SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

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

CAPACITIES

PRACE-2IP

PRACE Second Implementation Phase Project

Grant Agreement Number: RI-283493

D7.1
Scaling Support for Preparatory Access Users
Final

Version:	1.0
Authors:	Stefanie Janetzko, GCS-FZJ, Alexander Schnurpfeil, GCS-FZJ
Date:	21.08.2013

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: RI-283493**	
	Project Title: PRACE Second Implementation Phase Project	
	Project Web Site:http://www.prace-project.eu	
	Deliverable ID: D7.1	
	Deliverable Nature: Report	
	Deliverable Level: PU *	**Contractual Date of Delivery:** 31 / August / 2013
		Actual Date of Delivery: 31 / August / 2013
	EC Project Officer: 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:** Scaling Support for Preparatory Access Users	
	ID: D7.1	
	Version: 1.0	**Status:** ***Final***
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2010	
	File(s): D7.1.docx.docx	
Authorship	**Written by:**	Stefanie Janetzko, GCS-FZJ, Alexander Schnurpfeil, GCS-FZJ
	Contributors:	Pierre Kestener, CEA, Franck Houssen, CEA, Carlo Cavazzoni, CINECA, Francesco Salvadore, CINECA, Nicole Audiffren, CINES, Sami Saarinen, CSC, Martti Louhivuori, CSC, Jussi Enkovaara, CSC, Iain Bethune, EPCC, Joerg Hertzer, GCS-HLRS, Volker Weinberg, GCS-LRZ, Anupam Karmakar, GCS-LRZ, Paschalis Korosoglou, GRNET, Alexandra Charalampidou, GRNET, Pavlos Daoglou, GRNET, Maciej Szpindler, ICM, Maciej Cytowski, ICM, Isabelle Dupays, IDRIS, Margareta Boiarciuc, IDRIS, Vladimir Slavnic, IPB, Petar Jovanovic, IPB, Dusan Stankovic, IPB, Valentin Pavlov, NCSA, Vegard Eide, NTNU, Luis Fazendeiro, SNIC-Chalmers, Michael Schliephake, SNIC-KTH, Jonathan Vincent, SNIC-KTH, Soon-Heum Ko, SNIC-LiU, Joachim Hein, SNIC-LU, Mikael Rännar, SNIC-UmU, Petros Souvatzis, SNIC-UU, Jeroen Engelberts, SURFsara

	Reviewed by:	Stelios Erotokritou, CaSToRC Dietmar Erwin, GCS-FZJ
	Approved by:	MB/TB

Document Status Sheet

Version	**Date**	**Status**	**Comments**
0.1	11/July/2013	Draft	Structure and chapters 1 and 2
0.2	29/July/2013	Draft	Project reports included, Summary
0.3	05/August/2013	Draft	After task-internal review
0.4	20/August/2013	Draft	After project-internal review
1.0	21/August/2013	Final version	

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, Preparatory Access

Disclaimer

This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement n°RI-283493. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements. Please note that even though all participants to the Project are members of PRACE AISBL, this deliverable has not been approved by the Council of PRACE AISBL and therefore does not emanate from it nor should it be considered to reflect PRACE AISBL's individual opinion.

Copyright notices

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[1] http://www.prace-ri.eu/Call-Announcements
[2] http://www.prace-ri.eu
[3] http://www.quantum-espresso.org
[4] http://www.cp2k.org
[5] http://www.prace-project.eu/IMG/pdf/wp51_high_performance_mp2_for_condensed_phase_simulations.pdf
[6] http://en.wikipedia.org/wiki/HIRLAM
[7] http://empslocal.ex.ac.uk/people/staff/vnb262/software/BeatBox/
[8] http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview
[9] https://wiki.fysik.dtu.dk/gpaw/
[10] http://www.prace-ri.eu/IMG/pdf/wp68.pdf
[11] http://thetis.enscbp.fr/
[12] http://www.prace-ri.eu/IMG/pdf/wp69.pdf
[13] http://www.prace-ri.eu/IMG/pdf/wp72.pdf
[14] http://dp-code.org/
[15] http://www.prace-ri.eu/IMG/pdf/wp67.pdf
[16] http://www.oact.inaf.it/fly/
[17] http://www.prace-ri.eu/IMG/pdf/wp70.pdf
[18] http://www.clustal.org/
[19] http://www.prace-ri.eu/IMG/pdf/wp71.pdf
[20] http://www.libgeodecomp.org/
[21] http://www.tcm.phy.cam.ac.uk/~mdt26/casino2_introduction.html
[22] http://www.prace-ri.eu/IMG/pdf/wp74.pdf
[23] http://bmi.osu.edu/~umit/software.html
[24] http://www.epcc.ed.ac.uk/projects-portfolio/cardiac-arrhythmia-research-package-carp
[25] http://www.openfoam.com
[26] http://www.tddft.org/programs/octopus/wiki/index.php/Main_Page
[27] http://www.gromacs.org

List of Acronyms and Abbreviations

AISBL	Association International Sans But Lucratif (legal form of the PRACE-RI)
AMD	Advanced Micro Devices, Inc.
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BLAS	Basic Linear Algebra Subprograms
BSC	BarcelonaSupercomputing Center (Spain)
CEA	Commissariat à l'EnergieAtomique (represented in PRACE by GENCI, France)
CHALMERS	Chalmers University of Technology (Sweden)
CaSToRC	Computation-based Science and Technology Research Center
CINECA	Consorzio Interuniversitario, the largestItaliancomputing 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)
CUDA	Compute Unified Device Architecture
DECI	Distributed European Computing Initiative
DGEMM	Double precision General Matrix Multiply

DP	Double Precision, usually 64-bit floating point numbers
EC	European Community
EPCC	Edinburg Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
FFT	Fast Fourier Transform
FPU	Floating-Point Unit
FZJ	Forschungszentrum Jülich (Germany)
GB	Giga (= 2^{30} ~ 10^{9}) Bytes (= 8 bits), also GByte
GB/s	Giga (= 10^{9}) Bytes (= 8 bits) per second, also GByte/s
GCS	Gauss Centre for Supercomputing (Germany)
GÉANT	Collaboration between National Research and Education Networks to build a multi-gigabit pan-European network, managed by DANTE. GÉANT2 is the follow-up as of 2004.
GENCI	Grand Equipement National de CalculIntensif (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
GPGPU	General Purpose GPU
GPU	Graphic Processing Unit
GRNET	Greek Research and Technology Network (Greek)
HDF5	Hierarchical Data Format
HLRS	Höchstleistungsrechenzentrum Stuttgart (Germany)
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
IBM	Formerly known as International Business Machines
ICM	Interdyscyplinarne centrum modelowania matematycznego I komputerowego (Poland)
IDRIS	Institut du Développement et des Ressources en Informatique Scientifique (represented in PRACE by GENCI, France)
I/O	Input/Output
IP	Implementation Project
IPB	Institute of Physics, Belgrad (Serbia)
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)
LiU	Linköping University (Sweden)
LQCD	Lattice QCD
LRZ	Leibniz Supercomputing Centre (Garching, Germany)
LU	Lund University (Sweden)
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
MHz	Mega (= 10^{6}) Hertz, frequency =10^{6} periods or clock cycles per second
MKL	Math Kernel Library (Intel)
MPI	Message Passing Interface
NCF	Netherlands Computing Facilities (Netherlands)
NTNU	Norges teknisk-naturvitenskapelige universitet (Norway)
OpenMP	Open Multi-Processing
OS	Operating System

PA	Preparatory Access
PA C	Preparatory Access Type C
PETSc	Portable, Extensible Toolkit for Scientific Computation
PGI	Portland Group, Inc.
PI	Principal Investigator
PM	Person month
POSIX	Portable OS Interface for UNIX
PPM	Portable pixmap format
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PSNC	Poznan Supercomputing and Networking Centre (Poland)
QCD	Quantum Chromodynamics
RAM	Random Access Memory
RI	Research Infrastructure
ROMIO	High-Performance, Portable MPI-IO Implementation
SHMEM	Share Memory access library (Cray)
SMP	Symmetric MultiProcessing
SNIC	Swedish National Infrastructure for Computing (Sweden)
STFC	Science and Technology Facilities Council (represented in PRACE by EPSRC, United Kingdom)
SURFsara	Dutch national High Performance Computing & e-Science Support Center
TB	Tera (= 240 ~ 1012) Bytes (= 8 bits), also TByte
TDDFT	Time-Dependent Density Functional Theory
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
UmU	Umea universitet (Sweden)
UU	Uppsala University (Sweden)
VTK	The Visualization Toolkit

Executive Summary

PRACE-2IP Task 7.1 “Petascaling and Optimization Support for Preparatory Access projects” is responsible for providing support to European researchers to enable and optimize their applications on Tier-0 systems. The users may apply for this service via the Preparatory Access Call Type C. The work includes evaluating proposals for Type C in the Preparatory Access Call, assigning PRACE experts to the projects and performing the optimization support work.

In PRACE-2IP five Preparatory Access Calls have been carried out and 24 Type C projects have been supported. The codes have been optimized together with the involved Principal Investigators. As a result, five of these projects have already successfully applied for regular access to the Tier-0 systems with the optimized code; more projects are planning to apply for regular access in the next calls also. The above achievements demonstrate that the task was able to strengthen the ability of researchers to use the provided Tier-0 systems.

This deliverable reports on the work done and achievements of the Task 7.1. It includes statistics about the calls in PRACE-2IP as well as a description of the call organization, i.e. the Type C review process and the assignment of support experts to projects. Additionally, project monitoring and dissemination work is outlined. Finally, the performed optimization work itself is the key aspect of the task and is described in detail in this deliverable and documented in eight accompanying white papers.

1 Introduction

PRACE offers a wide range of different Tier-0 architectures to the scientific community as well as to industrial projects. The efficient usage of such systems places high demands on the used software packages and in many cases advanced optimization work has to be applied to the codes to make best use of the provided supercomputers. Consequently, there is a high need for preparatory access giving the opportunity to analyze and optimize codes prior to applying for resources in the regular calls [1]. The complexity of supercomputers requires a high level of experience and advanced knowledge of different kinds of concepts regarding programming techniques, parallelization strategies, etc. Such demands oftentimes cannot be covered by the applicants themselves but special assistance of supercomputing experts is needed. PRACE offers such a service through the Preparatory Access Call and specifically via Preparatory Access Type C (PA C) for optimization work with support by PRACE experts.

Task 7.1 “Petascaling and Optimization Support for Preparatory Access projects” is responsible for the evaluation of PA C proposals and the assignment of PRACE experts to these proposals. Furthermore the support work itself is performed and monitored within this task.

The structure of this document is as follows: Chapter 2 gives a more detailed description of the preparatory access calls and shows statistics on the usage of PA Type C in PRACE-2IP. The review process, the assignment of PRACE experts to the projects, and the monitoring of the support work is explained.

In chapter 3 the optimization work on the PA Type C projects handled in PRACE-2IP is described in detail. In section 3.1 and 3.2 full reports are given. As of cut-off March 2012 the task focuses on white papers to disseminate the results as early and as widely a possible. Therefore in section 3.3 and 3.4 abstracts of the white papers are reproduced. Section 3.5 and 3.6 describe work in progress.

The summary in chapter 4 shows statistics about the projects and outlines the results reached in Task 7.1 of PRACE-2IP.

2 Preparatory Access Calls

Access to PRACE Tier-0 systems is managed through PRACE regular calls which are issued twice a year. To apply for Tier-0 resource the application must meet technical criteria concerning scaling capability, memory demands, and runtime set up. There are many important scientific and industrial applications which do not meet these criteria. To support the researchers PRACE offers the opportunity to test and optimize their applications on the envisaged Tier-0 system prior to applying for a regular production project. This is the purpose of the Preparatory Access Call. The PA Call is a continuous call with a cut-off every three months for evaluation of the proposals. Therefore quarterly new projects obtain access for preparatory purposes to PRACE Tier-0 systems. It is possible to choose between three different types of access:

- Type A is meant for code scalability tests to include the outcome in the proposal for a future PRACE Regular Call. Users receive a limited number of core hours depending on the system to which they got access; the allocation period is two months.
- Type B is intended for code development and optimization by the user. Users get also a small number of core hours; the allocation period is 6 months.

- Type C is also designed for code development and optimization with the core hours and the allocation period being the same as for Type B. The important difference is that Type C projects get special support by PRACE to address the optimization requests. In addition to access to the Tier-0 systems the applicants apply for 1 to 6 PMs of supporting work to be performed by PRACE support experts.

PA Type C is the type that PRACE task 7.1 is dealing with as described in chapter 1.

All Tier-0 systems are available for PA. Currently these are the following systems:

- CURIE, BULL Bullx cluster at GENCI-CEA, France (thin, fat, and hybrid nodes are available)
- FERMI, IBM Blue Gene/Q at CINECA, Italy
- HERMIT, CRAY XE6 at GCS-HLRS, Germany
- JUQUEEN, IBM Blue Gene/Q at GCS-JSC, Germany
- MareNostrum, IBM System X iDataplex at BSC, Spain
- SuperMUC, IBM System X iDataplex at GCS-LRZ, Germany

One further system was a hosting system for some of the PA projects at the beginning of PRACE-2IP; it has been replaced in 2012 by JUQUEEN:

- JUGENE, IBM Blue Gene/P at GCS-JSC, Germany (predecessor of JUQUEEN)

2.1 Cut-off Statistics

In PRACE-2IP five cut-offs for PA took place. Here, suitable projects are identified by a technical review process and experts from the PRACE project are assigned to these projects to support the optimization work.

This section gives an overview on the outcome of the projects at each cut-off. The five cut-offs took place on the following dates: March 1st, 2012, June 1st, 2012, September 1st, 2012, December 1st, 2012, and March 1st, 2013.

Figure 1 presents the number of proposals and the number of projects which have been accepted for each cut-off.

