ	Eidgenössische Technische Hochschule Zuerich, ETH Zurich (Switzerland)
ESFRI	European Strategy Forum on Research Infrastructures; created roadmap for pan-European Research Infrastructure.
FFT	Fast Fourier Transform
FP	Floating-Point
FPU	Floating-Point Unit
FZJ	Forschungszentrum Jülich (Germany)
GB	Giga (= 2^{30} ~ 10^9) Bytes (= 8 bits), also GByte
Gb/s	Giga (= 10^9) bits per second, also Gbit/s
GB/s	Giga (= 10^9) Bytes (= 8 bits) per second, also GByte/s
GCS	Gauss Centre for Supercomputing (Germany)
GENCI	Grand Equipement National de Calcul Intensif (France)
GFlop/s	Giga (= 10^9) Floating point operations (usually in 64-bit, i.e. DP) per second, also GF/s
GHz	Giga (= 10^9) Hertz, frequency =10^9 periods or clock cycles per second

GigE	Gigabit Ethernet, also GbE
GNU	GNU’s not Unix, a free OS
GPGPU	General Purpose GPU
GPU	Graphic Processing Unit
HMPP	Hybrid Multi-core Parallel Programming (CAPS enterprise)
HP	Hewlett-Packard
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
HT	HyperTransport channel (AMD)
IB	InfiniBand
IBA	IB Architecture
IBM	Formerly known as International Business Machines
IDRIS	Institut du Développement et des Ressources en Informatique Scientifique (represented in PRACE by GENCI, France)
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input/Output
JSC	Jülich Supercomputing Centre (FZJ, Germany)
KB	Kilo (= 2^{10} ~10^{3}) Bytes (= 8 bits), also KByte
KTH	Kungliga Tekniska Högskolan (represented in PRACE by SNIC, Sweden)
LBE	Lattice Boltzmann Equation
LQCD	Lattice QCD
LRZ	Leibniz Supercomputing Centre (Garching, Germany)
MB	Mega (= 2^{20} ~ 10^{6}) Bytes (= 8 bits), also MByte
MB/s	Mega (= 10^{6}) Bytes (= 8 bits) per second, also MByte/s
MFlop/s	Mega (= 10^{6}) Floating point operations (usually in 64-bit, i.e. DP) per second, also MF/s
MGS	Modified Gram-Schmidt
MHz	Mega (= 10^{6}) Hertz, frequency =10^{6} periods or clock cycles per second
MKL	Math Kernel Library (Intel)
ML	Maximum Likelihood
Mop/s	Mega (= 10^{6}) operations per second (usually integer or logic operations)
MPI	Message Passing Interface
MPP	Massively Parallel Processing (or Processor)
NAS	Network-Attached Storage
NCF	Netherlands Computing Facilities (Netherlands)
NFS	Network File System
OpenCL	Open Computing Language
Open MP	Open Multi-Processing
OS	Operating System
OSS	Object Storage Server
OST	Object Storage Target
PGAS	Partitioned Global Address Space
PGI	Portland Group, Inc.
PI	Principal investigator, a person responsible for research project.
POSIX	Portable OS Interface for Unix
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PSNC	Poznan Supercomputing and Networking Centre (Poland)
QCD	Quantum Chromodynamics
QDR	Quad Data Rate

QR	QR method or algorithm: a procedure in linear algebra to compute the eigenvalues and eigenvectors of a matrix
RAM	Random Access Memory
SARA	Stichting Academisch Rekencentrum Amsterdam (Netherlands)
SGEMM	Single precision General Matrix Multiply, subroutine in the BLAS
SGI	Silicon Graphics, Inc.
SIMD	Single Instruction Multiple Data
SMP	Symmetric MultiProcessing
SNIC	Swedish National Infrastructure for Computing (Sweden)
SP	Single Precision, usually 32-bit floating point numbers
STFC	Science and Technology Facilities Council (represented in PRACE by EPSRC, United Kingdom)
TB	Tera (= 240 ~ 1012) Bytes (= 8 bits), also TByte
TFlop/s	Tera (= 1012) Floating-point operations (usually in 64-bit, i.e. DP) per second, also TF/s
Tier-0	Denotes the apex of a conceptual pyramid of HPC systems. In this context the Supercomputing Research Infrastructure would host the Tier-0 systems; national or topical HPC centres would constitute Tier-1

Executive Summary

The WP7 "Enabling Petascale Applications: Efficient Use of Tier-0 Systems" in PRACE-1IP is responsible for providing petascaling support for European researchers for PRACE Tier-0 systems. The task 7.1 (Applications enabling for capability science) provides optimization service by organizing calls, by evaluating proposals for optimization projects and by performing the actual optimization work.

This is the interim deliverable of Task 7.1. The deliverable reports the calls organized during the first half of PRACE 1IP, and the evaluation processes we have used for reviewing the proposals and assigning work to PRACE partners. We report also some results of a user survey which are related to application enabling; the full results are reported in D7.4.1.

In order to ensure that technical work started as early as possible, Task 7.1 organized an internal call at the start of PRACE-1IP. Six optimization projects were chosen with the total of 28 PMs. Most of the projects achieved good progress in their targets, and for the two of the projects the results enable participation in future regular calls for project access.

The optimization projects which were selected in the internal call have finished, and the results obtained in the projects are reported here. Also the collaboration done with other WP7 tasks is reported.

The user survey conducted by task 7.4 has revealed a need for PRACE optimization service, however, it also indicates that the users could be informed more clearly about the possibilities offered by PRACE.

During the first half of PRACE 1IP, task 7.1 has finished two evaluation rounds for type C Preparatory Access proposals to the PRACE RI in which total of 7 projects were accepted and 23 PMs of PRACE work assigned. The third evaluation round is currently under way with 6 proposals requesting a total of 19 PMs. No preparatory access project has finished yet, thus no results are reported in this deliverable; these will be reported in D7.1.2.

1 Introduction

The role of task 7.1 is to enable researchers to efficiently use the PRACE infrastructure by providing petascaling and optimization services. The optimization tasks in 7.1 are short (6 month maximum) and intensive (up to 6 PMs), with the ultimate goal to improve the performance of an application on PRACE Tier-0 systems. Larger, long-term projects are handled by task 7.2 in collaboration with scientific communities.

The optimization service is available through two sets of competitive calls: an internal 7.1 call and regular preparatory access calls. As explained in Section 3, the internal call was a one-off case, the main route to optimisation service is within the PRACE RI preparatory access.

Currently, the task 7.1 is divided into three subtasks: subtask A develops best practices for a PRACE optimization service, which includes, for example, executing calls and defining the evaluation processes as well as participation in the user survey executed by task 7.4. The subtasks B and C focus on the actual optimization work on Jugene and Curie, respectively, as these are the currently available Tier-0 systems. When more PRACE Tier-0 systems become available, subtasks related to these system architectures will be defined.

This report is organized as follows: Section 2 presents the results of a user survey related to application enabling support. In section 3, we present how the initial internal call for projects was organized and in section 4 we discuss the current process for evaluating the regular preparatory access calls and organization of the related PRACE optimization work. Section 5 describes the collaboration with other tasks of WP7. In section 6 the results of internal projects are reported and section 7 concludes the deliverable.

2 User survey

The user survey was carried out by task 7.4, although task 7.1 contributed questions related to the part of the survey about applications enabling support. The full survey contained 50 questions, from which 13 (questions 6-18) were related to the application enabling. The responses were collected between 23rd November 2010 and 17th January 2011, and a total of 411 valid responses were received. The user survey is reported fully in D7.4.1, and we present here summary of the results related to task 7.1

Most applications listed in the survey have several users, and only less than 5 % are sole user of their code. About half of the applications have open source licenses, and only less than 10% of the applications can be used only for a single research project. Thus, there is large number of applications where the optimizations by PRACE should be relatively easy to incorporate into the main distribution due to open source licensing, and where the optimizations are likely to benefit a wider group of users. The majority of the users are also developers of the application which should help the collaboration between user and PRACE in the optimization work. Also, due to dual user/developer role, many of the users have at least some ideas about the factors limiting the performance of their application.

Less than 20% of the respondents are using applications that currently have the parallel scalability required for Tier-0 usage, that is scaling at least to 2048 CPUs (in Jugene system the limit is 8192 CPU cores). However, about 40% of the users would desire Tier-0 scalability, and both strong and weak scaling are important to users. Thus, clearly a significant amount of European HPC users would benefit from parallel optimization of applications. In addition to parallel scaling, memory usage might become a bottleneck in future systems. A large fraction of users requires more than 2 GB of memory per CPU core, and if future architectures provide only more CPUs without similar increase in the amount of memory this might become a problem.

Irrespective of the desire for Tier-0 usage, over 40 % of the users feel that their applications should be improved, and 15 % are interested in collaborating with PRACE in the optimization work. For 60 % of the users, the optimization effort is expected to be small or medium, thus fitting well into task 7.1's agenda of short optimization projects.

In summary, there is a significant number of users and applications that are interested in PRACE support for optimization work, and where the PRACE contribution can enable getting the research into a new level. However, the amount of users interested in PRACE support could most likely be increased by better disseminations of possibilities offered by PRACE. Reporting about projects where PRACE collaboration has been successful should be a good way to increase the interest in application enabling work with PRACE.

3 Internal call

The call for preparatory access was opened in early November 2010, with the first evaluation cut-off date in late January 2011. The actual consequence of this has been that successful applications which applied for preparatory access, could only be starting work in the March 2011 timeframe. In order to bridge the gap between the start of 1IP and March 2011, task 7.1 decided to have one internal call, to get applications enabling work started, in September/October 2010. The internal call enabled task 7.1 to prepare for actual preparatory access proposals in several ways. First, PRACE partners were able to get more familiar with available Tier-0 systems. The internal call also enabled task 7.1 to gather experience on how to assign PRACE experts to corresponding projects, and on how to inform project collaborators about accessing the different platforms. Also, the experts in WP7 got insight about which codes have computational weaknesses that we can realistically address; users might not submit these for preparatory access. Some codes and scientific problems are better suited for architectures other than BG/P and optimizing these codes for architectures similar to future PRACE Tier-0 machines such as Curie help WP7 to prepare for them. The internal call also benefitted users as it helped to ensure that more applications scale well.

The internal call was only meant for the initial phase of the project and will not be repeated. The preliminary allocation of effort for this call was 24-30 PM's (28 PMs were eventually awarded) from task 7.1, allowing WP7 to handle 4-8 projects depending on their size. Target of the projects was to increase scalability and improve the performance of applications to Tier-0 level, which here means the minimum partition size that PRACE requires for projects (30 TFlop/s). Selection criteria were based on innovation, scientific excellence and importance for a scientific user community. The bottlenecks and targets should be clear and well defined so that work required and the impact of the improvements may be judged. A clear scientific case with associated datasets should be defined and it was desirable to have the researchers/developers involved at least for advice giving role, although deeper collaboration was encouraged. Work was about to be performed on Tier-0 platforms, or on Tier-1 platforms where acceptable Tier-0 scalability can be tested.

3.1 Selection process

The selection process was based on WP7 partners proposing, and voting for applications to optimize. As this call was only for distributing work to partners, an internal selection without external reviewers was considered sufficient by the PRACE Technical Board. Internal selection ensured the process was short, so that work could be started as soon as possible.

Collaboration between the partners when forming the proposals was encouraged. Partners with PM's in WP7 were eligible to propose up to two applications for optimization. The proposals were circulated among the partners, and the partners were given the following criteria for reviewing and to base their votes on:

1. The work required to reach the relevant goals of the optimisation work should be 6 PM's or less. The selection criteria were:
   - Clearly defined bottlenecks and plans for how to address the problems
   - An initial set of people identified for doing the optimisation work.

2. The project should be useful to the general scientific community and potential PRACE customers. The selection criteria were:
   - The work would enable a new code, or a new group of researchers, to utilize PRACE resources
   - Possible to add the optimisations to the main distribution of the code.
   - The code is used by a wider community, and not only by a single group or a single institution.
   - Collaboration with scientific group
   - Collaboration between partners, or a will to do so.
3. The underlying scientific problems and the datasets used for enabling have to be potentially relevant for Tier-0. Currently, PRACE uses minimum Tier-0 constraints for regular access. The goal of 7.1 optimisation work is to reach these minimum constraints: 8000 cores on BG/P (Jugene)
   - 2000 cores on Nehalem QDR-IB machine (Curie)

Based on their review, partners voted for the applications with the following guidelines:

- Each partner had five votes
- Had to use all five votes, or none
- Each vote had the same value
- Only one vote per proposal
- Partners were allowed to vote for proposals from their own site

3.2 Voting results

The call was opened on October 6th and closed on October 30th, and 14 valid proposals were submitted by the time. The review and voting was carried out between November 1st and November 8th with the results which are presented in the table below:

Proposal	Proposing partner	Total votes	Total PM's	Cumulative
Million atom KS-DFT with CP2K	**EPCC**	**11**	**6**	**6**
Elmer Scaleup	**CSC**	**11**	**4**	**10**
BigDFT	**CEA**	**9**	**6**	**16**
Large Scale Simulations of the NonThermal Universe	**CINECA**	**7**	**2**	**18**
Optimizing TELEMAC-2D for Large-scale Flood Simulation	**STFC**	**7**	**4**	**22**

Semi-dilute polymer systems in shear flow – a particle based hydrodynamics approach	**FZJ**	**6**	**6**	**28**
Lattice QCD and HPC challenges	LRZ	4	3	31
Numerical seismic wavefield propagation in high-resolution complex 3D volcanoes	ICHEC	4	6	37
DALTON	SNIC	3	3	40
Petascaling Python	CSC	3	3	43
Porting and optimization of PLUMED to BlueGene/P	CINECA	3	3	46
MolKnot	GRNET	2	6	52
PLANE WAVE (Parallelized Adapted NEar-real- time WAVE propagation)	GRNET	2	6	58
REMOTE	ICHEC	2	6	64

Table 1: Result of voting in internal call

The PRACE Technical Board decided to support the first six projects in the table (in bold font to guide the eye). The cumulative column shows the cumulative PM requirements for the proposals. The motivation for the selection of six projects was that these will demand in total 28 PM's to complete, which was within the PM’s reserved for this activity. Also, the selected projects received at least five votes from other than the proposing partners, so the projects had wide support.

The selected applications presented a variety of scientific disciplines, varying from nanoscale materials science to astrophysics.

The results obtained in the selected internal projects are presented in Section 6.

4 Preparatory access calls

The main infrastructure for scientific users to obtain petascaling and optimisation services from PRACE is through regular preparatory access calls as announced at www.prace-ri.eu/hpc-access. There are 3 types of preparatory access: type A (for scaling tests) and B (porting and scaling/optimisation work) involve only the applicants, while proposals for type C request support from PRACE experts, as delivered by task 7.1. Currently, the systems available for preparatory access are IBM Blue Gene/P – JUGENE – hosted by GCS in Jülich, Germany with the maximum number of compute cores available of 294912, and Bull Bullx cluster – CURIE – funded by GENCI and installed at CEA, Bruyères-Le-Châtel, France. CURIE is based on general x86 architecture with a mix of thin and fat nodes interconnected through a QDR Infiniband interconnect.

The objective of preparatory access calls is to prepare users and their application codes for PRACE regular calls, which require good scalability.

4.1 Evaluation process

The preparatory access call was opened in November 8th 2011. The call is continuous, so that proposals can be submitted at any time, although, the proposals are evaluated only in fixed intervals. Currently, the evaluation cut-off is approximately every two months.

The type C proposals undergo both a technical evaluation by the computing centres hosting the requested Tier-0 system(s) and evaluation by task 7.1. The technical evaluation ensures that the project can be executed, e.g. that requested libraries, compilers etc. are available on the system.

For the 7.1 evaluation, a review group was formed in late 2010. Currently, the review group consists of 24 experts from different PRACE partners having varying scientific and high performance computing expertises. The purpose of the 7.1 evaluation is to ensure that the proposed optimisation work is feasible, and that the PRACE contribution is justified e.g. as the work benefits a larger group of researchers. There are two aspects in the 7.1 evaluation:

1. Actual evaluation (feasibility of work plan etc.)
2. Search for partners to perform the enabling work

The actual evaluation is performed by two members of the review group per proposal by answering following questions:

- Are the performance problems and their underlying reasons well understood by the applicant?
- Is the amount of support requested reasonable for the goals proposed?
- Will the code optimisation be useful for a broader community, and is it possible to integrate the development results achieved during the project in the main release of the code(s)?

The search for PRACE partners is done by the task leader; in order to keep the evaluation process reasonably short it is done in parallel with the actual evaluation. The process for finding PRACE partners is described in the following subsection.

4.2 Work assignment

The main guiding principles when searching for the PRACE partners for the optimization work are benefit to the project and equality among PRACE partners. Based on these principles, the PRACE partners (with enough remaining 7.1 effort) will be contacted in the following order:

1. A partner having a strong link with the proposal. This kind of link could be for example previous work with the application code by the partner. Same country of origin between the PI and PRACE partner is not necessarily a strong link in this case.
2. If no strong link exists, a brief summary about the requested work is sent to 7.1 mailing list so that interested PRACE partners can volunteer for the work.
3. If there are no volunteers, the list-of-experts is consulted and suitable PRACE partners are contacted directly. Suitable partners are those having knowledge e.g. about the algorithms or programming approaches used in the application code of the proposal.
4. From the volunteers or suitable partners, those with the least amount of previous enabling work with preparatory access proposals are contacted first.

If there are several partners with the same priority for work assignment, these partners will be contacted in random order.

Each accepted type C preparatory access project gets assigned a contact person from WP7 who is responsible for coordinating or performing the PRACE work and who acts as contact person between the applicant and PRACE.

4.3 Performed evaluations

After the launch of preparatory access call in November 2010, three evaluation rounds have been performed during the first half of PRACE 1IP with the following cut-off dates: January 26th, March 6th and May 8th. The number of type C proposals together with the requested PRACE PMs has been steadily increasing as shown in Figure 1.

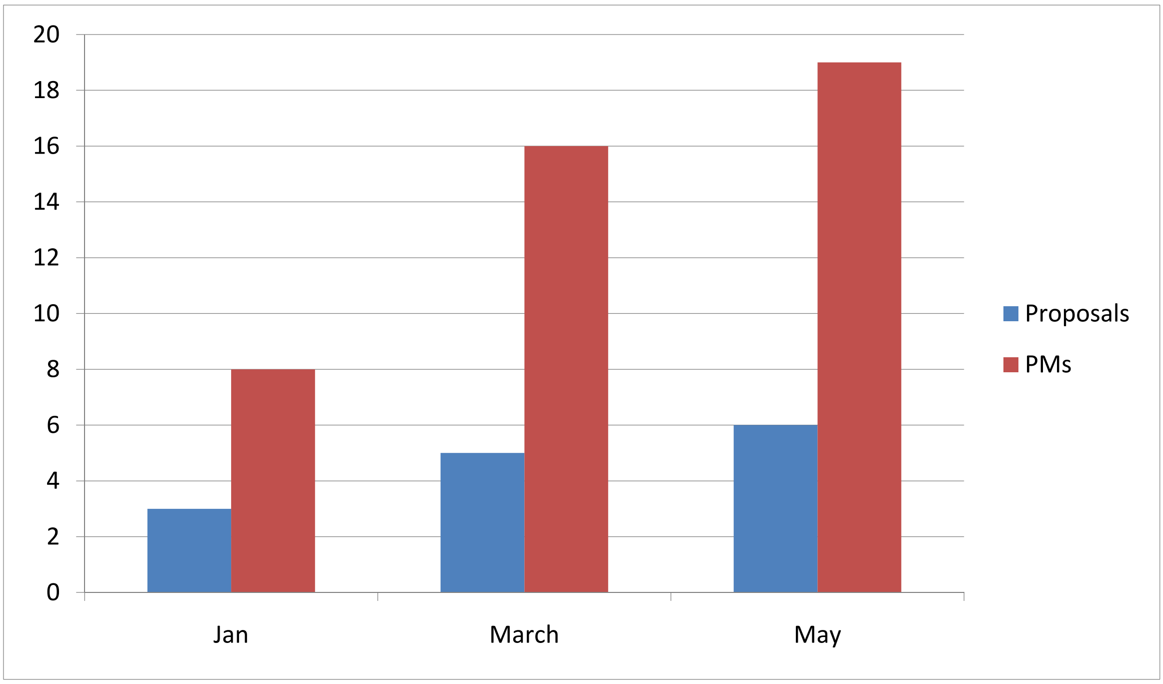

Figure 1: Number of submitted preparatory access proposals and requested PMs

Due to the high quality of proposals as well as good availability of expert support in PRACE, all but one proposal (requesting one PM) in January and March evaluations were accepted. The May evaluations have not been finished at the time of writing this document.