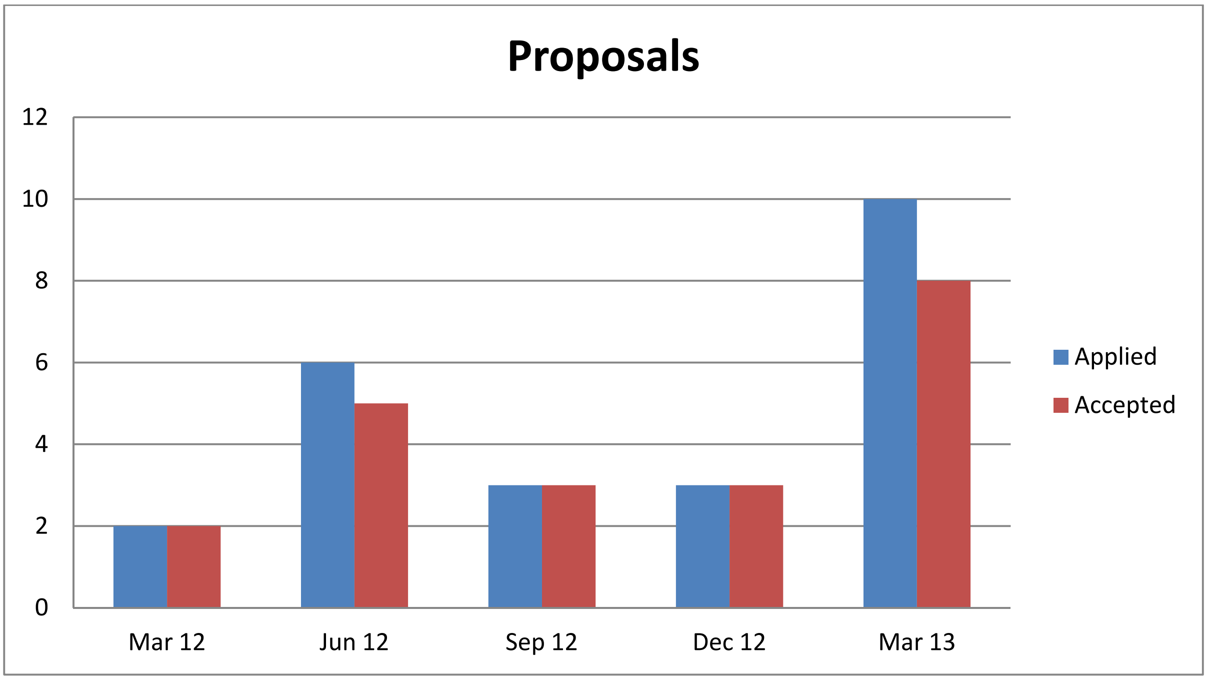

Figure 1: Number of submitted and accepted proposals for preparatory access per cut-off

It can be seen that only a small number of proposals had to be rejected, one proposal in cut-off June 2012 and two proposals in cut-off March 2013.

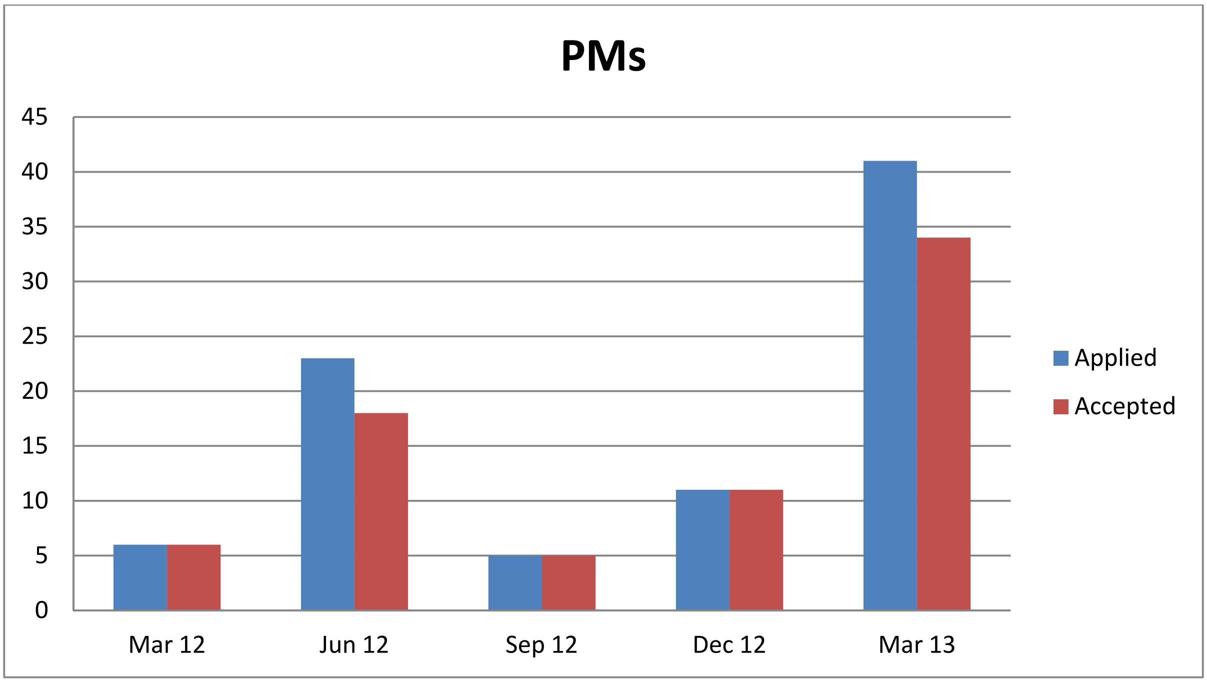

Figure 2: Number of requested and accepted person-months for support in preparatory access proposals

In Figure 2 the number of requested and accepted person-months (PMs) is shown. On average a PA-C proposal receives 3-4 PMs.

It is also interesting to see that researchers from a broad range of scientific fields could be reached by the PA C call; this is shown in Figure 3.

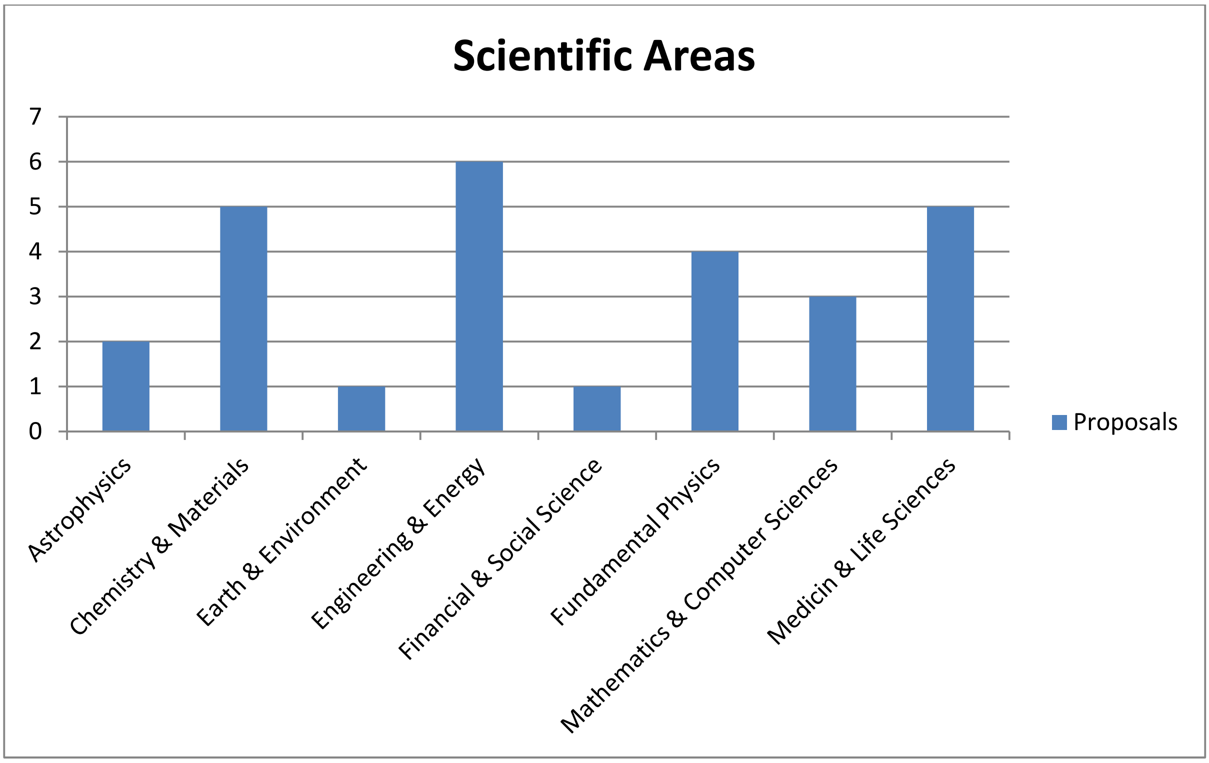

Figure 3: Number of proposals per scientific field

2.2 The Review Process

The management of the review procedure, the assignment of PRACE collaborators and the supervision of the PA C projects are handled by task 7.1. In this section the review process for the preparatory access proposals of Type C is explained.

All preparatory access proposals undergo a technical review performed by technical staff of the hosting sites to ensure that the underlying codes are principally able to run on the requested system. In parallel, all PA C projects are additionally reviewed by work package 7 in order to assess their optimization requests. Each proposal is assigned to two WP7 reviewers. The review is performed by PRACE partners who all have a strong background in supercomputing. Currently a list of 37 experts is maintained and the task leader has the responsibility to contact them to launch the review process. As the procedure of reviewing proposals and establishing the collaboration of submitted projects and PRACE experts takes place four times a year it is necessary to keep the review process timely and efficient. A close collaboration between AISBL, T7.1 and the hosting sites is important in this context. The review process for technical and WP7 review is limited to two weeks. In close collaboration with AISBL and the hosting sites the whole procedure from PA cut-off to project start on PRACE supercomputing systems is carried out in less than six weeks.

Based on the proposals the Type C reviewers need to focus on the following aspects:

- Does the project require support for achieving production runs in the chosen architecture?
- Are the performance problems and their underlying reasons well understood by the applicant?
- Is the amount of support requested reasonable for the proposed goals?
- 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)?
- Will there be limitations in disseminating the results achieved during the project?

Additionally the T7.1 task leader decides on the question whether the level and type of support requested is still available from PRACE. Finally the recommendation from WP7 to accept or reject the proposal is made.

Based on the provided information from the reviewers the Board of Directors has the final decision on whether proposals are approved or rejected. The outcome is communicated to the applicant through AISBL. Approved proposals receive the contact data of the assigned PRACE collaborators, refused projects are provided with further advice on how to address the rejection reasons.

2.3 Assigning of PRACE Collaborators

To ensure the success of the projects it is essential to assign suitable experts from the PRACE project. This means based on the described optimization issues and support requests from the proposal experts are chosen who are most familiar with the subject.

This is done in two steps: First, summaries of the proposals describing the main optimization issues are distributed via corresponding mailing list. Here, personal data are explicitly removed from the reports to keep the anonymity of the applicants. Interested experts can get in touch with the task leader offering to work on one or more projects.

In case the response is not sufficient to cover the support requests of the PA C projects, the task leader contacts 7.1 experts directly and asks them to contribute. In order to identify suitable collaborators a list of experts is maintained along with their special fields of expertise.

There is one exception to the procedure in the case when a proposal has a close connection to a PRACE site which e.g. already worked on the code: In this case this site is asked first if they are able to continue this collaboration in the context of the PA C project.

This procedure has proved to be extremely successful; no proposal had to be refused due to lack of suitable support so far.

To be able to manage the whole review process with six weeks the assignment of PRACE experts takes place concurrently with the review process. This has shown to be a suitable approach. The overhead resulting in the assignment of projects that are rejected in the end is negligible.

After the review process described in section 2.2 is finished the support experts are introduced to the PIs and can start the work on the projects. The role of the PRACE collaborator includes the following tasks:

- preparing a detailed work plan together with the applicant,
- participating in the optimization work,
- reporting the status report in the phone conference every second month,
- participating in the writing of the final report together with the PI (the PI has the main responsibility for this report), due at project end and requested by the PRACE office,
- writing a white paper containing the results which is published on the PRACE web site.

2.4 Monitoring of Projects

Another task is the supervision of the Type C projects. This turns out to be a challenge as the projects' lifetimes (six months) and the intervals of the cut-offs (3 months) differ. This means that projects do not necessarily start and end at the same time but overlaps exist, i.e. at each point in time different projects might be in different states. Therefore, regular two-monthly phone conferences take place in task 7.1 to discuss the status of running projects, to give advice on how to proceed with new projects and to manage the finalization and reporting of finished projects.

The conference addresses all PRACE collaborators who are involved in these projects. All the project relevant information is maintained on a PRACE wiki page which is available to all PRACE collaborators.

Additionally the T7.1 task leader is available to address urgent problems and additional phone conferences are held in such cases.

Bi-yearly a WP7 Face-to-Face meeting is scheduled. This meeting gives all involved collaborators the opportunity to discuss the status of the projects and to exchange their experiences.

2.5 Hand-over between PRACE IP Projects

The support for Preparatory Access Type C projects has been and is part of all PRACE projects (PRACE-1IP, -2IP, -3IP). For the hand-over between the projects the tasks decided on the following exact dates.

The hand-over between PRACE-1IP and PRACE-2IP for running projects took place in June, 2012. The December 2011 cut-off for PA C projects was still under the responsibility of

PRACE-1IP. However, as these projects finished after the end of PRACE-1IP, the responsibility for the enabling work of these projects was transferred from PRACE-1IP to 2IP at June, 13, 2012. The enabling work was finalized under PRACE-2IP and the projects are reported in this deliverable. The March 2012-cutoff and the accepted projects were already handled fully in PRACE-2IP.

The hand-over between PRACE-2IP and PRACE-3IP will take place right at the end of PRACE-2IP, August 31, 2013. The cut-off which took place in March 2013 was still under the responsibility of T7.1 in PRACE-2IP. Because the approved projects will run until October 2013, i.e. beyond PRACE-2IP, PRACE-3IP-experts also needed to be assigned to these projects. Currently the optimization work is still being done in PRACE-2IP, but 3IP will take over from September 1 on. For these projects an interim report can be found in this deliverable describing the work done in PRACE-2IP. The final results will be described in the PRACE-3IP deliverable for PRACE-3IP-T7.1, the task which will take over responsibility for the PA support. PRACE 3IP Task 7.1 also already took over responsibility for the upcoming cut-offs including the cut-off June 2013.

2.6 Dissemination

The task uses different channels for dissemination. For each PA call the PRACE sites are asked to distribute an email to their users to advertise preparatory access and especially the possibility of dedicated support via PA C. A template for this email was created in T7.1.

In the PRACE annual report for 2012 Preparatory Access Type C was highlighted as unique opportunity to improve code performance and for getting ready for production usage on PRACE Tier-0 resources.

Also each successfully performed PA C project should be made known to the public and therefore the PRACE collaborators are asked to write a white paper about the optimization work carried out. These white papers are published on the PRACE web site [2] and are also referenced in this deliverable.

3 Optimization Work

This chapter describes the optimization work carried out on the preparatory projects of Type C. The strategy to report on this work is outlined here:

For the first cut-offs (sections 3.1 and 3.2) for each project a report written by the assigned PRACE experts is provided. This report includes the project objectives, the optimization work and results as well as a conclusion (this is referred to as the collaborator report).