5 Collaboration with other WP7 tasks

Up to now, task 7.1 has collaborated with task 7.4 (Applications Requirements for Tier-0 Systems) in preparing the user survey, as well as with tasks 7.5 (Programming Techniques for High Performance Applications) and 7.6 (Efficient Handling of Petascale-Class Applications Data) within the internal optimization projects. Also, the reporting forms and templates for internal projects and for preparatory access projects were prepared jointly with task 7.2

In the Elmer project (see section 6.1), 7.6 has helped in implementing data reduction. This worked has enabled the Elmer code to save data values only on given boundaries or bodies, save point values in predefined uniform grid, save just the data on isosurface defined on-the-fly, and when using mesh split techniques, choose the coarseness level for saving. Together these techniques will enable a more clever choice of output data that supports the chosen visualization techniques. Also task 7.5 has collaborated in the Elmer project, by implementing an interface to Compressed Sparse eXtended (CSX) sparse matrix-vector multiplication as well as in the context of Finite Element Tearing and Interconnecting (FETI) domain decomposition method for linear problems.

In the CP2K project, task 7.5 has worked on implementing models and methods for reducing the volume of communication during parallel sparse matrix multiplication operations on distributed memory architectures.

More detailed description about the work done in collaboration with tasks 7.5 and 7.6 can be found in the corresponding project reports in section 6. It is expected that for the remainder of the projects in task 7.1, support from tasks 7.5 and 7.6 will be significant.

6 Results from internal projects

Most of the internal projects made good progress in their objectives. The objectives of the internal call were largely met, as PRACE experts obtained significant new knowledge about Tier-0 systems and about typical bottlenecks applications may face on such systems. At least two of the projects, "TELEMAC-2D" and "Non-thermal universe" are planning to apply for project access in the PRACE regular call based on the results obtained in the internal project. The results of the internal projects are summarized below; in addition each project delivered a white paper which will be published on the forthcoming PRACE training portal.

6.1 Elmer Scaleup

1.General information

Project name	Elmer Scaleup
Proposal reference number	2010PA0468
Scientific field of the project	
Project leader	Name: Peter Råback Affiliation: CSC – IT Center for Science Contact information:
PRACE staff involved (please give the information for all involved PRACE persons)	Name: J. Ruokolainen, M. Lyly, T. Kozubek, V. Vondrak et al. Affiliation: CSC, VSB Amount of work in person months: 4 (at CSC)
Computer system(s) employed	

2. Project information

Scientific goals of project
The target of the work is the Elmer finite element software suite. It has been developed, first as a national Finnish project, since 1995. In year 2005 Elmer was published under GPL which has increased the international usage quite dramatically. The user base of Elmer is perhaps some thousands of researchers around the world making it one of the most popular finite elements codes published under open source. *Of the usage of Elmer only quite a small part is HPC-related. Still Elmer has shown excellent scaling on appropriate problems up to thousands of cores and Elmer is one of the codes in the PRACE benchmark suite. Of the user communities the one working in the area of computational glaciology is perhaps the most significant. The 3D computation of true continental ice sheets requires supercomputing due to the large problem size. There exists eventually bottle-necks in all phases of the workflow: preprocessing, solution and postprocessing. There are also other blooming user communities in different application areas that could make use of the improved parallel performance.* *For more information on Elmer visit the homepage at http://www.csc.fi/elmer or the community portal at http://www.elmerfem.org.*
Computational approach
Elmer uses the finite element method for the solution of partial differential equations. The parallelization is achieved by domain decomposition. Usually iterative methods with blockwise preconditioning is used which require mainly the parallelization of the matrix-vector product. Also interfaces to massively parallel external libraries, such as Hypre, exist. In I/O the mesh generation in done in serial processing. Thereafter the mesh is partitioned and may be extruded or split into smaller elements on the parallel level. In postprocessing each partition writes its own data which must be fused by the visualization software.
Performance goals
On simplified problems (Poisson type problems & transient flow simulations) Elmer should scale even on Tier-0 systems with sufficiently large problems (weak scaling). Improvement on the parallel I/O is needed to also be able to utilize the results from the computations in a better way. On more difficult problems massively parallel scaling is not always obtained. Here the target is more moderate but continued work on preconditioners and domain decomposition methods is hoped to result to improved robustness. Then Tier-1 level is a realistic goal.

3. Results

Summary of results obtained

The scaling of Elmer was improved on all steps of the workflow.

In prepocessing the mesh splitting scheme was improved to allow the conservation of mesh grading for simple problems. When the mesh splitting scheme is sufficient for describing the geometry in detail it provides an efficient way of obtaining a mesh dense enough without any problems.

For the solution of linear systems FETI domain decomposition method is implemented. It utilizes a direct factorization of the local problem and an iterative method for joining the results from the subproblems. The scaling of FETI are shown to be almost linear with the size of the problem (see Table 1). FETI is also able to solve problems which so far lacked a robust method that would guarantee convergence. This is the case particularly for the Navier's equation but also better strategies for the Helmholtz and Stokes equation's may be obtained.

For postprocessing binary output formats and a HDF5 I/O routine were implemented. Both may be used for further parallel visualization with Paraview. The performance of HDF5 was not as good as expected but it is hoped that for larger tests the true virtues of HDF5 will be more eminent. However, HDF saved significantly the disk space compared to previous implementations, and also dramatically reduced the number of files.

N_{elem}/P	P	N_{dofs}	T(s)	N_{iter}
10^3=1000	3^3	107 811	0.62	21
	4^3	255,552	0.91	24
	5^3	499,125	1.13	26
	6^3	862,488	1.53	27
	7^3	1,369,599	2.27	27
20^3=8000	3^3	750,141	12.21	26
	4^3	1,778,875	11.32	29
	5^3	3,472,875	17.24	31
	6^3	6,001,128	19.72	32
	7^3	9,529,569	20.92	32
30^3=27000	3^3	2,413,071	66.8	25
	4^3	5,719,872	87.1	30
	5^3	11,171,625	93.3	31
	6^3	19,304,568	118	34
	7^3	30,654,939	110	34

Table 1: Scaling studies of the FETI method for Navier's equation using Cholmod for the factorization of the local problem and iterative projected CG method for the parallel problem. For a given number of elements (N_elem) per partition (P) the number of iterations (N_iter) and the time consumption (T) grow only slightly with the number of partitions. The tests were performed with HP CP4000 BL Proliant supercluster. With faster connections the scaling should be even better.

Benefits for the possible PRACE project proposal

The project was very beneficial for the development of Elmer towards massively parallel applications. The strong community support of PRACE was an important asset and it directed the work towards new frontiers, such as the implementation of FETI. The effective time-span of just a few months was rather short for implementing many new features. Therefore their testing was unfortunately not fully performed. If possible, it would be very desirable to continue this kind of collaboration within PRACE. Europe needs software development and PRACE provides a good hub for breaking the boundaries between countries.

6.2 BigDFT

1.General information

Project name	BigDFT
Proposal reference number	
Scientific field of the project	simulation of large atomic systems with DFT formalism.
Project leader	Name: Thierry Deutsch Affiliation: CEA Contact information:
PRACE staff involved (please give the information for all involved PRACE persons)	Name: Luigi Genovese, Brice Videau Affiliation: CEA / ESRF Amount of work in person months: 2PMs
Computer system(s) employed	