During PRACE-2IP run time it turned out that the writing of different reports (at that time three: final report by the PI for the PRACE office (not public), collaborator report and white paper for T7.1) was confusing for PRACE collaborators and also for involved PIs. It has been agreed from the beginning that the PIs should only write the final PI report. For the PRACE collaborators it has now been agreed that they focus on publishing white papers after the end of the PA project and that an additional collaborator report is not needed anymore. By placing more emphasis on white papers we continue to strengthen the outreach of PA C. Therefore for these projects the basic general information as well as the abstract of the white paper can be found in this deliverable (sections 3.3 and 3.4). Additionally the white papers on the PRACE-RI web site are referenced.

In section 3.5 projects with an end date after the creation of this deliverable are described, so the white papers are not available up to now. For these projects the same structure as for the collaborator report is used to describe the work in the deliverable. The projects in section 3.6 are running beyond PRACE-2IP and the work performed in PRACE-2IP is described also similar to a collaborator report with slightly different sections: project objectives, optimization work and preliminary results as well as an outlook.

It should be noted that all project reports provided in this deliverable have been written by the PRACE T7.1 support experts who collaborated on the projects to perform the enabling and optimization tasks.

3.1 Cut-off December 2011

As described in section 2.5 the projects from this cut-off started in PRACE-1IP, but the final work has been taken over by PRACE-2IP after the end of PRACE-1IP. The collaborator reports from such projects can be found in this section.

3.1.1 *Increasing the QUANTUM ESPRESSO capabilities: towards the DFT simulation of realistic nanoparticles*

Project leader:	Arrigo Calzolari, CNR-NANO Istituto Nanoscienza
PRACE expert:	Carlo Cavazzoni, CINECA
PRACE facility:	CURIE hybrid nodes
Research field:	Material Science
Application code:	Quantum ESPRESSO[3]
PA number:	2010PA0699

Project objectives

The goal of the project was to find the most effective way of simulating large metal nano particles exploiting the computational power of Tier-0 system. In particular the research group was interested to find out the amount of memory required per core and use this to compare the performance and the scalability of fat nodes, thin nodes and GPU nodes. All this was done with the goal of being able to apply for the regular PRACE Tier-0 call.

The subjects of the simulations were the following:

- Prototype material for optoelectronic applications
- Easy-growth nanoparticles through chemical processes (colloidal synthesis)

The numerical challenge is the high memory requirements per MPI task due to high electrons-to-atoms ratio in pseudo-potential calculations.

Optimization work and results

To address the problem related to the requirement of memory per tasks, we asked the PI to set up a set of similar input datasets of increasing size (number of atoms) such that we could run a series of tests to evaluate the relation between the amount of memory required and the system size. In Table 1 we report the results of the tests as measured on CURIE:

# atoms	# tasks	GByte/task	Mem/size
75	64	0.9	1
489	64	2.2	1.37
922	64	6.7	2.22
1214	64	11.0	2.77

Table 1: Test-results on CURIE - relation between amount of memory and system size

(First column: number of atoms, second column number of tasks, third column: number of Gigabyte of memory per task, and fourth column: normalized ration between memory per task and system size (number of atoms)).

Table 1 shows that the memory per task as a function of the system size does not scale linearly, and that the hybrid MPI/OpenMP code is required to fit into the memory of each node. This result can also be used to evaluate the maximum system size that is possible to simulate with Quantum ESPRESSO using today’s PRACE Tier-0 systems.

To perform the scalability tests and the comparison between the performance of the hybrid nodes and the fat nodes we choose to use 8 OpenMP threads for each task.

We also reviewed and optimized the OpenMP implementation of the code in order to limit as much as possible the memory footprint of the application.

For the scalability test we did not take the largest system (1214 atoms) but the one with 922 atoms, so that we could work on improving the scalability itself without using up too many resources (quite obviously improving the scalability for the smaller system sizes leads to improving the scalability for the larger sizes too).

Below we report the results of the scalability tests. The details of the simulated system are as follows: Cadmium selenide nanocrystal (CdSe dot)

- # atoms = 922 (466 Cd +456 Se)
- # electrons = 8328 (4164 bands)
- Exchange and Correlation functional: PBE
- K-Points: Gamma only
- Pseudopotential: USPP
- Ecut = 25 Ry, Ecutrho = 200 Ry
- # projectors = 3x466 + 2x456 = 2310
- # g-vec tot (smooth) = 11.092.281

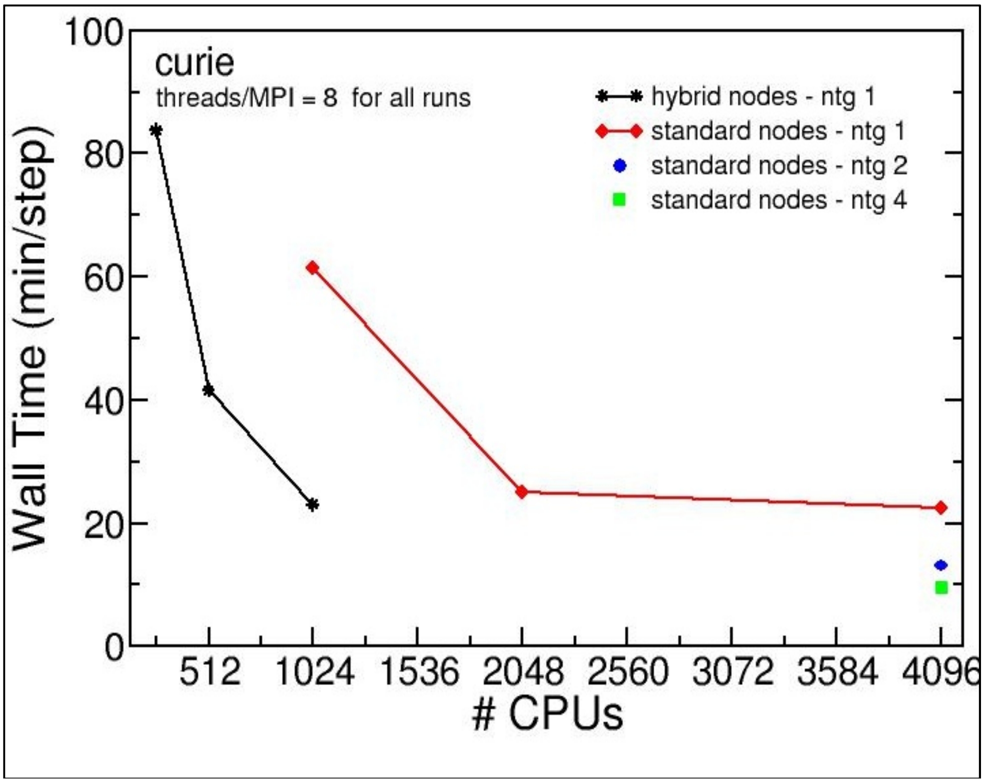

Figure 4: QuantumESPRESSo scaling (note that here CPUs means COREs)

In Figure 4 we show two scalability curves for standard nodes (red line) and hybrid nodes (black line), and two values obtained using two different data distributions for the parallel 3D FFT (blu circle and green square). In Table 2 the values are also listed. More precisely the label “ntg” means number of task groups. It is important to remark that task groups have little effect on the performances if they are used with a small number of tasks (they could even do more harm than good).

The test on the hybrid node has been performed with the code enabled for CUDA. Moreover this PA has been used to better tune the CUDA parallelization, especially concerning the distribution of work between tasks/threads and CUDA devices. Our set up allows to assign one GPU per task and to distribute the work between the GPU and the 8 OpenMP threads.

	# CPUs	Total time (min)	Time per step (min)
Hybrid ntg1	256 (32x8) 512 (64x8) 1204 (128x8)	335 196 92	83.75 41.50 23.00
Standard ntg1	1024 (128x8) 2048 (256x8) 4096 (512x8)	246 100 90	61.50 25.00 22.50
Standard ntg2	1096 (512x8)	52	13.00
Standard ntg4	4096 (512x8)	38	9.50

Table 2: QuantumESPRESSO scaling values

Conclusions

The results of our tests show that OpenMP is important both to overcome memory per task constraints and to utilize all the cores of the nodes when GPUs are used. GPUs are bounded to tasks, so that one is limited to initiate a number of MPI tasks which is equal to the number of devices (unless the design and data distribution of the application would be changed completely). The OpenMP parallelization and the CUDA parallelization can coexist without major change in the code parallelization structure.

Looking at the results we can see that hybrid nodes with GPUs are 2 to 4 time faster than standard nodes, whereas if task groups are used on standard nodes, the scalability of the code is extended by a factor 2.

To sum up: hybrid nodes are the best choice to save CPU time, standard nodes are the best choice for maximum speed at the cost of more CPU time used.

The PA project shows that Tier-0 systems are suitable to simulate large nano particles, but the user has to know that there are many parallelization and setup parameters (tasks/threads, GPU/CPU, data distribution parameters like “ntg”) that have a big impact on scalability and performance. For the future it would be useful to develop some sort of “estimator” that can give to the users a guidance to setup all these parameters.

3.1.2 *High Performance MP2 for condensed phase simulations*

Project leader:	Prof. Joost VandeVondele, ETH Zürich
PRACE experts:	Iain Bethune, Ruyman Reyes, EPCC
PRACE facility:	HERMIT
Research field:	Chemistry and Materials
Application code:	CP2K[4]
PA number:	2010PA0723

Project objectives

The project leader, Prof. Joost VandeVondele, has recently implemented Møller-Plesset second-order perturbation theory (MP2) in CP2K with excellent performance and scalability using the GPW method, which reduces the formal scaling of MP2 from $O(n^5)$ to $O(n^3)$ in the number of basis functions. In order to support an application for large-scale HPC resources via PRACE regular Project Access, Prof. VandeVondele requested support from PRACE to optimise the key kernels of the MP2 calculation in CP2K. The work consisted of three stages: (1) serial optimisation of several key computational kernels; (2) OpenMP implementation of parallel 3D Fourier Transformation to support mixed-mode MPI/OpenMP use of CP2K; and (3) benchmarking the performance gains achieved by the new code on HERMIT for a test case representative of the proposed production simulations.

Optimization work and results

We implemented a template and auto-tuning framework of the key integration kernel routines, and added OpenMP parallelism to the 3D FFT. To evaluate the performance improvements achieved as a result of the above work we carried out benchmarking of the new version of the code using the PRACE HERMIT system, using a user-supplied `NH3_32_bulk` test case which computes the MP2 energy of 32 Ammonia molecules arranged in a cubic cell of side 10Å. Periodic boundary conditions and a TZV2P basis set were employed. This is representative of the type of calculations which were to be performed under Regular Access.

Figure 5 shows a comparison of the execution time of MPI-only execution with fully populated nodes - one MPI rank per CPU core. The blue line shows the wall-clock execution time required for the test case with the initial version of the code, whilst the red line shows the execution time using the current version of CP2K including the improvements implemented during this project. It should be noted that the execution time slightly increases with a larger number of cores with the new version with respect to the previous version. Figure 6 shows the speedup obtained in the modules affected by the project together with the overall speedup of execution. Since we are not using threads in this particular execution, we expect the FFT time to remain unchanged. We see a speedup of 8% from the optimised grid operations across all core counts. The drop in overall speedup at 8192 and 16384 cores is due to the introduction of the routine `replicate_mat_to_subgroup` in the MP2 implementation. This routine uses a message-round-a-ring approach with blocking `MPI_SendRecv` calls to distribute matrix data to the subgroups. As confirmed by thes timing reports, the cost of this scales linearly with the number of MPI processes, and may become a bottleneck to further scaling (on 16384 cores it takes ~10% of the total runtime). Investigating a tree-based broadcast is recommended, although due to lack of time, this could not be carried out within the scope of this project.

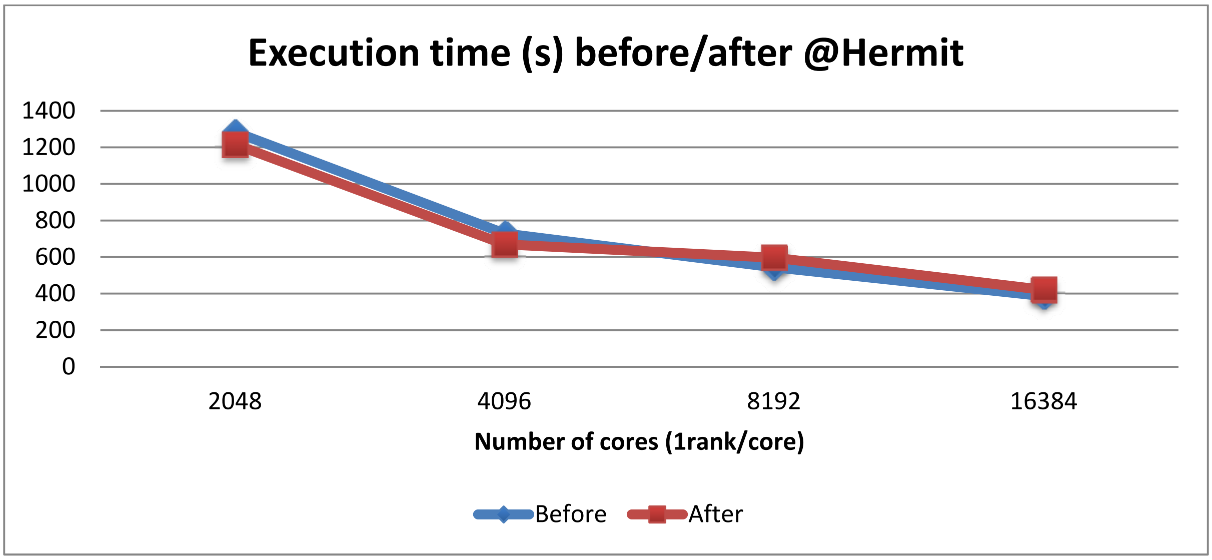

Figure 5: Execution time of the NH3 test case using 1 MPI rank per core

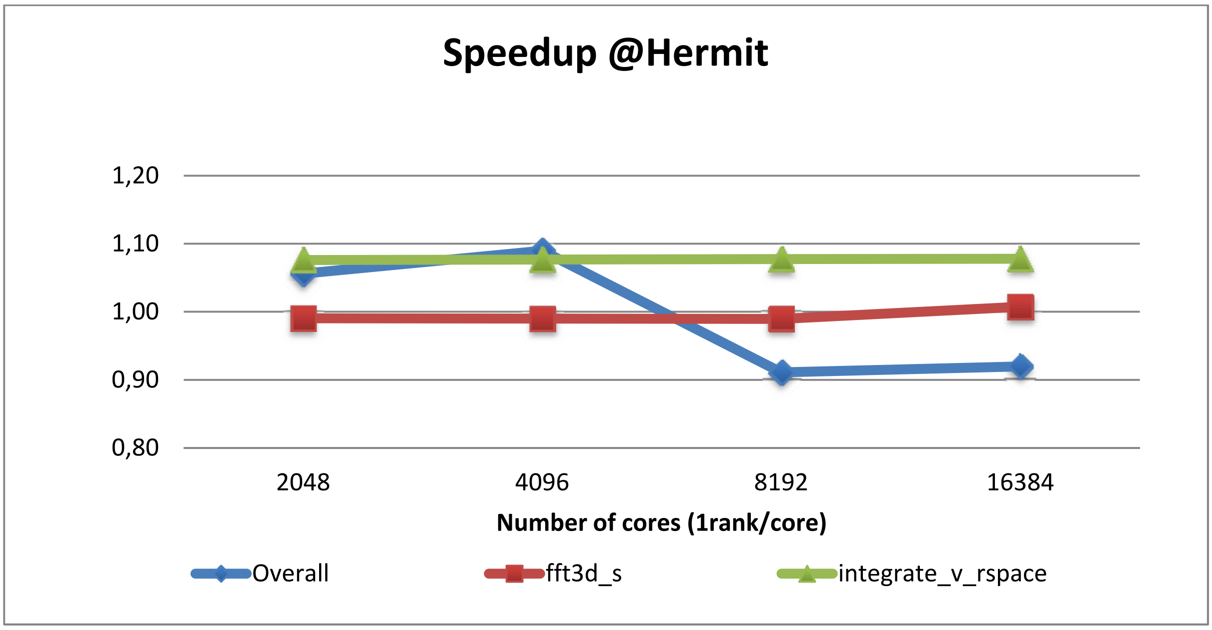

Figure 6: Speedup of each individual routine before and after optimization compared with the overall speedup in HERMIT.

We also measured the wall-clock time for calculations using multiple OpenMP threads per MPI rank. In this case we chose 8 threads to make best usage of the local caches and memory affinity, since there are 8 cores per NUMA region on the AMD ‘Interlagos’ processor architecture. The overall runtime of the initial and new versions of the code are shown in Figure 7 and the speedup over the initial code is shown in Figure 8. The speedup of the new FFT routines is around 3, which is a good result given that the largest FFT grid used in this calculation is 90^3. Again, the small but constant speedup from the optimised grid operations is observed. Overall, we find speedups of between 8% and 66% over the initial version of the code. The reason for this variation is the relative cost of the FFT compared with the MP2 integrals and Hartree-Fock computation.

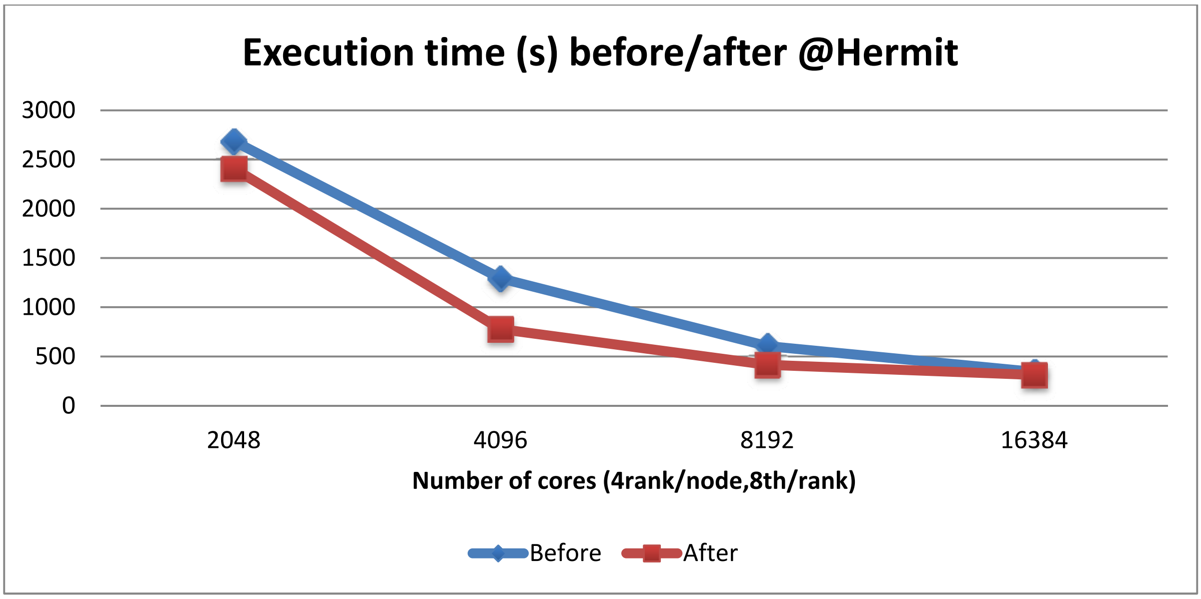

Figure 7: Execution time of the NH3 test case using 1 MPI rank per NUMA node (4 MPI ranks per node, 8 threads per rank)

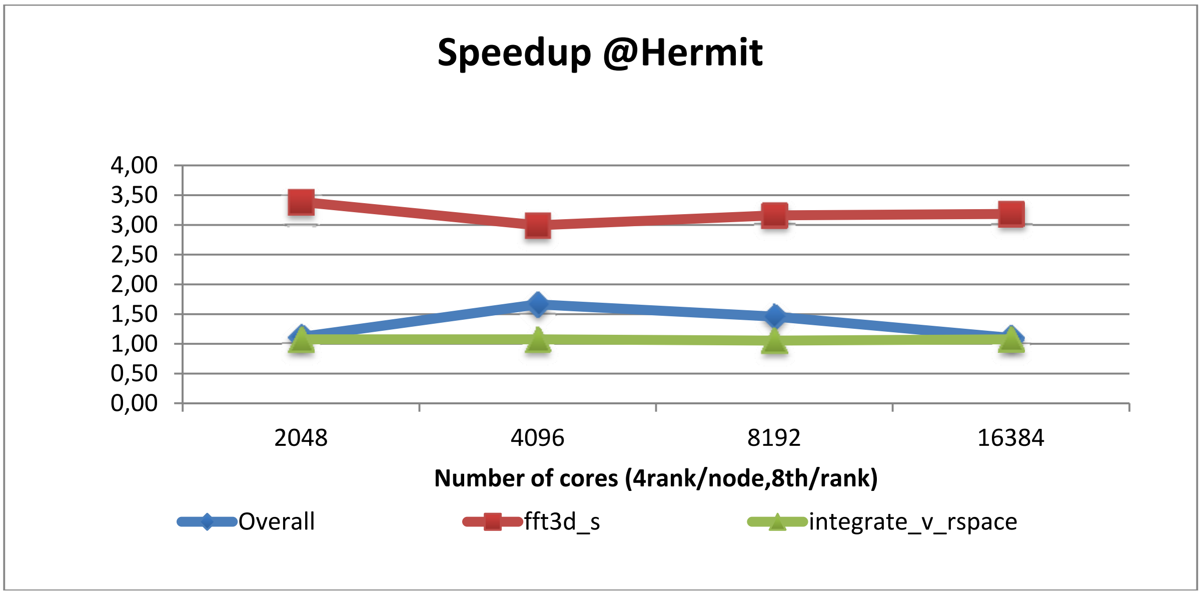

Figure 8: Speedup of the optimised version vs. the original one of individual routines in the current version of CP2K (4 MPI ranks per node, 8 threads per rank)

It is worth noting that when using 8192 or 16384 cores, the runs with 8 OpenMP threads per rank are 44% and 34% faster compared to the MPI-only runs. This highlights the value of using mixed-mode parallelism for these types of calculations at large scale.

Conclusions

To summarise, we have optimised two key parts of the CP2K code needed for efficient MP2 calculations at large scale. Firstly, we have used loop structure optimisation via templates and an auto-tuning approach, to generate code for key integration kernels that can be effectively optimised by the compiler, to give maximal performance on any given machine architecture. For the Gfortran on AMD Interlagos environment, we found a speedup of around 8% over the original code for these routines. Larger speedups may be obtained on other architectures for which the initial code is far from optimal. We have also implemented a new OpenMP-parallel 3D FFT routine, which gives speedups of 2-10x when using 8 OpenMP threads depending on the size of the FFT grid. For a test case representative of a real user job we found the FFT performance was improved by a factor of 3.

We have identified a communication-bound routine `replicate_mat_to_subgroup` which scales poorly with increasing numbers of MPI processes, although we note that already at 8192 cores it is more efficient to use mixed-mode MPI/OpenMP parallelism, which will mitigate this cost to some extent.

Finally, we note that Prof. Joost VandeVondele, whose use of the PRACE RI this project was designed to support, had submitted a request for 40 million CPU hours to perform groundbreaking MP2 calculations. Because of this we consider this preparatory access project to have been a success.

The project also published a white paper which can be found online under [5].

3.1.3 *I/O-optimization to improve parallel scalability of the meso-scale NWP-model HARMONIE*

Project leader:	Sami Niemelä, Finnish Meteorological Institute FMI
PRACE experts:	Sami Saarinen, CSC
PRACE facility:	HERMIT
Research field:	Earth Sciences & Environment
Application code:	Harmonie[6]
PA number:	2010PA0667

Project objectives

The aim of the project was to improve I/O of the Harmonie – a meso-scale weather forecasting code – to enable more frequent output fields than currently feasible. This would in turn imply better scalability and suitability to deploy finer resolutions i.e. larger model dimensions on current and future massively parallel HPC-environments.

Optimization work and results

We concentrated on I/O-optimization with emphasis on field output time reduction as it was considered crucial in improving Harmonie's parallel scalability.

In the original Harmonie code – based on synchronous I/O – the spectral and grid point space fields are collected to the master computational task for physical file outputting. Data collection (i.e. gathering) phases were called in routines WRSPECA and WRGP2FA and involved a large number of MPI_send / MPI_recv and some MPI_Gatherv MPI-calls over the default communicator. In addition, some considerable amount of norm calculations, field merge processing and other bit fiddling takes place. All this halts smooth progress of computational tasks and thus increases the time to complete the forecast.

In the optimized version the leading idea was to offload the work carried out by the routines WRSPECA and WRGP2FA to the sub-space(s) of I/O tasks (SS I/O) – distinct from the computational tasks. In the first instance the fundamental concept was to allocate a complete (or half-) node for each SS I/O task making available at the same time much more memory for such tasks. Later on, we also added OpenMP-multithreading to each SS I/O-task to be able to benefit a little from (otherwise) idle cores in a node. Optimization thus meant introduction of sophisticated asynchronous I/O scheme which speeds up computational tasks by making their progress smoother and nearly uninterrupted by I/O activity.

Figure 9 shows typical gains obtained from the used model size 512x600xL60. The red (bottom) curve represents the ideal situation where no output of spectral or grid-point fields occur (input patterns are still present, however). The blue (middle) curve represents optimized version running with 8 asynchronous SS I/O-tasks. And finally, the yellow (top) curve describes the situation before optimization. All these plots are for 6 hourly weather forecasts. In reality the gains are bigger due to the fact that a typical operational forecast is run against a 36 hour window.

Given a larger model size i.e. finer resolution, substantial gains are obtainable by using sub-space I/O scheme. At the end of project we had a brief access to spatially 10 times larger model size and it reacted gracefully when our optimization was applied leading to larger gaps between aforementioned curves. More SS I/O-tasks were also required to achieve this outcome.

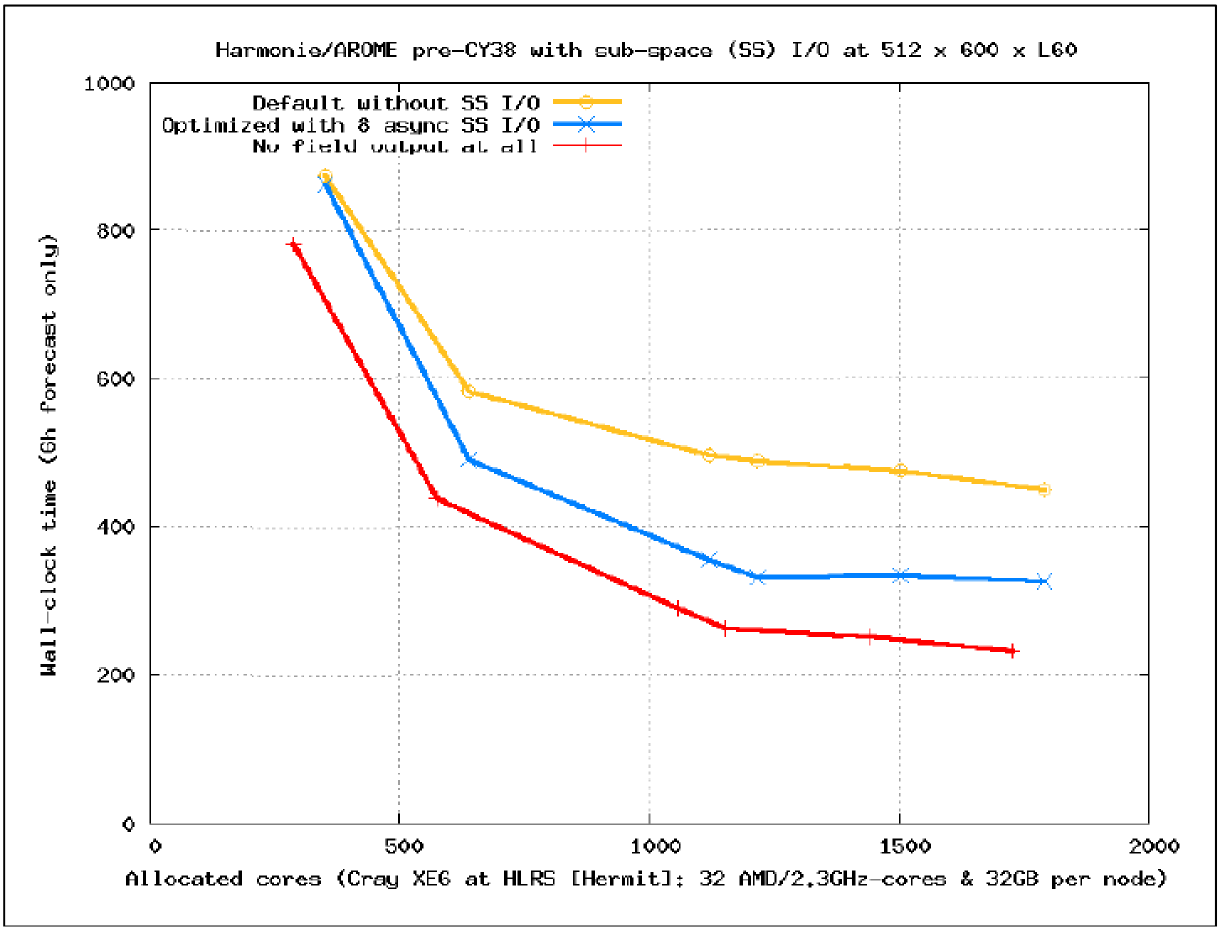

Figure 9: Scaling Curves of HARMONIE at HERMIT

Conclusions

We believe we succeeded quite well in our objective: the SS I/O works and fulfills the promises.