2. Project information

Scientific goals of project
In 2005, the EU FP6-STREP-NEST BigDFT project funded a consortium of four European laboratories (L Sim - CEA Grenoble, Basel University - Switzerland, Louvain-la-Neuve University - Belgium and Kiel University - Germany), with the aim of developing a novel approach for DFT calculations based on Daubechies wavelets. Rather than simply building a DFT code from scratch, the objective of this three-years project was to test the potential benefit of a new formalism in the context of electronic structure calculations. *As a matter of fact, Daubechies wavelets exhibit a set of properties which make them ideal for a precise and optimized DFT approach. In particular, their systematicity allows to provide a reliable basis set for high-precision results, whereas their locality (both in real and reciprocal space) is highly desired to improve the efficiency and the flexibility of the treatment. Indeed, a localized basis set allows to optimize the number of degrees of freedom for a required accuracy, which is highly desirable given the complexity and inhomogeneity of the systems under investigation nowadays. Moreover, an approach based on localized functions makes possible to control explicitly the nature of the boundaries of the simulation domain, which allows to consider complex environments like mixed boundary conditions and/or systems with a net charge.*
Computational approach
In the Kohn-Sham (KS) formulation of DFT, the electrons are associated to wavefunctions (orbitals), which are represented in the computer as arrays. In wavelets formalism, the operators are written via convolutions with short, separable filters. *Convolutions are among the basic processing blocks of BigDFT. Special care has to be taken regarding their performances. The CPU convolutions of BigDFT have thus been thoroughly optimized. The convolutions can be expressed with three nested loops.* *The wavelet properties are also of great interest for an efficient computational implementation of the formalism. Thanks to wavelets formalism, in BigDFT code, the great majority of the operations are convolutions with short and separable filters. The experience of the BigDFT team in optimizing computationally intensive programs made BigDFT a computer code oriented for High Performance Computing.* *Since late 2008, BigDFT is able to take advantage of the power of the new hybrid supercomputers, based on Graphic Processing Units (GPU). So far, this is the sole DFT code based on systematic basis sets which can use this technology.*
Performance goals
Two data distribution schemes are used in the parallel version of our program. In the orbital distribution scheme, each processor works on one or a few orbitals for which it holds all its scaling function and wavelet coefficients. In the coefficient distribution scheme each processor holds a certain subset of the coefficients of all the orbitals. Most of the operations such as applying the Hamiltonian on the orbitals, and the preconditioning is done in the orbital distribution scheme. This has the advantage that we do not have to parallelize these routines with MPI. The calculation of the Lagrange multipliers that enforce the orthogonality constraints onto the gradient as well as the orthogonalization of the orbitals is done in the coefficient distribution scheme. A global reduction sum is then used to sum the contributions to obtain the correct matrix. Such sums can easily be performed with the very well optimized BLAS-LAPACK libraries. Switch back and forth between the orbital distribution scheme and the coefficient distribution scheme is done by the MPI global transposition routine MPI ALLTOALL(V). For parallel computers where the cross sectional bandwidth scales well with the number of processors this global transposition does not require a lot of CPU time. Another time consuming communication is the global reduction sum required to obtain the total charge distribution from the partial charge distribution of the individual orbital. *In the parallelisation scheme of the BigDFT code another level of parallelisation was added via OpenMP directive. In particular, all the convolutions and the linear algebra part can be executed in multi-threaded*

mode. This adds further flexibility on the parallelisation scheme. Several tests and improvements have been performed to stabilise the behaviour of the code in multilevel MPI/OpenMP parallelization. At present, optimal performances can be reached by associating one MPI process per CPU, or even one MPI per node, depending on the network and MPI library performances. This has been possible also thanks to recent improvements of the OpenMP implementation of the compilers. Utilities to profile the code behaviour in are under implementation at the moment, such as to identify the possible bottlenecks at runtime.

The operation of the BigDFT code are well suited for GPU acceleration. Indeed, on one hand the computational nature of 3D separable convolutions may allow to write efficient routines which may benefit of GPU computational power. On the other hand, the parallelisation scheme of BigDFT code is optimal in this sense: GPU can be used without affecting the nature of the communications between the different MPI process. This is in the same spirit of the multi-level MPI/OpenMP parallilisation. Porting has been done within Kronos' OpenCL standard, which allows for multi-architecture acceleration.

3. Results

Summary of results obtained

We have evaluated the amount of time spent for a given operation on a typical run. To do this we have profiled the different sections of the BigDFT code for a parallel calculation. In Fig.1 we show the percent of time which is dedicated to any of the above described operation, for runs with two different architectures: French CCRT Titane platform (Bull Novascale R422) is compared to Swiss Rosa Cray XT5. The latter have better performances for communication, and the scalability performances are quite good. However, from the "time-to-solution" viewpoint, the former is about two times faster. This is mainly related to better performances of the linear algebra libraries (Intel MKL compared to Istanbul linear algebra) and of the processor. These benchmarks are taken for a run of the BigDFT code with the same input files (a relatively small Benchmark system of 128 atoms of ZnO), starting from the same sources. Then we have performed the same experiments, with the same system, different machines, to see how the relative performances between the communications and the libraries may influence enduser behaviour of the code. Results are presented in Fig.2. It can be seen that overall results may be significantly affected by these parameters. It is worth noticing that, in this case, Fig.1 shows that parallel efficiency is not always a significative evaluation parameter. In Fig. 3 the efficiency of the OpenMP parallelisation is presented for the full code in another test case (a B80 system). It is important to show that this is weakly affected by the number of MPI processes.

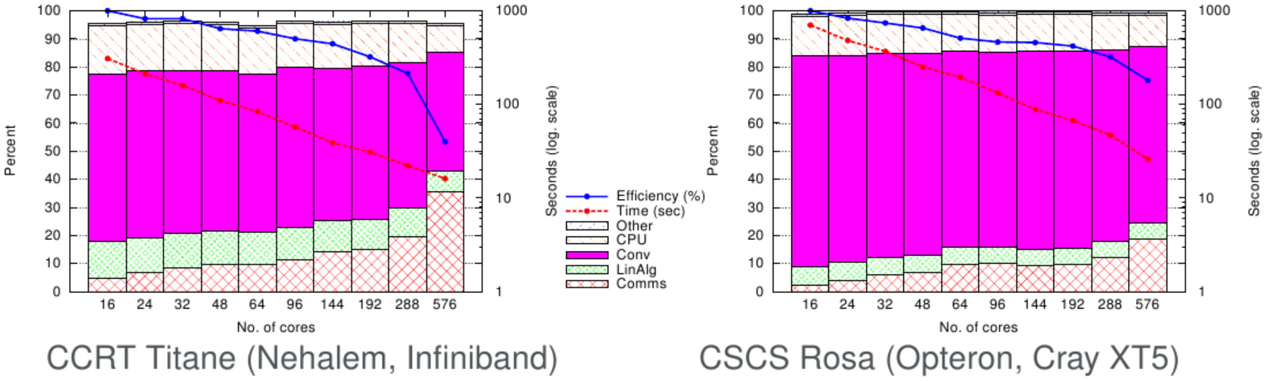

Figure 1: Comparison of the performances of BigDFT on different platforms. Runs on CCRT machine are worse in scalability but better in performances than runs on CSCS one (1.6 to 2.3 times faster).

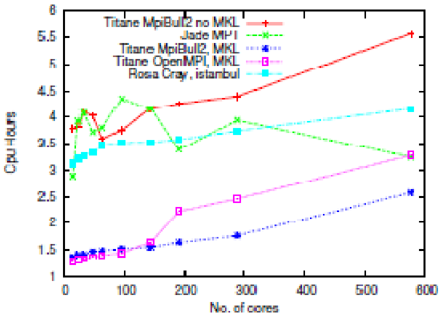

Figure 2: Number of CPU hours needed to terminate the job of Fig. 1 in different architectures with different MPI and linear algebra libraries.

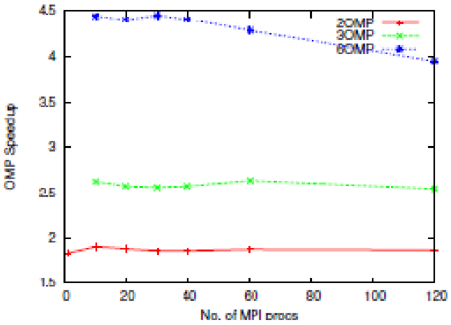

Figure 3: Speedup of OMP threaded BigDFT code as a function of the number of MPI processes. The test system is a B80 cagem and the machine is Swiss CSCS Palu (Cray XT5, AMD Opteron).

Benefits for the possible PRACE project proposal
In the following studies, the performance of the communications of BigDFT as a function of the number of MPI process will be the principal quantity. Indeed, in most of the architectures the MPI ALLTOALLV approach has revealed to be a bottleneck for systems with more than 1000 MPI processes. The main idea of this project is to understand how to reduce the overhead related to communications, by changing scheme and/or by a clever usage of multi-level MPI/OpenMP paradigms. We have already submitted a preparatory PRACE project and it is not yet started for our analysis, so outcomes for this will be important to understand our future actions (apply for a regular PRACE project for example).

6.3 Optimizing TELEMAC-2D for Large-scale Flood Simulation

1.General information

Project name	Optimizing TELEMAC-2D for Large-scale Flood Simulations
Proposal reference number	
Scientific field of the project	CFD
Project leader	Name: Charles Moulinec/Andrew Sunderland Affiliation: STFC Daresbury Laboratory Contact information; charles.moulinec@stfc.ac.uk, andrew.sunderland@stfc.ac.uk
PRACE staff involved (please give the information for all involved PRACE persons)	Name: Charles Moulinec/Andrew Sunderland/Yoann Audouin Affiliation: STFC Amount of work in person months: 4 PMs
Computer system(s) employed	IBM Jugene, IBM Pwr7 system

2. Project information

Scientific goals of project

With more and more people now living in flood-risk areas, it is essential to develop tools to assess the impact of a flooding on wetted regions, and ultimately to better warn people in advance of serious events. Numerical tools are of vital importance in aiding a better understanding of flooding impact. TELEMAC [1][2] enables, among other applications, the simulation of river systems, and can model free-surface flows, including flooding, wetting and drying. The system is highly portable and has been under development for over 20 years by EDF R&D with ~200 purchased licences that have been distributed worldwide. The whole system will go Open-Source in mid 2011, with TELEMAC-2D, the BIEF (Bibliotheque d'Elements Finis) and the pre-processing libraries already available to any user. TELEMAC-2D is based on the depth-integrated Shallow Water (hydrostatic) Equations when the horizontal length scale of the flow is greater than the vertical scale.