It must be noted, however, that the code base consists of several millions lines of mainly Fortran95 code spread over nearly 10,000 files. The code is written in a collaborative manner with some 20 years plus of age on its shoulders and cannot be changed dramatically in order to keep the general, mutually agreed data structures intact.

As we went along, we found additional problems in the code. One of them is still I/O related: the way how Harmonie handles the field input streams seems to be written to some extent sub-optimally, and gets exposed especially with a large number of computational tasks as well as with higher (yet realistic) resolutions. We hope to be able to address these in the near future.

The second avenue to pursue is the memory usage optimization. As the trend is towards larger model sizes with memory footprint per core not necessarily increasing so much or at all, we need to analyze and eliminate codes' memory slack sooner rather than later.

Finally, the ability to test with really large number of cores proved to be useful in order not only to reach scalability limits, but also to detect some bugs exposed only on large core tasks.

The PI stated that he hopes to be able to liaise with the PRACE project also in the future in order to sort out these problems; afterwards the Harmonie code should be ready for Tier-0.

3.2 Cut-off March 2012

In this cut-off two projects have been accepted and are reported in this section.

3.2.1 *Implementation of an Efficient Semi-implicit PDE Solver in the Cardiac Simulation Environment Beatbox*

Project leader:	Sanjay Kharche, University of Liverpool
PRACE experts:	Isabelle Dupays, Margareta Boiarciuc, IDRIS Maciej Szpindler, ICM
PRACE facility:	JUGENE / JUQUEEN
Research field:	Mathematical Sciences
Application code:	Beatbox Cardiac Simulation Environment[7]
PA number:	2010PA0784

Project objectives

Beatbox is a cardiac simulation code which is being developed at the University of Liverpool in collaboration with expert members of other universities and private companies. It incorporates the latest cardiac cellular models allowing the detailed simulation of ionic currents, intracellular processes, and to study the effects of drugs or surgical therapy in cardiac tissue.

The focus of this preparatory access PRACE project was to implement a PDE solving algorithm. Such an algorithm will improve the scalability of Beatbox by means of efficient parallelization and optimization.

The main performance bottlenecks in Beatbox have been identified to be the domain decomposition, the explicit PDE solver, and the MPI I/O. For the first two aspects (the domain decomposition, the explicit PDE solver), an expert from ICM has been assigned; for the third one (parallel file I/O issues), two experts from the IDRIS have been assigned.

Optimization work and results

Domain decomposition improvements and PDE solver

The Beatbox application runs with different geometry modes (2D/3D and with or without mesh geometry). This work addressed 3D mode with full geometry only. In this mode the computational domain is represented as a regular mesh with equidistant nodes (vertices). Each node describes a discrete point of the domain with its geometrical properties and physical state of the underlying heart tissue. Additionally, information on the tissue properties in the mesh vertex is available. Originally the domain decomposition was implemented using ideal 3D sub-cubes of the full domain regarding the number of MPI processes initialized. This method does not take tissue properties of the mesh nodes into consideration. It leads to a potential poor load balance between processes while tissue is not equally distributed between cubical sub-domains in realistic (representing heart tissue) scenarios.

The PI asked for decomposition improvements by library integration i.e. ParMetis [8]. It was decided to use ParMetis to address decomposition related issues. ParMetis is a parallel implementation of widely used graph partitioning library called Metis. General requirements of the library is to have mesh represented as a graph – vertices as a graph nodes and vertex neighbourhood as a graph adjacencies; it is also required to have a mesh initially distributed and with each sub-graph being non-empty. These requirements lead to a plan for ParMetis integration with the Beatbox code.

It was decided to use the original Beatbox decomposition as an initial ParMetis partition. Graph representation of the mesh has been implemented and vertex renumbering scheme has been introduced while ParMetis requires nodes to be continuously numbered (identified) across all parallel processes.

To address load balancing issues tissue related information from the vertices has been used as weights for a ParMetis graph. It was essential while ParMetis can handle graph weights as a partitioning constraint. Naive tissue weights correspondence has been implemented. In order to utilize most of the ParMetis features geometry parameters of the mesh have been also used as graph vertices geometrical positions.

After all required graph structures were implemented and tested the ParMetis function `ParMETIS_V3_PartGeomKway` has been used resulting in improved partitioning of the graph. Visual analysis of the resulting decomposition provides initial evaluation of the refinement quality (see Figure 10).

ParMetis integration has been implemented as a function indented to be called from Beatbox's state function (`state.c`), resulting with a preposition on new decomposition. Function metis_repart has been delivered to the PI.

While Beabox uses halo regions to exchange some of the information between sub-domains it also needs to be implemented also for the case of ParMetis based non-uniform decomposition. No guidelines on halo implementation have been given from the PI. To fully integrate the ParMetis with Beatbox code it is essential to have halo usage resolved. This needs additional effort. For this reason, full results on the load balancing improvement cannot be determined. It was also discussed to separate the ParMetis part as a standalone pre-processing utility but the project has ended before this could be achieved.

PDE solver improvement has been not addressed as all project time has been spent on decomposition related work. PETSc library usage has been considered but no further details were discussed.

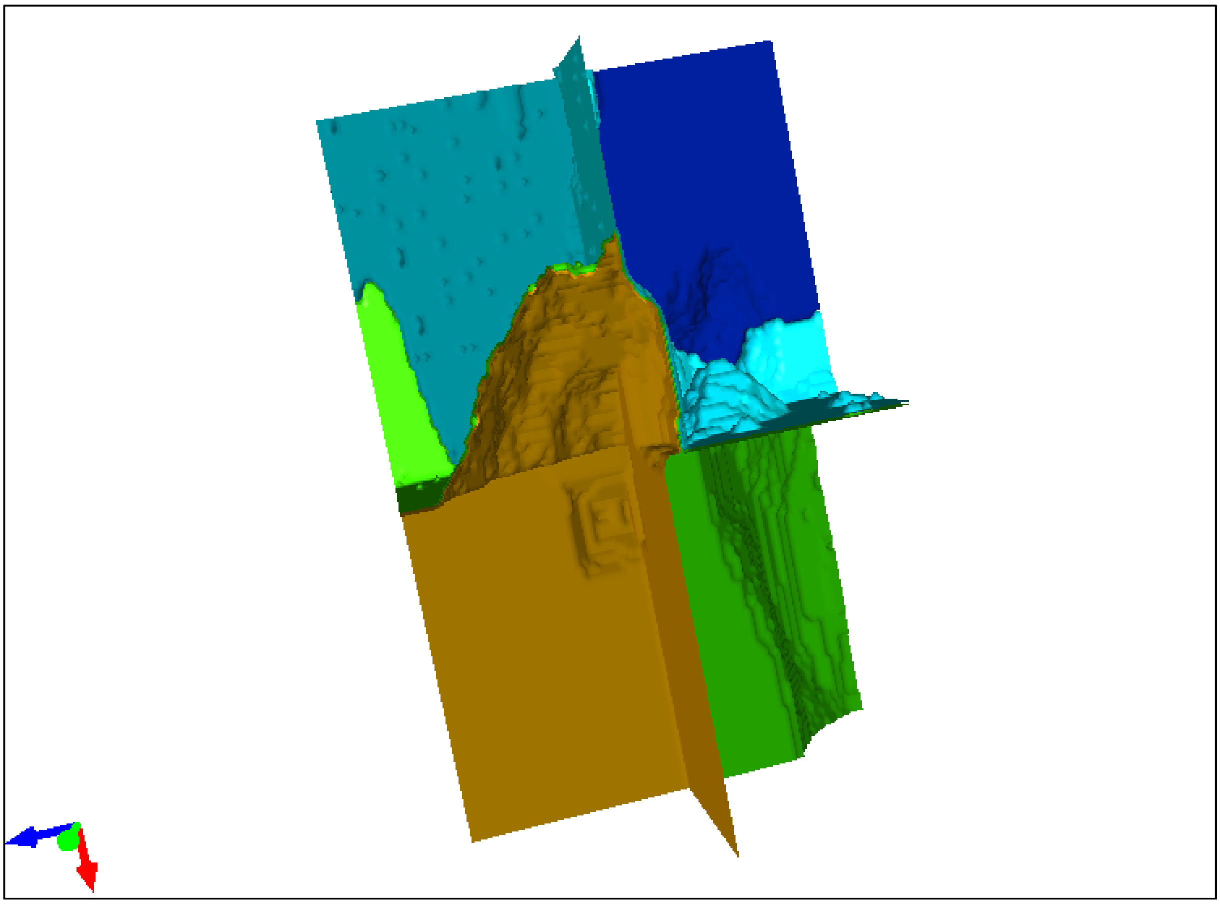

Figure 10: ParMetis based 2x2x2 decomposition of the human atrium

MPI I/O issues

The PI requested improvement of the parallel I/O in his code. Therefore the I/O strategy of the code has been analyzed. Periodically, output of numerical solvers is taken in the form of ASCII records, PPM files, or binary dump of the complete simulation. The main file output in Beatbox is handled by ppmount.c and dump.c files. As functions in dump.c file are used only once in any given simulation, the ppmount.c file has been investigated further on. The IDRIS experts finally concluded that the way of using MPI I/O seems to be quite efficient, so no further improvements have been performed for this task.

Conclusions

The ICM expert has implemented a function for decomposition improvement following the guidelines from the PI. While the details on the full integration with the user application have not been established, final results of the work could not be determined in terms of application scalability during the project lifetime. Decomposition improvement itself has been clearly identified as significant improvement to the original scheme used. Additional effort is required to address the full integration of the work results with the PI code. PDE solver improvements have been requested but have not addressed because of the project time limitations.

The PI has submitted a proposal for the DECI-10 call; for future Tier-0 access important steps have been made, but the code does not seem to be fully suitable yet.

3.2.2 *Optimizing GPAW on GPUs*

Project leader:	Prof. Martti Puska, Aalto University School of Science
PRACE experts:	Martti Louhivuori, Jussi Enkovaara, CSC

PRACE facility:	CURIE hybrid nodes
Research field:	chemistry and materials
Application code:	GPAW[9]
PA number:	2010PA0897

Project objectives

GPAW is a density-functional theory (DFT) electronic structure calculation program package based on the projector augmented wave (PAW) method. It uses real-space uniform grids and multigrid methods or atom-centered basis-functions. Electronic structure simulations are widely used to solve scientific problems in materials physics and chemistry and account for a large portion of the computing resources consumed in high-performance computing.

The CPU version of GPAW scales nicely using MPI and is capable of utilizing tens of thousands of cores efficiently. A GPU version of GPAW has also been implemented and tested on a limited number of GPUs. Having a multi-GPU accelerated version of the code capable of good scaling behavior and order of magnitude speed-ups would be highly beneficial for the community. This project aimed to accelerate GPAW using multiple GPUs.

First, the parallel scaling of the GPU version of GPAW on a large number GPUs needed to be profiled and investigated in order to gain an understanding of the real bottlenecks and optimal performance parameters. This information could also then be used to find the optimal set-up of stencil communication schemes for boundary data exchange. Second, issues that had been identified in the MPI communication of the GPU version needed to be resolved. Possibilities for implementing new parallelization levels over electronic states, similar to the CPU version, would also be looked into.

Optimization work and results

GPAW is written in the Python programming language with performance critical sections included as extensions written in C. The GPU version uses extensively the PyCUDA toolkit for CUDA integration and custom CUDA kernels to speed-up the performance critical parts.

GPAW uses a multi-grid solver for solving the Poisson equation in a fine grid and custom CUDA kernels have been written for all the basic operations: finite-difference stencils for the Laplace operator as well as restriction and interpolation operations between different grids. The calculations are done using both shared memory and registers to reduce global memory redundancy. The custom CUDA kernel for finite difference calculations is automatically generated from C code for each order-k stencil to optimize shared memory and register usage and to completely unroll inner loops. The entire Poisson solver is implemented to use only the GPU-side to avoid slow memory copies between the GPU and the host. Most basic linear algebra uses NVIDIA’s CUBLAS library, but to allow for simultaneous updating of a group of eigenvectors custom blocking versions of level 1 BLAS routines are used.

The code is parallelized using MPI with support for domain decomposition of the real space grid or for decomposition over k-points. Domain decomposition involves communication with the nearest neighbor nodes and in the GPU version this requires data movement between the GPU and CPU memories and communication of the data between CPUs using MPI. Several different approaches to this process were implemented with different levels of overlapping computation and communication and, depending on grid and transfer sizes, a suitable one is selected.

After optimization of the MPI communication, the GPU version of GPAW achieved a good level of speed-up compared to the CPU version, with the GPU version being up to 15 times faster than the CPU version. This is a marked improvement from the approximately 8x speed-up of a single GPU over a single CPU.

To test the strong scaling, ground state energy calculations of a C^{60} fullerene molecule containing 240 valence electrons were performed using increasing number of MPI tasks. Good scalability was achieved for up to four GPU-cards but beyond that the scaling factor leveled off. The testing was done on the Vuori cluster at CSC, which has 7 GPU nodes connected via an Infiniband network. Each node has two Intel Xeon X5650 CPUs and two GPU cards, either NVIDIA Tesla M2050 or M2070.

To test the weak scalability, two systems were tested on two different machines: a varied length carbon nanotube (80-320 atoms with 1-12 MPI tasks) on the Vuori cluster and a varied size bulk silicon substrate (95-320 atoms with 1-192 MPI tasks) on the CURIE supercomputer at GENCI. CURIE has 144 GPU nodes connected via an Infiniband network with each node having two Intel Xeon E5640 CPUs and two NVIDIA Tesla M2090 GPUs. On Vuori, the GPU version showed increasing speed-ups compared to the CPU version with a maximum of approximately 12 times faster with 12 MPI tasks. On CURIE, the speed-ups leveled off at 15 times faster with 8 MPI tasks with a slow decrease beyond 128 MPI tasks as is shown in Figure 11.

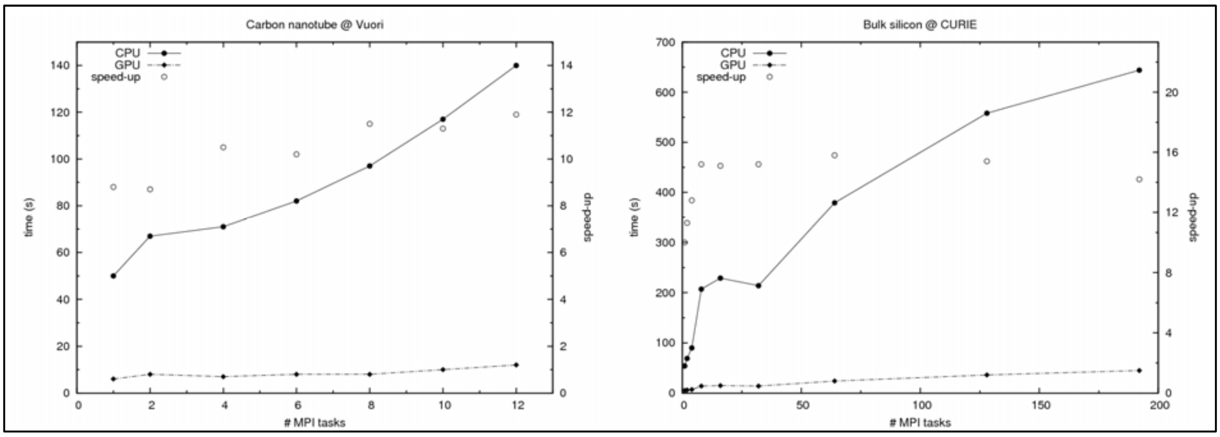

Figure 11: Weak scaling results of both CPU and GPU versions of GPAW on Vuori cluster (left) and CURIE supercomputer (right) showing run times (left axis) as well as a relative speed-up (right axis) of using the GPU version over the CPU version.

Conclusions

The GPU version of GPAW has been optimised for usage on multi-GPU clusters using MPI for the parallelisation. Compared to the CPU version, significant speed-ups (up to 15 times) were achieved using multiple GPUs for ground state DFT. Also an excellent weak scaling performance was achieved and demonstrated on a local cluster and the CURIE supercomputer. Further improvements (especially of the strong scaling) may be attempted e.g. by implementing new parallelisation levels similar to the CPU version. GPAW is clearly suitable for Tier-0 systems and with a suitable simulation system the GPU version of GPAW can be used effectively in parallel on a large number of GPU nodes.

The project also published a white paper which can be found online under [10].

3.3 Cut-off June 2012

In the June 2012 cut-off five projects were accepted. These projects are outlined here together with the abstracts of their white papers. There is a reference given to each white paper on the PRACE-RI webpage. The titles of the white papers are identically to the project title (in the section headline); the authors are the PRACE experts in collaboration with the project leader.

3.3.1 *Massive Parallel Navier-Stokes Solver for Breaking Waves*

Project leader:	Stéphane Glockner, Institut de Mecanique et d'Ingnierie de Bordeaux
PRACE experts:	Nicole Audiffren, Hilde Ouvrard, CINES
PRACE facility:	CURIE thin and fat nodes
Research field:	Engineering and Energy
Application code:	THETIS[11]
PA number:	2010PA0937

Abstract of the white paper

It has already been shown that the numerical tool Thétis based on the resolution of the Navier-Stokes equation for multiphase flows gives accurate results for coastal applications, e.g. wave breaking, tidal bore propagation, tsunamis generation, swash flows, etc. In this study our goal is to improve the time and memory consumption in the set-up phase of the simulation (partitioning and building the computational mesh), examining the eventual benefits of an hybrid approach of the Hypre library, and doing fine tuning in implementation of the code on Curie Tier-0. We also implement parallel POSIX VTK and HDF5 I/O. Thétis is now able to run efficiently up to 1 billion mesh nodes at 16384 cores on CURIE in a production context.

The white paper can be found online under [12].

3.3.2 *Linear Scaling Methods for Quantum Hall Transport Simulations*

Project leader:	Prof. Stephan Roche, Catalan Institute of Nanotechnology
PRACE experts:	Paschalis Korosoglou, Alexandra Charalampidou, GRNET
PRACE facility:	CURIE thin and fat nodes, HERMIT
Research field:	Fundamental Physics
Application code:	no name
PA number:	2010PA0977

Abstract of the white paper

This study has focused on an application for Quantum Hall Transport simulations and more specifically on how to overcome an initially identified potential performance bottleneck related to the I/O of wave functions. These operations are required in order to enable and facilitate continuation runs of the code. After following several implementations for performing these I/O operations in parallel (using the MPI I/O library) we showcase that a performance gain in the range 1.5-2 can be achieved when switching from the initial POSIX only approach to the parallel MPI I/O approach on both CURIE and HERMIT PRACE Tier-0, systems. Moreover we showcase that because I/O throughput scales with an increasing number of cores overall the performance of the code is efficient up to at least 8192 processes.

The white paper can be found online under [13].

3.3.3 *GPU Implementation of TD-DFT algorithm and parallel IO improvements for the DP code*

Project leader:	Francesco Sottile, Ecole Polytechnique Palaiseau
Project team members:	Claudia Rödl
PRACE experts:	Franck Houssen, Pierre Kestener, CEA, Dusan Stankovic, Petar Jovanovic, Vladimir Slavnic, IPB
PRACE facility:	CURIE fat and hybrid nodes
Research field:	Chemistry and Materials, Fundamental Physics

Application code:	DP[14]
PA number:	2010PA1148

Abstract of the white paper

Main goal of this PRACE project was to evaluate how GPUs could speed up the DP code (a linear response TDDFT code). Profiling analysis of the code has been done to identify computational bottlenecks to be delegated to the GPU. In order to speed up this code using GPUs, two different strategies have been developed: a local one and a global one. Both strategies have been implemented with cuBLAS and/or CUDA C. Results showed that one can reasonably expect about 10 times speedup on the total execution time, depending on the structure of the input and the size of datasets used, and speedups up to 16 have been observed for some cases.

The white paper can be found online under [15].

3.3.4 *FLY on Cray: Porting of a Cosmological Parallel Treecode*

Project leader:	Dr. Vincenzo Antonuccio-Delogu, INAF/National Institute for Astrophysics
PRACE experts:	Maciej Cytowski, ICM, Joachim Hein, SNIC-LU, Jörg Hertzer, GCS-HLRS
PRACE facility:	HERMIT
Research field:	Astrophysics
Application code:	FLY[16]
PA number:	2010PA0940

FLY is an N-body code for cosmological simulations of the evolution of the Large Scale structure of the Universe. The capability to simulate problems with a very large number of particles N (i.e. with a high mass resolution) is a particular focus of FLY. A three phase Barnes-Hut algorithm is deployed to reduce the problem complexity. The FLY MPI version scales as N log(Np), with N denoting the number of bodies and Np denoting the number of MPI tasks. In order to improve the efficiency of the code, the latest version of FLY utilizes threads via OpenMP to parallelize the MPI tasks within the nodes.

Abstract of the white paper

In this white paper we report work that was done to investigate and improve the performance of a mixed MPI and OpenMP implementation of the FLY code for cosmological simulations on a PRACE Tier-0 system Hermit (Cray XE6).

The white paper can be found online under [17].

3.3.5 *Optimization of multiple sequence alignment software clustalw*

Project leader:	Prof. Dr. Plamenka Borovska, Technical University of Sofia
PRACE experts:	Soon-Heum Ko, SNIC-LiU
PRACE facility:	JUQUEEN
Research field:	Medicin and Life Sciences
Application code:	ClustalW[18]
PA number:	2010PA1155

Abstract of the white paper

This activity with the project PRACE-2IP is aimed to investigate and improve the software ClustalW on the supercomputer Blue Gene/Q, so-called JUQUEEN, for the case study of the influenza virus sequences. Porting, tuning, profiling, and scaling of this code has been accomplished in this aspect. A parallel I/O interface has been designed for efficient sequence dataset input, in which sub-groups' local masters take care of read operation and broadcast the dataset to their slaves. The optimal group size has been investigated and the effects of read buffer size on read performance has been experimented. The application to ClustalW software shows that the current implementation with parallel I/O provides considerably better performance than the original code in view of I/O segment, leading up to 6.8 times speed-up for inputting dataset in case of using 8192 JUQUEEN cores.

The white paper can be found online under [19].

3.4 Cut-off September 2012

In this cut-off three projects have been accepted which are described in this section.

3.4.1 *Strong Scaling of LibGeoDecomp on Blue Gene/Q*

Project leader:	Andreas Schäfer, Friedrich-Alexander-Universität Erlangen-Nürnberg
PRACE experts:	Luis Fazendeiro, SNIC-CHALMERS
PRACE facility:	JUQUEEN
Research field:	Mathematics and Computer Science
Application code:	LibGeoDecomp[20]
PA number:	2010PA1184

This project detected an internal compiler error in the compilation of their code. The vendor was informed, but the compiler fix was not available in time. Finally the PI was able to perform scaling runs (one successful run even up to the whole system with 1.8 Mio ranks), but the envisaged optimization work could not be performed fully and no white paper could be written.

3.4.2 *CASINO for large solid catalyst systems: configuration numbers and population control*

Project leader:	Prof. Philip Hoggan, Clermont University
PRACE experts:	Margareta Boiarciu, IDRIS, France
PRACE facility:	CURIE thin and fat nodes, JUQUEEN
Research field:	Chemistry and Materials
Application code:	CASINO[21]
PA number:	2010PA1186

Abstract of the white paper

The main goal of this project was to port the CASINO code to a Blue Gene/Q system (JUQUEEN) and use it as a test reaction benchmark of hydrogen dissociation on copper surfaces. Some bottlenecks have been identified and investigated in this project. The main restriction of efficient use of supercomputers is efficient use of the RAM, in particular for loading and handling configuration data.

The white paper can be found online under [22].

3.4.3 *Combinatorial Models for Topology Aware Mapping*

Project leader:	Prof. Cevdet Aykanat, Bilkent University
PRACE experts:	Joerg Hertzer, GCS-HLRS, Michael Schliephake, SNIC-KTH
PRACE facility:	CURIE thin and fat nodes, HERMIT
Research field:	Mathematics and Computer Science
Application code:	PaToH[23]
PA number:	2010PA1188

The PRACE collaborators from this project provided support on the usage of the execution systems. During project life-time it turned out that the group of the PI didn’t needed as much support on the optimization itself as they expected in their application. They continued the envisaged tests on the Tier-0 systems on their own and reported that they are now ready to apply for Tier-0 regular access.

3.5 Cut-off December 2012

The projects from this cut-off are not fully completed at the time this deliverable is written (they run until end of July), so the white paper is not finalized yet. Therefore these projects are reported in the form of the collaborator report for this deliverable.

3.5.1 *MAGICS - Modelling the ArrhythmoGenic Influence of the Cardiac conduction System*

Project leader:	Edward Vigmond, Universite Bordeaux 1
PRACE experts:	Joachim Hein, SNIC-LU
PRACE facility:	CURIE thin nodes
Research field:	Medicine and Life Sciences
Application code:	CARP[24]
PA number:	2010PA1332

Project objectives

The application CARP utilised by the project is designed to perform high-resolution cardiac simulations. When dealing with complex human heart simulations meshes with millions of degrees of freedom are required. At each time step the simulation is required to solve a substantial distributed linear system.

The scientific aim of the project is to understand the role of the Purkinje System which is the electrical conduction system of the heart. It is responsible for relaying the signals leading to the rapid and coordinated contraction of the heart. It is very difficult to study the Purkinje System experimentally or clinically.

The project aims to improve the strong scalability of the code by deploying better pre-conditioners, optimising the partitioning of the mesh and algorithmic improvements in the deployed solver library. The code exists in three different versions. The first version uses the PETSc library for its linear algebra operations with the Boomer AMG preconditioner. The second version, independent of PETSc, deploys the parallel toolbox with a custom designed algebraic multigrid preconditioner. The third version uses a GPU version of the parallel toolbox.

Prior to the project the parallel toolbox has only been tested on up to 60 cores but showed better performance than the PETSc version. The project aims to test, improve and tune the parallel toolbox version on a large number of cores, to assess the scalability of this version and to further compare the performance to the PETSc version.

Optimization work and results

When applying for the PA project the PI asked for a limited amount of consultancy from PRACE mainly for code profiling. The following items were discussed with the project team so far. The team received advice on MPI-IO functionality to write alternating array structures to file.

To assess the performance of the individual parts of the code, it was decided to use the Scalasca tool. Since the parallel toolbox uses C++ coding, we had an in-depth discussion on how to get reliable results from a C++ code from the Scalasca performance analysis tool. This is required to avoid the run-time under the tool becoming excessive, which would turn the results meaningless.

The project requires a special PETSc library version built together with the hypre and parmetis library. This special version was not initially installed on the Curie system and built as part of this project, which was involved since it required a number of prerequisites with very specific version numbers. The build procedures were shared with the GENCI support experts.

Initial tests of CARP using the parallel toolbox showed excellent scaling up to 96 mpi tasks but performed significantly worse than expected on 128 mpi tasks. The applicants shared the output from an instrumented run using Scalasca. From the Scalasca output the problem could be traced to the parallel toolbox making many calls to MPI_Alltoallv using a rather small message size. Based on this advice the developers of the parallel toolbox retuned their MPI usage and achieved strong scaling on up to 192 processors, which coincided with their expectations for the chosen problem size.

Conclusions

At the time of reporting, the project has not been completed as the project got a prolongation. The support work therefore still runs until end of August and at the moment no final conclusions can be drawn yet.

3.5.2 *Numerical model for the thermal energy storage tank with integrated steam generator (TES-SG) based on the OpenFOAM software package*

Project leader:	Dr. Evgeny Votyakov, The Cyprus Institute CaSToRC
PRACE experts:	Alexandra Charalampidou, Paschalis Korosoglou, GRNET, Joerg Hertzer, GCS-HLRS
PRACE facility:	HERMIT
Research field:	Engineering and Energy
Application code:	OpenFOAM[25]
PA number:	2010PA1359

Project objectives

The OpenFOAM software package is used in the work of the PI to design and optimize a 3D numerical model of a thermal energy storage (TES) tank with integrated steam generator (SG). The model includes heat and mass transfer partial differential equations solved by means of OpenFOAM. OpenFOAM (Open Source Field Operation and Manipulation) is a C++ toolbox for the development of customized numerical solvers with pre-/post-processing utilities for the solution of continuum mechanics problems, including computational fluid dynamics (CFD). The code is released as free and open source software under the GNU General Public License.

TES is a technology to store thermal energy in a reservoir for later use. The numerical model will be part of the OPTS project (Optimization of Thermal Storage with integrated steam generator), which aims at developing a new TES system based on single tank configuration using stratifying Molten Salts (MS, Sodium/Potassium Nitrates 60/40 w/w) as the heat storage medium at temperatures reaching 550 °C. The TES tank is integrated with a Steam Generator (SG) to provide efficient, reliable and economic energy storage for the next generation of trough and tower plants.