A research project between Bundesanstalt fuer Wasserbau (BAW, Karlsruhe, Germany) [3] and the Science and Technology Facilities Council (STFC, Daresbury, UK) [4] has recently been agreed to investigate flooding of the Rhine river from Bonn to the North Sea. The originality of this work resides in the fact that the flooding of this long section of river (about 250kms) will be undertaken in one simulation by a 2D approach (TELEMAC-2D) with a fine resolution of less than a metre in some parts of the mesh. This geometry has been meshed with 5M of elements. Some results already exist for portions of the Rhine river between Bonn and the North Sea, which have been studied by BAW, but the whole mesh has never been run. These intermediate data will be used for comparison. Two larger meshes have been identified to investigate the quality of the results and the sensitivity of the results to the grid size. The first mesh (20M) will be built by applying one level of refinement to the 5M element mesh and the second mesh by refining it twice (80M elements). Tier-1 systems can be used for the smaller cases, but simulations on Tier-0 systems are required to run calculations involving element meshes of 80M and beyond.

References

[1] TELEMAC system, http://www.telemacsystem.com
[2] Jean-Michel Hervouet, Hydrodynamics of Free Surface Flows: Modelling with the finite element method, Wiley, 2007.
[3] BAW, http://www.baw.de
[4] STFC Computational Engineering Group, Computational http://www.cse.scitech.ac.uk/ceg
5] METIS 5.0, http://glaros.dtc.umn.edu/gkhome/metis/metis/download
[6] Argonne Blue Gene /P Intrepid, http://www.top500.org/system/9158
[7] PT-SCOTCH, http://www.labri.fr/perso/pelegrin/scotch

Computational approach

The TELEMAC system is a multi-scale hydrodynamics free-surface suite able to solve Shallow Water Equations (TELEMAC-2D) and Navier-Stokes Equations (TELEMAC-3D) depending on the topology of the configuration and the approximation in the calculation of the vertical velocity. The system relies on the BIEF Finite Element Library. This library contains basic operations, a few linear solvers, and some of the discretisation schemes used in the hydrodynamics solvers. As the scientific project aims at solving the Shallow Water Equations, the following description is restricted to the computational properties of TELEMAC-2D. The steps to perform a simulation with the TELEMAC system proceed as follows:

- *Generation of the grid (triangular elements), with a mesh generator taking into account the bathymetry. This step is serial with the current existing tools. However it is possible to globally* refine an existing mesh to increase resolution. This is also performed in serial, but with a tool that has been recently optimised,

- Pre-processing including mesh partitioning by METIS 5.0 (serial version) [5] and calculation of the connectivities, boundary conditions, halo cells, and pre-processing for the method of characteristics for advection (if used). The mesh partitioning and all other pre-processing tasks are performed by the same tool, i.e. PARTEL. Serial mesh partitioning is limited by memory availability whereas the rest of the pre-processing tasks are limited by time constraints. There exist two versions of PARTEL, a fully serial one (PARTEL1) and a partially parallel one (PARTEL2), which runs partitioning serially, but perform the rest of the pre-processing in parallel. This version uses global arrays, because it was designed to speed-up the pre-processing process, therefore to date no optimization in terms of memory has been undertaken,
- TELEMAC-2D relies on the shallow water equations. The equations might be solved coupled or with the help of a wave equation, depending on the option chosen. The space discretisation is linear. Several advection schemes are available and used depending on the flow, namely, the method of characteristics, the streamline-upwind Petrov-Galerkin (SUPG), Residual Distributive Schemes (N-Scheme and Psi-Scheme). Matrix-storage is edge-based. Several linear solvers are available in the BIEF library, e.g. Conjugate Gradient, Conjugate Residual, CGSTAB and GMRES. TELEMAC-2D is fully parallelised by MPI.

Input files consist of a parameter file (ASCII) read by all the processors, and a geometry file (binary, SELAFIN format) and a boundary file (ASCII) per MPI task read by each processor. Those files are generated by one of the PARTEL tools (either serial or parallel), which prepares the initial geometry (binary, SELAFIN format) and boundary files (ASCII) for each MPI task.

Output files are handled in the same way, with a result file (binary, SELAFIN format) per processor, as well as another output file (ASCII) per processor, showing the evolution of the simulation. The result file (binary, SELAFIN format) can also be used to restart a simulation.

Performance goals

TELEMAC-2D has been successfully run on Argonne BlueGene/P (BG/P) Intrepid up to 16,384 cores for a straight channel demonstration case of 25M elements. Here, performance was shown to scale very well up to 16,384 cores (VN mode). To generate input data for these jobs PARTEL was run serially over a long period of time for grid partitioning (using METIS 5.0) on an IBM Pwr7 cluster. Evidently, the parallel PARTEL2 would perform pre-processing for the 25M case much quicker, but the memory overheads, due to replicated data structures are limiting the problem sizes that we can address. Moreover, another restriction with PARTEL2 is that pre-processing runs require the same number of parallel tasks that the target calculation with TELEMAC-2D (25M case required 16K cores of BG/P in SMP mode to pre-process). In fact the pre-processing of any grid with > 25M elements for subsequent Tier-0 simulations therefore requires a new memory-optimized parallel version based on PARTEL1. In the existing parallel PARTEL2 the mesh partitioning is still serial, performed by serial METIS-5.0, but the remainder of the pre-processing calculation, i.e. the search for the connectivity between sub- domains, the construction of the boundary conditions and of the halo cells, is parallelized. This first parallel version of PARTEL2 has been designed to speed-up the pre-processing stage without any consideration to optimize memory usage. It is re-designing this aspect of PARTEL that we have therefore targeted for optimization in this project, with the specific aim of facilitating runs on Tier-0 systems using grids composed of over 25M elements. This will also allow pre-processing with PARTEL to take place on differing core counts (i.e. significantly lower but still parallel jobs on Pwr7 cluster) to the total MPI task count using TELEMAC-2D (i.e. large numbers of cores on IBM BG/P). This feature is described in the results section and whitepaper.

The main scientific project objective is to facilitate the running TELEMAC-2D to simulate flooding due to occur in the region of the Rhine river with a 80M element grid model and beyond.

To fulfill this objective, parallel PARTEL will need to be re-written particularly in terms of memory usage to be able to handle more than 25M elements. The demonstration case of a 200M element grid will be used to test this new tool. The target in this case is to enable TELEMAC-2D runs on the 80M and/or 100M+ datasets on Tier-0 machines such as Jugene and Curie by preparing suitable parallel partitioned inputs using this new optimized tool.

3. Results

Summary of results obtained

There were two stages to the proposed optimization workplan:

1) *Localize the majority of global arrays to break the barrier of ~25M elements that PARTEL2 can now handle. The demonstration case would be used for testing datasets of up to 200M elements on up to 32,768 cores. This limit of 32,768 sub-domains is set by METIS which requires more than 256GB RAM for partitioning into 65,536 sub-domains,*

2) *Implementing PT-SCOTCH [7] instead of METIS 5.0 to break the 32,768 core limit and thereby enable future runs of TELEMAC-2D on larger numbers of cores (e.g. up to ~225+ Tflops Jugene peak)*

Summary of Results (further details will be described in associated whitepaper)

1. *The pre-processing stage routine in partel.f has been re-written and optimized in order to be run on NTASKS (MPI tasks in PARTEL) (to date typically up to 3 256GB RAM multicore nodes of an IBM Pwr7 cluster) to deal with up to 100K NSUB subdomains. This pre-processing stage is still split into two stages, with Fortran files, and has been optimized as follows:*
   - *The first stage run on NTASKS cores is the reading of the original mesh, then partitioning it into NSUB subdomains and writing 2 files per NTASKS cores. These two files contain information for NSUB/NTASKS subdomains. The first of those two files contains the geometry quantities (position and connectivity between elements and nodes) and the second one information concerning boundary conditions and interfaces between subdomains.*
   - *The second stage is also run on NTASKS and reads the output of the first stage, the data is now distributed, thereby reducing markedly the local memory consumption. The outputs from this stage are the geometry and boundary files readable by TELEMAC-2D.*
   - *First results of the new pre-processing stage applied to a 200M element grid partitioned into 32,768 subdomains on 8 and 24 MPI tasks on the Pwr7 cluster are now obtained in close to three hours (9524 secs) rather than the previous run-times of several days with PARTEL1. The results from the new tool have been verified using different parallel runs with NTASKS=8 and NTASKS=24.*

2. *Metis 5.0, ParMETIS, Scotch 5.11.1, and PT-Scotch 5.11.1 have all been implemented in the pre-processing stage. A demonstration case run on the Pwr7 cluster has shown that serial Scotch 5.11.1 is able to partition a 400M element demonstration case into 294,912 subdomains. This has fully prepared suitable partitioned grids for future parallel runs using large numbers of cores (up to the largest available job size) on Jugene for the large datasets described in this project.*

Benefits for the possible PRACE project proposal

Large-scale simulations have now been prepared for future large-scale runs on Prace Tier-0 systems. We plan to apply for a regular PRACE project for TELEMAC-2D. This will allow us to undertake the large-scale simulations on Jugene and Curie (and any other Tier-0 systems) that have been facilitated by this initial project. It is expected that IO overheads in Telemac-2D will begin to diminish scalability on large core counts and this could be the focus of any follow-on project.

6.4 Million Atom KS-DFT with CP2K

1.General information

Project name	Million Atom KS-DFT with CP2K
Proposal reference number	PRA1IC
Scientific field of the project	Computational Chemistry / Materials Science
Project leader	Name: Iain Bethune Affiliation: EPCC Contact information: ibethune@epcc.ed.ac.uk
PRACE staff involved (please give the information for all involved PRACE persons)	Name: Iain Bethune, Adam Carter, Xu Guo Affiliation: EPCC Name: Paschalis Korosoglou Affiliation: GRNET/AUTH Amount of work in person months: 6 planned, 6.2 actual
Computer system(s) employed	Jugene (also Palu, Cray XE6 at CSCS)

2.Project information

Scientific goals of project

Linear scaling KS-DFT is a 'holy-grail' for the community of computational chemists, physicists and material scientists. A significant amount of research has lead to various algorithms and (serial) reference implementations. However, the crossover point where O(N) algorithms can compete with traditional O(N^3) is surprisingly far. Applications to three dimensional, condensed phase systems with good accuracy are therefore essentially absent. Only in combination with massive parallelism can large systems be computed in reasonable time. On the other hand, a massively parallel implementation enables a completely new scientific field to be explored. Ideal linear scaling combined with perfect (weak) parallel scaling will allow systems of near arbitrary size to be computed in constant time provided the computational resources are available. As petascale and exascale resources become available, calculations will abruptly move into the regime where O(N) makes perfect sense and model size will essentially be limited by the curiosity and requirements of researchers.

Possible large (~1,000,000 atom) systems we could run as a result of this project include:
1) A solvated small virus. The Satellite Tobacco Mosaic Virus (STMV) has only been recently simulated (2006) by the Schulten group using classical molecular dynamics with a model of slightly more than 1M atoms (see e.g. http://www.ks.uiuc.edu/Research/STMV/).
2) A next generation nanotransistor - for example computing the strain within a single-gate ultra thin body nanoelectronics device. Simulations of these systems are currently run on Jaguar using empirical models with the OMEN and NEMO codes (Klimeck et al.).
3) A grain boundary between two TiO_2 nanocrystals.

Computational approach

Linear scaling KS-DFT in CP2K is based on a sparse density matrix formulation, and as a result the Self Consistent Field (SCF) loop is reduced to a series of parallel sparse matrix multiplications. These are performed by the Distributed Block Compressed Sparse Row (DBCSR) library, which is designed for massively parallel MPI/OpenMP operation. The matrix multiply is based on Cannon’s algorithm, and involves mostly nearest neighbour communication. For the large systems of interest, the runtime is fully dominated by these matrix multiplies.

Performance goals

Weak scaling experiments have been performed with realistic accuracy settings for bulk liquid water up to ~140000 cores of JaguarPF and 560000 atoms. This gives an indication about the requirements for the virus simulations. Weak scaling is approximately obtained, and ongoing developments will further improve this. The data suggests that one iteration can be performed on 1M atoms using 220000 cores in ~500s. With the current implementation, the expected number of iterations is about 100, and hence the calculation will be possible within 24 hours of computing on a petascale computing facility.

As important as the pure performance is, however, the memory bottleneck is a real concern on the low memory nodes of the BlueGene/P. In CP2K, all 'significant' data structures (matrices, grids) are fully distributed, but information related to the atoms (basis, coordinates, topology) is not. First testing indicates that for the virus, the latter amounts to 1Gb of data per MPI process. While this is not a real concern on JaguarPF (16Gb/node) this will be an important issue on Jugene.

3. Results

Summary of results obtained

Porting and verification:
CP2K was successfully compiled in MPI-only and mixed-mode MPI/OpenMP versions on the BlueGene/P. The entire CP2K regression test suite of over 2000 tests was run for both versions. The MPI version passed 80% of the tests. A number of tests did not complete due to possible memory issues. The mixed-mode version passed 61% of tests. Many of the tests did not complete, and the reasons for this are not yet fully understood.

Memory reduction:
Memory usage reporting was implemented for the BlueGene/P using an ISO C Binding interface to a C system library call. Development was undertaken to replace a key linked list data structure, whose size scales with the number of atoms, with an array-based implementation. This was successful, although it did not give the expected memory reduction (~17%), so investigation is still ongoing

Benchmarking:
As the memory reduction code was not completed successfully, it was not possible to benchmark using the large STMV system. Instead, a smaller liquid water benchmark was used (6144 atoms). This showed good strong scaling up to 8192 MPI processes, with the vast majority of the runtime concentrated in the DBCSR matrix multiplication routines as expected. Attempts to benchmark the mixed-mode executable failed due to (presumably) the same issue causing the regression tests to hang.

Benefits for the possible PRACE project proposal

This is a PRACE WP 7.1 internal project – no PRACE project application is planned directly as a result of this work. However, another internal project in WP 7.2 is ongoing to improve the OpenMP provision in the code, and to encourage the wider CP2K user community to use the mixed-mode version of the code.

6.5 Large Scale Simulations of the Non-Thermal Universe

1.General information

Project name	Large Scale Simulations of the Non-Thermal Universe
Proposal reference number	PRA2IC
Scientific field of the project	Cosmology
Project leader	Name: Franco Vazza Affiliation: Jacobs University, Campus Ring 1, 28759 Bremen, Germany Contact information: f.vazza@jacobs-university.de
PRACE staff involved (please give the information for all involved PRACE persons)	Name: Claudio Gheller Affiliation: CINECA Amount of work in person months: 3 Name: Maciej Cytowski Affiliation: ICM Amount of work in person months: 3
Computer system(s) employed	

2. Project information

Scientific goals of project

The overall objective of the project is to employ the cosmological Adaptive Mesh Refinement code ENZO (Bryan et al. 1995; O'Shea et al. 2004; Norman et al. 2007), extended by several new numerical implementations produced by our group, to study with unprecedented spatial resolutions non thermal phenomena (shocks, turbulence, Cosmic Rays acceleration, magnetic fields and Active Galactic Nuclei feedback) active in massive galaxy clusters during their cosmological evolution. This will represent a big step forward for the achievement of a holistic view of non-thermal phenomena related to galaxy clusters in the evolving Universe.

This objective can be pursued only exploiting large HPC systems, that provide the computational resources necessary to achieve the requested degree of resolution and accuracy and to treat complex physical processes. The EU funded project PRACE, supports the scientific research by providing computational resources on the largest HPC systems available in Europe (Tear-0 systems). This work is a preparatory phase aiming at enabling the ENZO code to run efficiently and effectively on the Jugene, Blue Gene/P system available at the Forschungszentrum Juelich in Germany. Such preparatory work will enable our group to submit a scientific proposal in one of the next calls for large scale projects in the PRACE framework.

Computational approach

ENZO is an adaptive mesh refinement (AMR) cosmological hybrid code highly optimized for supercomputing ENZO couples an N-body particle-mesh solver describing the dark matter, with an adaptive mesh method for ideal fluid-dynamics (Berger & Colella, 1989), that follows the evolution of the baryonic component. The two components are coupled via gravitational field, calculated by means of a FFT-multigrid based approach. The fluid dynamics is solved adopting an Eulerian hydrodynamical solver based on the the Piecewise Parabolic Method, that is a higher order extension of Godunov's shock capturing method. The PPM algorithm is at least second-order accurate in space (up to the fourth-order, in the case of smooth flows and small time-steps) and second-order accurate in time. In the cosmological framework, the basic PPM technique has been modified to include the gravitational interaction and the expansion of the Universe.

The AMR approach adopts an adaptive hierarchy of grid patches at varying levels of resolution. Each rectangular grid patch (referred to as a "sub-grid") covers some region of space in its parent grid which requires higher resolution, and can itself become the parent grid to an even more highly resolved child grid. ENZO's implementation of structured AMR poses no fundamental restrictions on the number of grids at a given level of refinement or on the number of levels of refinement. However, owing to limited computational resources it is practical to institute a maximum level of refinement. Additionally, the ENZO AMR implementation allows arbitrary integer ratios of parent and child grid resolution, though in general for cosmological a refinement ratio of 2 is use.

The code is parallelized by domain decomposition into rectangular sub-grids, including the top/root grid (which is the only level in a non-AMR run). Message passing paradigm is adopted and implemented by means of the MPI library, I/O makes use of the HDF5 data format (see http://www.hdfgroup.org/HDF5/).

The public ENZO code has been extended by our group with a number of ad hoc techniques and algorithms, which were successfully applied to various ENZO runs.

Performance goals

Our goal is to perform simulations of a cosmological box of 100 Megaparsecs (Mpc) with a resolution of the order of a few tens of kpc. This can be reached by:

3) using a uniform computational mesh (UNIGRID approach) of linear size of about 2000-4000 cells
4) using an AMR grid, with base grid of about 500 cells and 2 to 4 refinement levels

The total computing time should be result of the order of a few millions of CPU hours.
The number of computational nodes should be such that:

5) enough memory is available
6) the efficiency is still acceptable (>0.5)

3. Results

Summary of results obtained

Accomplished work:

1. porting and optimization of ENZO and the associated initial conditions generator INITS on the Jugene platform. In order to compile ENZO, a specific Makefile has been created, through the following basic steps: i) the choice of the suitable cross-compilers and libraries; ii) the specification of the proper compiling and linking flags; iii) the definition of some preprocessor variables to execute pieces of code specific for the architecture. in addition to essl and xlf90_r libraries (and, of course, HDF5 and zlib), which are required also for enabling ENZO on other platforms (e.g. IBM SP Power6 or Linux clusters), further libraries, such as xlfmath and pthread, are needed for the Blue Gene/P.

2. Optimization: meaningful effort was necessary to improve the application performances, which in its default configuration exhibits only a poor scalability. A detailed profiling of the application has been obtained using both GPROF and SCALASCA profilers. Such an analysis allowed to detect the main sources of performance loss, which are:

- the default transposing of the grid when computing the FFT for the gravitational potential;
- the default mode for gravity calls.

The improvement of the performances was possible by properly configuring ENZO, by means of suitable setup options. This, however, resulted to be a challenging task, since such options are not documented and the optimal setting was found only analyzing the details of part of the source code. A further limit on ENZO's scalability is intrinsic to the FFT solver. The parallel FFT library used by ENZO, in fact, adopts a planar domain decomposition, that cannot scale above the 1D size of the computational grid (N_{1D}). The improvement of this feature, however, would have required an effort beyond the possibility of this work and has been neglected, noticing also that the performances, however, tend to improve for larger meshes (i.e. problem size), since the problem size grows much faster (depending on the cube of the linear size) than the number of processors, leading to an increasing per-processor workload. Furthermore, the FFT has a lower impact, its computational cost scaling with N_{1D} log(N_{1D}).