Optimization work and results

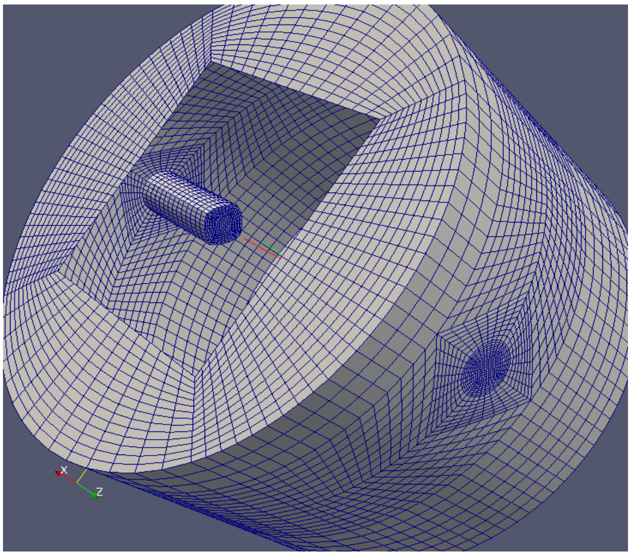

Figure 12: Thermal Energy Storage tank

A realistic 3D model which solves jointly Navier-Stokes and heat conduction equations including buoyancy effect with Boussinesq approximation was developed using OpenFOAM.

The mesh consists of macroscopic hexahedra (blocks) of different shapes and geometric sizes. The blocks are subdivided in cells by the OpenFOAM utility named blockMesh (Figure 12). The amount of cells in each block is proportional to N^3, where N is a characteristic number (number of partitions of the typical edge in the system).

This mesh allows performing equally loaded decomposition of the computational domain. Therefore, all of the cores treat an almost equal number of the cells, and the communication time between adjacent cores (blocks) is minimal. Computationally, to work on such a structured mesh is the same as to work on the equally distant cubic grid.

In order to run OpenFOAM in parallel on distributed processors the mesh is decomposed using the OpenFOAM decomposePar utility. In addition to the manual decomposition method which was used to assign equal numbers of cells to all processors, simple decomposition methods have also been tested. Scalability tests were performed on the meshes of total capacity equal to $320N^3$ cells with N=16, 32, 64. The largest mesh (N=64) consists of 83,886,080 cells.

For N=32 and using three different methods for decomposition our timing results of the solver (CPU Time) are shown in Figure 13. As it can be seen for our case study the decomposition strategy, which attempts to minimize the number of processor boundaries, provides the best scaling overall. Similar results are obtained when using N=64 at the decomposition stage. For this case we have tested only the manual and scotch decomposition methods.

It should be noted that the results given in this section include only the timing results of the solver (parallel) step (not the decomposition and reconstruct steps). When these additional (serial) steps would have been also taken into consideration the manual decomposition method may be preferable in some cases assuming the overall absolute wall time is the critical metric.

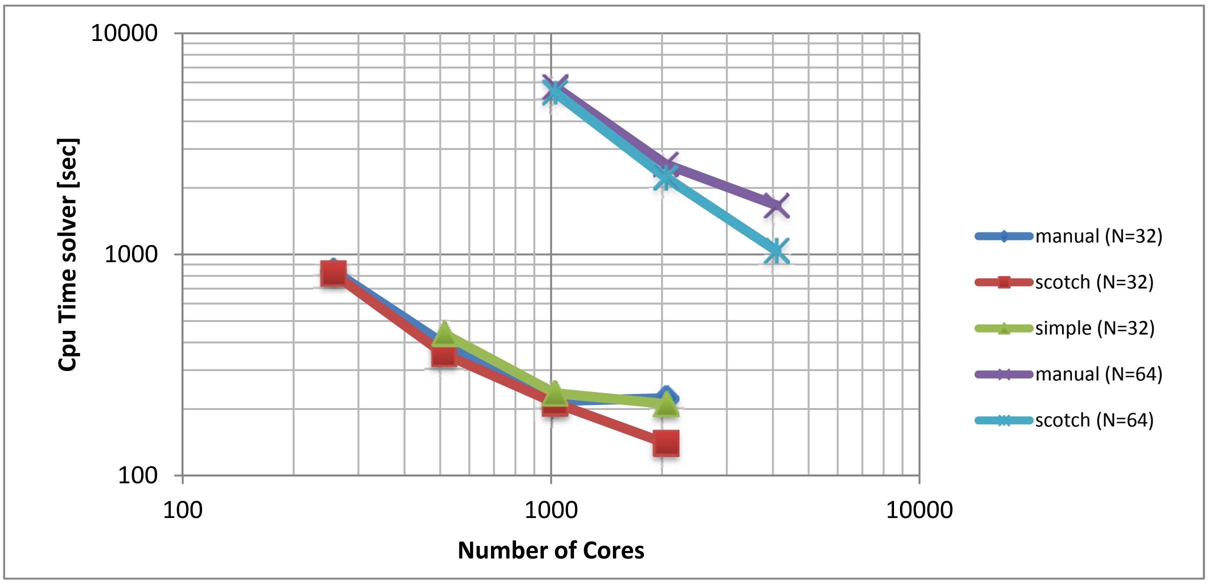

Figure 13: Solver timings for N=32 and N=64 meshes using different methods of decomposition.

Conclusions

OpenFOAM with this application proved to be suitable for current Tier-0 systems up to several thousand cores. The efficiency depends on the number of cells of the finite volume model used, increasing significantly with more detailed models, and requiring greater number of cells. Therefore, for more complicated problems, which require more detailed models, OpenFOAM is expected to scale well on higher core numbers also.

Although this was not the original intention of the project, within this project already useful simulation results could be obtained.

3.5.3 *Optimization of the code octopus*

Project leader:	Prof. Fernando Manuel da Silva Nogueira, University of Coimbra
PRACE experts:	Jonathan Vincent, SNIC-KTH, Petros Souvatzis, SNIC-UU
PRACE facility:	CURIE thin nodes, CURIE hybrid nodes, JUQUEEN
Research field:	Chemistry and Materials
Application code:	OCTOPUS[26]
PA number:	2010PA1404

Project objectives

Octopus is a real-space, real-time computer code to simulate the dynamics of electrons and nuclei under the influence of external time-dependent fields in the framework of Time-Dependent Density Functional Theory (TDDFT).

The main focuses of the code are time-dependent calculations. However, the first step of such calculation is the solution of the unperturbed reference ground state. Both the solution of the ground state and the solution of the time-dependent problem are process- and data-parallelized using MPI. Unfortunately, the ground state self-consistent field calculations require significantly more MPI inter-process communication than the subsequent time dependent calculations.

The communication for the ground state calculations of Octopus consist of two types: Firstly to make sure that the states calculated on different processors are orthogonal and secondly halo exchange between the different domains.

The current implementation of the ground state calculations involves an implementation of SCALAPACK, which is crude, not well optimized, and needs to be improved. So the main objective of the project was to improve the parallel scaling of the ground state calculations by improving the SCALAPACK implementation, or other appropriate methods.

Optimization work and results

The optimization work was considerably delayed by various factors, e.g. Petros was unfortunately unavailable for much of the project, and Jonathan was also away for much of April and May. The code is also quite complex making quick progress difficult.

Similarly as a prelude to work it is necessary to understand the current SCALAPACK implementation and the limitations of it, this requires deep understanding of the code.

During this analysis it was realized that the developers had misunderstood how BLACS works, so it should be possible to make a better SCALAPACK implementation, as BLACS grids can be generated from any MPI communicator.

Preliminary analysis of the code has also been completed, looking at how well the code performs examining chlorophyll complexes with varying number of atoms. From the tables Table 3 and Table 4 it can be seen that the MPI synchronization time (i.e. the time spent in MPI collectives waiting for all the callers to arrive) is very large even for a relatively small number of MPI tasks, indicating that the code does not scale to a large number of cores.

MPI tasks	Number of atoms			
	180	441	650	1365
24	14.7%	12.7%		
48	18.6%	22.6%	20.3%	
96	22.0%	35.4%	38.5%	
192	26.3%	45.1%	56.7%	28.5%
384				32.2%
768				35.3%

Table 3: MPI synchronization time (fraction of total time) for an LDA calculation of Chlorophyll.

MPI tasks	Number of atoms			
	180	441	650	1365
24	5.4%	5.1%		
48	6.4%	7.8%	5.8%	
96	6.9%	10.3%	7.0%	
192	10.1%	14.3%	6.9%	6.2%
384				7.7%
768				11.3%

Table 4: MPI execution time (fraction of total time) for an LDA calculation of Chlorophyll.

Conclusions

As previously stated it should be possible to improve how SCALAPACK interacts with the parallel infrastructure of the Octopus code, by using a better integration with the MPI interface.

There is a considerable communication bottleneck in the SCF part of the code, with even for small numbers of cores 30% or more of the runtime taken up in the MPI sync in the collectives (principally MPI_bcast and MPI_allreduce) indicating some load imbalance, which could hopefully be significantly improved by fine tuning the SCALAPACK implementation. This will be considered after the formal end of the project.

3.6 Cut-off March 2013

These projects will run beyond the end of PRACE-2IP and PRACE-3IP will take over on September 1. Therefore the projects are not finalized, but the work done in PRACE-2IP is described here. There is one exception, 2010PA1500 (3.6.1) is supported only by PRACE-2IP, the support work ending August 31, and the project is reported here in a full collaborator report.

3.6.1 *Explicit solvent Molecular Dynamics Simulation of ribosome unit with Gromacs*

Project leader:	Prof. Leander Litov, University of Sofia "St. Kliment Ohridski"
PRACE experts:	Valentin Pavlov, NCSA
PRACE facility:	FERMI, JUQUEEN
Research field:	Life Sciences
Application code:	GROMACS[27]
PA number:	2010PA1500

Project objectives

The aim of the project is to test the performance scalability of the latest Gromacs version (4.6.3 as of the time of this writing) on IBM Blue Gene/Q and to optimize for production runs a simulation setup for explicit solvent molecular dynamics simulations of huge macromolecular ribosome systems containing several millions of atoms. Tests of different optimizations of Gromacs routines that calculate atomic forces will be performed.

The behavior of Gromacs 4.6.3 molecular dynamics simulations involving millions of atoms running on Blue Gene/Q is still not deeply investigated and such a study is of great importance for the research community which has access to HPC platforms like FERMI at CINECA and JUQUEEN at JSC. The performance of the Gromacs Blue Gene/Q kernels will be investigated and code will be optimized in collaboration with Gromacs development team.

Optimization work and results

The system under test was an E.Coli ribosome in water solute measuring 2,230,000 atoms. The intent was to measure performance with different variants for the non-bonded schemes (reaction field vs. PME) and with different cut-off schemes (charge group vs. verlet pair-lists). Unfortunately, we found that Gromacs 4.6.3 on Blue Gene/Q cannot process the system with the PME scheme, regardless of the configuration of the package. We have tried different compilers, options, different FFTW linkings, but the package keeps warning that the system is not well equilibrated. On the other hand, the same system with the same configuration files runs fine on the GPU version of the package, and also when run on an ordinary Linux cluster, so we conclude that there is some subtle problem that presents itself on a Blue Gene/Q system. This requires further investigation and we'll continue working with the original developers in order to figure out the reasons for this condition and fix it up.

In Table 5 and Figure 14 we present the scalability tests we performed with the reaction field non-bonded scheme, with the two different cut-off schemes (charge group vs. verlet pair-lists).

	Performance, simulated time per day (ns)	
Nodes	Charge Group 1 node = 64 mpi x 1 thread	Verlet 1 node = 2 mpi x 32 threads
32	1,344	0,900
64	2,200	1,662
128	3,180	2,662
256	4,280	3,879
512		5,094
1024		5,689

Table 5: Scalability tests Gromacs

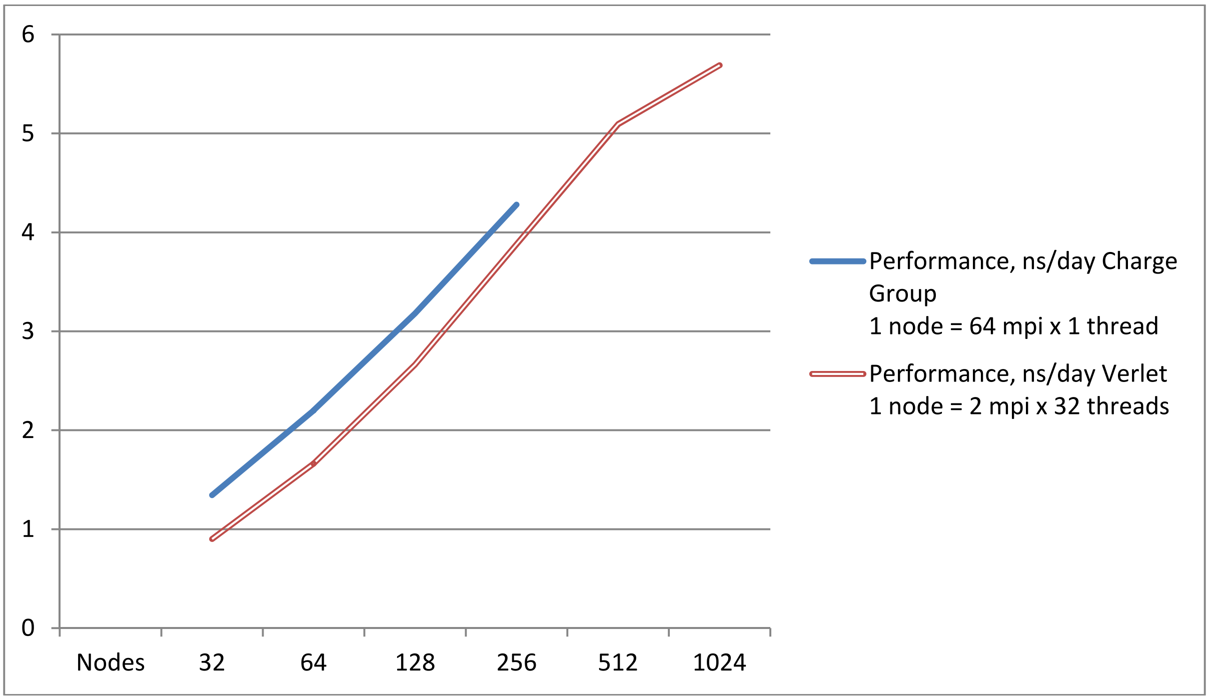

Figure 14: Scalability curves Gromacs

The results are in accordance with the expectations extrapolated from the data on the Gromacs web-site (http://www.gromacs.org/Documentation/Cut-off_schemes) stating that the group scheme is a little bit faster.