3. INITS parallelization: in order to generate initial conditions on a large computational mesh, the initial conditions generator, INITS (available only in its serial version), has been parallelized. The work accomplished to parallelize INITS can be summarized as follows:

- Compilation, testing and performance analysis on a x86_64 node.
- Parallelization (described in details below).
- Analysis of the correctness of the results.
- Compilation and test run of the parallel code on x86_64 cluster.
- Compilation and test run on IBM Blue Gene/P system.
- Performance analysis on IBM Blue Gene/P system.

Particularly relevant the adoption of P3DFFT as the parallel FFT library. P3DFFT, in fact, apart from ensuring good performances. it supports a rectangular, rather than plane parallel, domain decomposition. This allows to overcome the typical limitation imposed by plane parallel FFTs, that scale with the linear size of the computational mesh. The P3DFFT library allows to scale with the square of the linear size, permitting to exploit a much larger number of processors (hence, much larger computational volumes). Data writing was implemented using the HDF5 parallel library. The quality of the results has been checked comparing them to those generated by the original INITS code, using the h5diff utility, that performs a bitwise comparison between the datasets in the HDF5 files. Only about 10% of the numbers differ, but the difference is always at the precision level of our double variables. This differences does not have any consequence on the corresponding simulations that gives exactly the same results in the two cases.

4. Tests and benchmarks: In a first stage, we have performed a number of tests using computational meshes with sizes that are currently adopted for our high-end production runs (512^3, 1024^3). In this way, we can easily analyze the performances that can be obtained on the Blue Gene/P architecture and compare them with the results achieved on different HPC systems. These results can be used to infer the behaviour of the code on larger Jugene's configurations. The tests can follow two different approaches. UNIGRID simulations are performed on a constant resolution cubic mesh, while AMR simulations adopts a grid refining method, increasing the spatial resolution where this is required. Both approaches can be used for the applications targeted by the present project, therefore we have analyzed their performances and suitability in detail before choosing a possible production set-up.

UNIGRID: the analysis of the speed-up curves shows that the performance improves with increasing the number of processors, even not linearly, giving the best result with a maximum number of processors twice the linear size of the computational mesh. This limit depends mainly on the calculation of the gravitational field, that poses a limit on the scalability of the code to a number of processors (np) that is 2-4 times the linear size of the mesh. Using a number of processors equal to the 1D size of the problem, efficiency for both 512^3 and 1024^3 is extremely good (about 0.75 the former, 0.9 for the latter); it is still acceptable (about 0.46 and 0.65 respectively) doubling np, but unacceptable for larger np values. However, as can be noticed also for the speed-up, the situation tends to improve with increasing the size of the problem. Our conclusion is that, for larger configurations (e.g. 2048^3, 4096^3), an acceptable efficiency can be obtained even using a number of processors 4 times the 1D size.

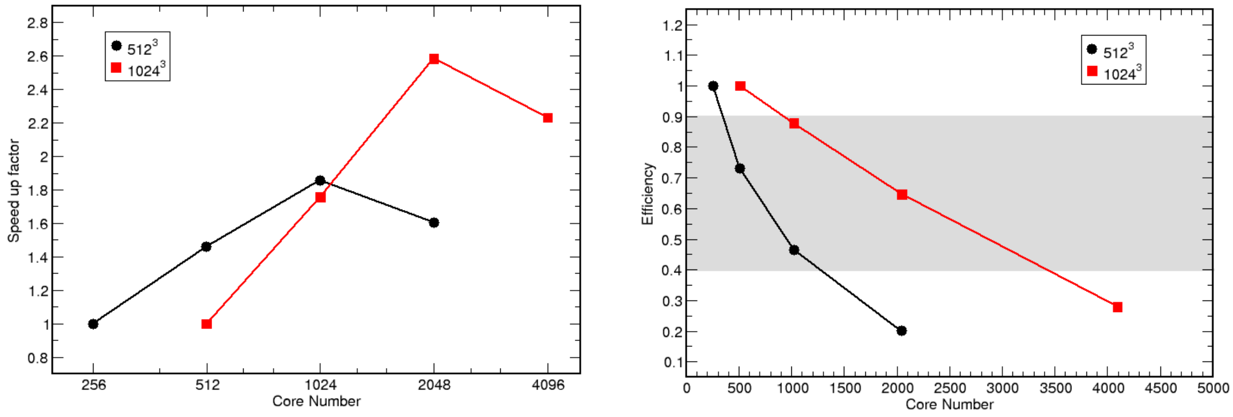

Figure 1: Speed-up (left) and efficiency (right) of the 512^3 and the 1024^3 tests.

AMR: AMR allows to get high resolution only where this is required. AMR can achieve the same effective resolution of a UNIGRID run, requiring, for the physical variables, much less memory. The memory estimate, however, is much more uncertain, since it depends on the evolution of the simulated system, on the maximum number of refinements levels allowed and on the refinement criteria. The same holds for the computing time. The main drawback of the AMR approach, is the presence of the Hierarchical Tree (HT) structure, that is responsible for the management of the refined regions. No parallelization is implemented on the HT, since its information are necessary to all processors for an efficient and scalable implementation of the code. This however leads to a large overhead of memory, that strongly limits the maximum size and resolution at which the simulation can be accomplished, especially when memory size of each computing node is small. A number of tests have been performed in order to check the size of the HT with the simulation time. In all the tests, the size of the HT exceeds that of the node memory well before the end of the simulation. We then conclude that the AMR configuration is not suitable for our application on the Jugene system.

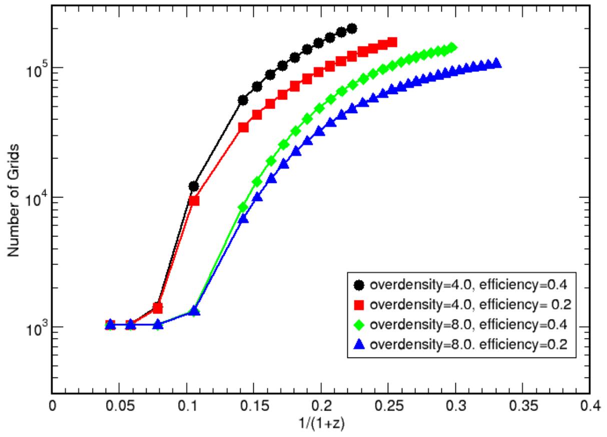

Figure 2: Number of sub-grids generated in various AMR tests, performed changing the mass overdensity and the efficiency parameter used for the grid refinement criterion, as a fuction of the redshift, z. The redshift is a quantity used to define the cosmic time: z=0 is the current (end of the simulation) time. The curves end when the requested memory per overcomes the available one.

5. Final Setup, tests and benchmarks

The results of our work led to the detailed specification of the optimal case to be run on the Jugene platform, according both to the requirements of our scientific target and to the performances and features of the available computing resources. The following conclusions can be stated:

- The size of Jugene's node memory, prevents us to adopt an AMR approach, since the hierarchy tree of the whole computational box is replicated on each node. At most about 150000 tree nodes (sub-grids) can be stored and managed, that are far too few to complete a cosmological run in any meaningful setup tested. UNIGRID configuration must be adopted.
- Scalability of the code in UNIGRID mode allows to use at most a number of processors about four times the 1D size of the computational mesh.

According to the scalability tests previously presented, the optimal configuration adopts a number of cores about twice the linear size of the mesh. However, the usage of a number of cores four times N_{1D} is still acceptable, the efficiency improving with the size of the problem. Therefore, the 2048^3 configuration is the most effective, since it can use 2048 Jugene nodes in both DUAL and VN mode, hence exploiting half or even all the available 8192 cores, avoiding wasting of computing resources and achieving good performances. Unfortunately, in the time-frame and with the CPU resources available for the present project, we could not perform tests on this, or larger, configurations. We have estimated the CPU time needed by the 2048^3 and 4096^3 cases using both a linear and a cubic extrapolation from the smaller tests (see the table below). The former provides a sort of lower limit in the CPU time per time step per core, the latter is instead an upper bound. According to these estimates, in order to complete a run of about 1000 timesteps for a 2048^3 mesh, between 215000 and 250000 CPU hours are required, corresponding to a wall clock time between 52 and 60 hours on 4096 cores (DUAL mode). For the 4096^3 mesh only the 16384 nodes SMP mode configuration can be used. The usage of a larger number of cores, in fact, would lead to to a strong deterioration performances by ENZO. In such configuration, we estimate that between 2000000 and 3400000 CPU hours are necessary to complete a simulation. Therefore, a single run would take between 123 and 207 wall clock hours to finish.

N_{1D}	Memory (GB)	Min. Nodes	Opt. Core	T(Opt. Core)	T(2xOpt. Core)
512	60	32	1024	21	25
1024	480	256	2048	60	70
1500	1510	1024	3000	120	130
2048	3900	2048	4096	189-217	199-220
4096	31000	16384	16384	NA	457-765

In the table, Memory requirements, minimum number of Jugene nodes, optimal number of cores, and wall clock time versus problem size. The minimum number of nodes is estimated such that enough memory is available for the corresponding problem and the smallest possible partition of the Blue Gene/P is allocated; the optimal number of cores is estimated considering the memory requirements and the best Enzo performance. Columns 5 and 6 show the wall clock time (in seconds) to complete a single time-step using different numbers of cores. For the 2048^3 and 4096^3 cases the values are linearly (lower value) and cubic (higher value) extrapolated from smaller cases.

Benefits for the possible PRACE project proposal

In this work, the ENZO code has been successfully ported on the Jugene architecture. Its performances and its computational requirements have been analyzed and optimized. According to this analysis, the AMR configuration has been ruled out, being too memory demanding, due to the presence of the HT data structure that is replicated on each processor and that tends to grow as the evolution of the simulated system proceeds. The UNIGRID configuration has been accurately benchmarked and the best possible configuration for the simulations of interest identified. In particular, the proper number of nodes for computational meshes of different size was set, in order to ensure that both enough memory is available and the best performance is achieved. It has been found that such optimal number of cores is twice the linear size of the mesh. In order to generate initial conditions for the production-size runs the INITS code (initially sequential) has been parallelized. Finally we have identified the most suitable configurations to fulfil our scientific requirements matching the available HPC resources and estimating the corresponding necessary CPU time. This corresponds to a mesh size of 2048^3 cells, on 4096 cores and requires a CPU time of the order of 250000 hours. The described work provides the necessary setup and information to submit a proposal in one of the next call for projects of the PRACE project.

6.6 Semi-dilute polymer systems in shear flow – a particle based hydrodynamics approach

1.General information

Project name	Semi-dilute polymer systems in shear flow - a particle based hydrodynamics approach
Proposal reference number	PRA3IC
Scientific field of the project	Molecular dynamics
Project leader	Name: Godehard Sutmann, Alexander Schnurpfeil Affiliation: Forschungszentrum Jülich / JSC Contact information: a.schnurpfeil@fz-juelich.de
PRACE staff involved (please give the information for all involved PRACE persons)	Name: Annika Schiller, Florian Janetzko, Stefanie Meier Affiliation: Forschungszentrum Jülich / JSC Amount of work in person months: 6 planned, 6 used
Computer system(s) employed	JUGENE, JUROPA

2. Project information

Scientific goals of project

A characteristic feature of soft matter systems is that a macromolecular component of nano- to micrometer size is dispersed in a solvent of much smaller molecules. The meso-scopic length scale of the dispersed component implies that both crystalline and fluid phases are characterized by long structural relaxation times. Soft matter systems have therefore interesting dynamical properties, because the time scale of an external perturbation can easily become comparable with the intrinsic relaxation time of the dispersed macromolecules. We employ here the multi-particle collision dynamics (MPC) technique, also called stochastic rotation dynamics (T. Ihle and D. M. Kroll, Phys. Rev. E 63, 020201(R) (2001)).

This particle based hydrodynamics method consists of alternating streaming and collision steps. In the streaming step, particles move ballistically. In the collision step, particles are sorted into the cells of a simple cubic (or square) lattice. All particles in a cell collide by a rotation of their velocities relative to the center-of-mass velocity around a random axis (A. Malevanets and R. Kapral, J. Chem. Phys. 110, 8605 (1999)). A random shift of the cell lattice is performed before each collision step in order to restore Galilean invariance (T. Ihle and D. M. Kroll, Phys. Rev. E 63, 020201(R) (2001)). This method has been applied very successfully to study the hydrodynamic behavior of many complex fluids.

Computational approach

The program MP2C is implemented in module oriented Fortran 90. Message passing between processors is realized with the MPI standard. The parallel algorithm is based on a 3-dimensional domain decomposition approach, where particles are sorted onto processors according to their spatial coordinates. The present version of the program is implemented on a 2n subdivision of processors. At present, no load balancing strategy is implemented, so that volumes and boundaries of spatial domains are not altered during a simulation. This has the administrative advantage that the neighbour processor IDs do not change and the local communication pattern can be determined in the setup phase of the program. For a system in d dimensions with applied periodic boundary conditions the number of next neighbours of a given processor is therefore (3d-1), which may be different for different kinds of boundary conditions, e.g. close to surfaces, communication is performed only within a half-space. A requirement for a future release of the program is to include load balancing in order to have a sufficient scaling behaviour for inhomogeneous systems.

The algorithm to propagate fluid-particles is based on a combined streaming and collision step. In a first step particles are propagated ballistically according to their current position and velocity. In a second step, fluid particles are sorted into collision cells, where the relative velocities with respect to the center-of-mass velocity of the cell is rotated around a randomly chosen axis with a given rotation angle. The combined parameters of rotation angle, density of particles and applied time step determine the fluid properties, like viscosity or diffusivity. Given the fluid parameters, these properties can be calculated analytically for reference. In a limiting case, it can be shown that the method is mapped to the Navier-Stokes equations. Through coupling of the fluid to solvated particles, hydrodynamic modes and interactions can naturally be included into, e.g. polymer of macromolecular dynamics simulations. In this case, solvated molecules are simulated via molecular dynamics and additionally coupled to the fluid. Coupling is performed by including the solvated particles into the stochastic collision step.

I/O is performed via two methods at present:

1. for small sets of data ASCII or binary data are written out into one file, where data are collected on a master processor. This strategy is only used for infrequent I/O intervals and moderate number of processors, since it does not scale.
2. For large data sets, the parallel library SIONlib (developed at JSC, www.fz-juelich. de/sionlib, (i) W. Frings and F. Wolf and V. Petkov, SIONlib: Scalable parallel I/O for tasklocal files, Proceedings of Supercomputing 2009, Portland, Oregon. (ii) J. Freche, W. Frings and G. Sutmann, High Throughput ParallelI/ O using SIONlib for Mesoscopic Particles Dynamics Simulations on Massively Parallel Computers, Proc. of Intern. Conf. ParCo 2009, (IOS Press, Amsterdam, 2010), p.423)

Performance goals

MP2C‘s scaling behaviour is deteriorating when benchmark runs are performed on beyond 262K cores. This effect occurs most likely due to communication overhead.
In this context core mapping might be an issue on JUGENE.
Besides that, reducing necessary communication certainly helps to overcome the problem.

3 Results

Summary of results obtained

Installation and Configuration:
MP2C has been integrated in JuBE and test runs were performed on JUGENE and JUROPA.

Software development:
OpenMP was implemented in the MPC part of OpenMP. Corresponding test runs show improved scaling behaviour of the hybrid code version for moderate particle numbers. In particular, it turned out that the hybridization of the „Cell Filling Step“ considerably increases the performance of the code. Among other tasks, this step controls the particle cell exchange. Fig. 1 presents the performance of the particle cell exchange as one main part of this step with different particle numbers calculated on JUROPA. The runs were executed on 1 up to 256 nodes. For the pure MPI run, it means that a maximal number of 2048 cores were spawned. For the hybrid version, 4 MPI tasks were launched per node each using 2 threads. Apart from moderate node sizes, the figure clearly shows that for larger particle numbers the MPI version of the corresponding routines (red line) show a better performance compared to the corresponding hybrid version (cyan line) with comparable parameters. The situation changes when we go over to smaller amounts of particles. At first, the pure MPI version shows a better scaling compared to the related run of the hybrid code, as long as we choose a moderate size for the number of tasks (blue line). If we further increase the number of MPI tasks the performance decreases beyond 256 cores, most likely due to a communication overhead. Finally, a cross over can be observed, the hybrid version shows a slightly better performance on up to about 1024 cores. This well known behaviour of MP2C was also observed on JUGENE in former simulations, where the communication overhead deteriorates the scaling properties in runs beyond 262K cores. On JUROPA we reconstruct these conditions by reducing the number of particles. One may expect that the hybrid version of MP2C more or less compensates the communication demands to a certain degree. This is particularly true if the particle exchange is being implemented in future versions of MP2C.

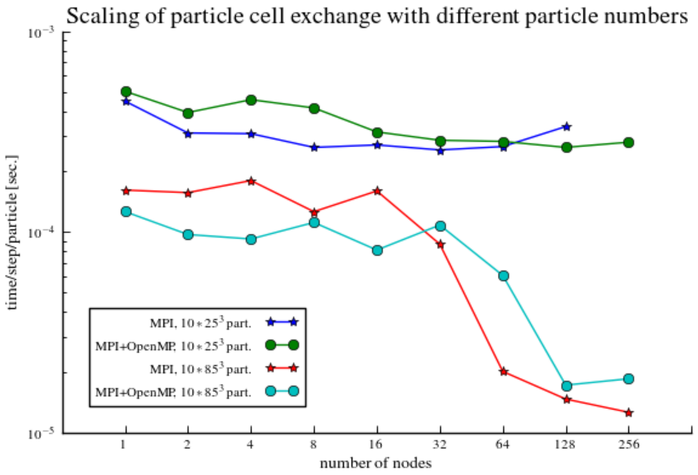

Figure 1: Performance of the cell exchange for a different number of particles.

Benefits for the possible PRACE project proposal

This is a PRACE WP7.1 internal project. Therefore, application for the PRACE Regular Call is not planned at this time.

7 Conclusions

During its first year, task 7.1 has worked towards its objectives. The task participated in the user survey together with task 7.4 focusing on application-enabling aspects. The user survey revealed a clear need for PRACE application-enabling work, however, there is also room for improving interest in PRACE collaboration.

Task 7.1 organized an internal call in order to start application enabling work before the actual preparatory access projects could be started. Six applications were selected through an internal review and voting process, the results of the projects are encouraging PRACE experts got experience about Tier-0 systems and about typical scalability bottlenecks. Task 7.1 was able to help also users by enabling new applications to be used in Tier-0 systems. For at least two cases, the internal projects will enable them to potentially apply successfully in future regular calls for project access.

In early 2011, the first proposals in the regular PRACE Type C preparatory access call have been approved by PRACE, so that task 7.1 efforts could be applied for work on these applications. Task 7.1 itself has been involved in the review of Type C proposals through a review group which was formed from internal PRACE experts. The evaluation process as well as the process for assigning optimization work to PRACE partners has been defined. Throughout the following evaluation rounds these process will be refined if needed.

During the next half of the project, results about the first Preparatory Access projects will be obtained. Task 7.1 will also start work on new type C Preparatory Access projects. As the new Tier-0 system, Hermit, becomes available, a new subtask for optimization work on Hermit will be formed. Task 7.1 will also contribute in preparing a new call for projects.

In summary, task 7.1 has made good progress towards its objectives. In particular, it has been demonstrated that the approach of focussed short-term support activities by PRACE experts is successful. It has the envisaged effect of enabling new applications for Tier-0 usage or significantly enhances their scalability.