Conclusions

The outcome of these experiments shows that there are subtle problems in the PME non-bonded scheme in Gromacs 4.6.3 that prevent it from dealing with the system under test. Further work is needed to identify the problems that cause this behaviour. The problems might be in selecting the proper compilation configuration or in the code itself. When using the reaction field non-bonded scheme the package shows the expected performance. It should be noted that Gromacs 4.6.x still does not include Blue Gene/Q optimized non-bonded kernels, but these will be added soon by the project developers. Despite this, having in mind that the shortcomings are only temporary, the package is quite suitable for Tier-0 systems when configured properly and with the soon-expected additions.

3.6.2 *Enabling Xnavis (URANS solver for fluid-dynamics) for massively parallel simulations of wind farms*

Project leader:	Riccardo Broglia, CNR-INSEAN Italy
PRACE experts:	Soon-Heum Ko, SNIC-LiU, Francesco Salvadore, CINECA
PRACE facility:	FERMI

Research field:	Computational Fluid Dynamics
Application code:	Xnavis
PA number:	2010PA1461

Project objectives

This project aims to extend the capabilities of the Xnavis software, an unsteady RANS (Reynolds-Averaged Navier-Stokes) based solver developed by the research group of CNR-INSEAN. The baseline code is capable of solving the complex flow field through the employment of multi-block overset grid approach; a dynamic overlapping grid algorithm allows the treatment of moving grids, i.e. simulations with bodies in relative motion. Whilst the baseline code incorporates a hybrid MPI/OpenMP parallelization which scales well under Tier-1 platform level (order of hundreds of cores), it necessitates further refinement for running on Tier-0 level (thousands of cores). Therefore, we propose the following details as our technical targets under this project:

1. Automation of decomposition and block-splitting pre-processor
2. MPI parallelization of overset pre-processor
3. MPI refactoring for memory minimization of Xnavis solver in moving body condition
4. Implementation of parallel I/O strategy
5. Scalability analysis (before and during the implementation of the other tasks)

Optimization work and preliminary results

First, we directed our effort towards a preliminary scalability analysis (task 5.). The results of a weak scalability analysis are encouraging but the size of the cases is limited because of the overset pre-processing stage which is currently serial and memory demanding (all MPI processes must allocate the entire grid). Moreover, the manual load balancing procedure is not adequate for high number of cores. Hence, we started devising and implementing the automatic block splitting and assigning pre-processor procedure (task 1.). The algorithm has been implemented in Fortran 2003 (using a tree structure for blocks) mainly by a collaborator of the PI but we strongly supported him suggesting the algorithm and even programming sections of the code. The code has been completed but additional tasks and more detailed tests may be adequate.

Outlook

We plan to improve the scalability analysis by adding results using the new load balancing pre-processor in order to test its functionalities and to assess the best working conditions (e.g. unbalancing tolerance). Moreover, two additional intermediate pre-processors have to be studied in the context of Tier-0 sizes. If needed, partial restructuring of these tools have to be planned and possibly executed.

The tasks 2 and 3 are the most demanding and, as the task 1, require a close cooperation among the PRACE support, the PI and his collaborators. We plan to devise a memory-saving procedure for overset pre-processor as well as for the main solver and to parallelize the overset pre-processor. The implementation of the algorithm is very demanding and the full completion during the PRACE preparatory project is not guaranteed at the moment.

As for the task 4, the baseline code accesses distinct file(s) from each MPI rank, which induces a significant overhead to the file server. The parallel I/O functionality through a ROMIO implementation will relieve the overhead of loading/storing multiple files and ease the management of simulation output. That will be accomplished in a two-month time period by the PRACE support experts.

3.6.3 *Scalability analysis, OpenMP hybridization and I/O optimization of a code for Direct Numerical Simulation of a real wing*

Project leader:	Matteo Bernardini, University of Rome La Sapienza
PRACE experts:	Maciej Cytowski, ICM
PRACE facility:	FERMI
Research field:	Engineering and energy
Application code:	Direct Numerical Simulation of a real wing code (no name)
PA number:	2010PA1454

Project objectives

The application code was already compiled and executed on FERMI IBM Blue Gene/Q. Preliminary scalability results are available. The code is MPI parallelized and to improve scalability it needs to be additionally parallelized with OpenMP. This is the main objective of the project. The first step is to develop a hybrid MPI/OpenMP version of the code. The second step is to test its performance on FERMI system.

Additional topics for collaboration will be discussed after OpenMP parallelization will be ready and tested. One of the ideas is to look at parallel visualization tools (and implementation) on FERMI system.

Optimization work and preliminary results

Initially the code was analysed with the use of Scalasca toolset. OpenMP parallelization is done in 70% of the code. The OpenMP performance achieved up to date is very good.

The PI is very responsive and helpful. There were four Skype teleconferences so far.

Outlook

OpenMP parallelization needs to be finished. After this step we will perform many scalability tests on FERMI. One of these tests will be a scalability improvement test where we will look at pure MPI scalability and compare it to a new MPI/OpenMP version.

Currently we are planning to have a look at parallel visualization techniques on IBM Blue Gene/Q.

3.6.4 *Next generation pan-European coupled Climate-Ocean Model - Phase 1 (ECOM-I)*

Project leader:	Jun She, Danish Meteorological Institute
PRACE experts:	Mikael Rännar, SNIC-UmU, Maciej Szpindler, ICM
PRACE facility:	FERMI, CURIE
Research field:	Earth Sciences and Environment
Application code:	HBM
PA number:	2010PA1470

Project objectives

ECOM (Next generation pan-European coupled Climate-Ocean Model) aims to design, implement, and optimize the computing performance of a coupled climate-ocean model for both operational marine forecasting and regional climate modeling on a pan-European scale. Main performance bottlenecks are associated with the MPI communication which has not been optimized at all. There is also a need for scalability tests in the end.

Optimization work and preliminary results

The work optimizing the MPI communication has started. The first step is to try to use more non-bocking communication. Identification of routines suitable for a rewrite to have a choice of using blocking or non-blocking communications is ongoing and some routines have been rewritten. There are no real results yet, since more routines will have to be rewritten in order to utilize the non-blocking communication advantages.

In parallel, enabling of the code for Blue Gene/Q system is being performed. The main target is OpenMP performance and scalability. The preferred parallelization model for HBM code is to introduce 2-level decomposition, MPI-based parallelism between the Blue Gene/Q nodes, and to maximize thread utilization within the node (up to 64 OpenMP threads). Initial porting attempts disclosed problems with a native Blue Gene/Q compiler (IBM XL Fortran) and its OpenMP realisation. The application has been successfully compiled and tested in a serial mode. Both file I/O and performance problems occurred with OpenMP mode. I/O related problems have been partially resolved while performance of the OpenMP version is still disqualifying. Further work with OpenMP will be continued to improve performance and enable scalability across the large number of Blue Gene/Q nodes for real scale scenarios.

Outlook

The project will continue till the end of PRACE-2IP and also into PRACE-3IP. The work with transformation to non-blocking communication will continue. There are also some other ideas if this is not enough. Finally there will also be some scalability tests.

The OpenMP version for Blue Gene/Q will be improved for stable code behaviour on this platform and a better performance to enable scalability with a hybrid parallelization with MPI+OpenMP.

3.6.5 *Increasing the QUANTUM ESPRESSO capabilities II: towards the TDDFT simulation of metallic nanoparticles*

Project leader:	Arrigo Calzolari, CNR-NANO Istituto Nanoscienze
PRACE experts:	Carlo Cavazzoni, CINECA
PRACE facility:	HERMIT
Research field:	Material Science
Application code:	Quantum ESPRESSO[3]
PA number:	2010PA0633

Project objectives

In this project we plan to implement novel strategies for reducing the memory requirements and improving the weak scalability of turboTDDFT, which is a planewave pseudopotential TDDFT (Time Dependent Density Functional Theory) code, included in the Quantum ESPRESSO (QE) package. The final goal is to obtain a net improvement of the code capabilities and to be able to study the plasmonic properties of metal nanoparticle (Ag, Au) and their dependence on the size of the system under test.

Optimization work and preliminary results

Before starting the enabling of the TDDFT QE kernel, as a first step in the project, we have ported the whole QE package on HERMIT and then we started to optimize the setup (compiler suite, libraries, MPI+OpenMP execution parameters) for the bulldozer AMD architecture, considering a general purpose (not specific for TDDFT) dataset.

In porting QE on HERMIT one of the main issues was related to the optimization of the hybrid MPI/OpenMP execution, which is mandatory for running large datasets. Using the default execution environment we observed that the wall clock time of the hybrid run was higher than the CPU time, which is the opposite of the expectation. This was confirmed by a benchmark run we have done to compare (under the same condition of Task/Threads) HERMIT and FERMI. The dataset is 256 water molecules and the benchmark has been executed using 256 tasks and 8 threads per task. The result is as follow: FERMI needs 117 seconds; HERMIT needs 360 seconds. It was like if the threads were not used at all, even if in the code output the correct number of threads were reported. This made us think that the problem could have been in the threads binding/affinity behaviour of the default environment. We directed our attention on the different possibilities the CRAY launcher command (aprun) offers to control such parameters, and we finally find out that we had to specify the flag: "-cc numa_node" to get the right thread placement. This solved our performance issues and allowed us to start investigating the scalability of the TDDFT kernel with the new parallelization scheme. The new performance number with the correct thread placement for the 256 water molecules benchmark is that HERMIT needs 120 seconds, which is close to the FERMI result and matches our expectations.

Outlook

The next step will be to work with the TDDFT dataset and improve the scalability and the performance of the TDDFT kernel. This will be done mainly extending the level of parallelization implemented in the TDDFT kernel. In practice this means implementing OpenMP (already implemented in main QE kernels but not yet in TDDFT) and adopting new parallelization scheme derived from other QE kernels (where it has already been implemented).

3.6.6 *Scalability of gyrofluid components within a multi-scale framework*

Project leader:	Bruce D. Scott, Max-Planck-Instutute for Plasma Physics, Euratom Association
PRACE experts:	Luis Fazendeiro, SNIC-CHALMERS, A. Karmakar, V. Weinberg, GCS-LRZ
PRACE facility:	JUQUEEN, SuperMUC
Research field:	Engineering and Energy
Application code:	GEM
PA number:	2010PA1505

Project objectives

The main goal of the project is to improve the parallel scalability of GEM, a 3D gyrofluid code. The code uses the electromagnetic gyrofluid model which is a superset of magnetohydrodynamic and drift-Alfvén microturbulance and also includes several relevant kinetic processes. GEM can be used with different geometries depending on the targeted use case, and has been proven to show good scalability when the computational domain is distributed amongst two dimensions. Such a distribution allows grids with sufficient size to describe small scale devices. In order to enable simulation of very large tokamaks (such as the next generation nuclear fusion device ITER (International Thermonuclear Experimental Reactor) in Cadarache, France) the third dimension has to be parallelized and weak scaling has to be achieved for significantly larger grids.

Optimization work and preliminary results

So far we have successfully ported the application to both Tier-0 systems. We started first instrumented runs using tools such as Scalasca and Intel Tracing Tools to investigate the communication structure of the code and isolate incorrect or inefficient MPI programming.

Outlook

The following steps will be performed in the remaining time for the project:

- detailed investigation of the scalability,
- detection of performance bottlenecks,
- improvement of the MPI communication, especially the nearest-neighbour communication,
- parallelization of the third dimension and detailed multidimensional scaling-analysis,
- system-specific optimization on SuperMUC and JUQUEEN.

3.6.7 *Direct numerical simulation of a high-Reynolds-number homogeneous shear turbulence*

Project leader:	Javier Jimenez, Universidad Politecnica Madrid
PRACE experts:	Jeroen Engelberts, SURFsara, Vergard Eide, NTNU
PRACE facility:	JUQUEEN
Research field:	CFD, Turbulence
Application code:	SHEAR
PA number:	2010PA1492

Project objectives

The SHEAR code has been run on smaller systems. The idea is to scale it up to large systems, like the Blue Gene/Q JUQUEEN in Jülich. Three bottlenecks have been identified preventing the scaling to larger systems:

1. I/O – initially the software uses serial writing of the data. Applicability of HDF5 will be tested and implemented.
2. Communication – at some point during the calculation the data needs to be transposed. This is implemented via an ALLTOALL MPI mechanism. The question is whether this can be scaled up to many cores – or if another mechanism needs to be looked after.
3. Computation – the fast Fourier transforms are currently implemented through a self-written algorithm. The use of the generic library, FFTW3, needs to be investigated and compared to their home made mechanism.

Optimization work done and preliminary results

Siwei Dong (the programmer of the project) prefers to work on the program SHEAR, a Fortran 90 hybrid program (uses both MPI and OpenMP), himself. Besides, he makes use of small programs he calls “toy code” to try out things and to debug problems together with the PRACE collaborators. Once in a while Jeroen Engelberts makes a copy of this directory via rsync and helps with problems, either directly or together with the helpdesk of FZJ.

Current status:

1. Siwei has made a first HDF5 implementation that seems to scale up to 256 nodes. Using more cores performance decreases dramatically. Currently, the HDF5 expert at FZJ is looking into this issue.
2. Siwei has performed a series of benchmark tests with ALLTOALL communication and is currently in discussion with FZJ whether or not an ALLTOALL would scale, or not.
3. No effort has been invested in the Fast Fourier Transform part yet.

Outlook

Jeroen Engelberts (SURFsara) and Vegard Eide (NTNU) have agreed that during the first part Jeroen would mainly help/support Siwei, while Vegard had other obligations. Later on Vegard will take over from Jeroen, who will be less available during the end of the project.

Hopefully, the HDF5 will need only little more tuning. After that the focus can be completely shifted to the ALLTOALL problem and the implementation of FFTW3.

3.6.8 *Massively Parallel Multiple Sequence Alignment Method Based on Artificial Bee Colony*

Project leader:	Dr. Plamenka Borovska, Technical University of Sofia
PRACE experts:	Alexandra Charalampidou, Pavlos Daoglou, GRNET
PRACE facility:	JUQUEEN
Research field:	Bioinformatics
Application code:	MSA_BG
PA number:	2010PA1467

Project objectives

The huge amount of biological sequences accumulated in the world nucleotide and protein databases requires efficient parallel tools for structural genomic and functional analysis.

The aim of the project is optimizing and investigating the parallel performance and efficiency of an innovative parallel algorithm MSA_BG for multiple alignments of biological sequences, which is highly scalable and locality aware. The MSA_BG algorithm is iterative and is based on the concept of Artificial Bee Colony metaheuristics and the concept of algorithmic and architectural spaces correlation.

Optimization work and preliminary results

The code under study has been parallelized with MPI. Preliminary benchmark tests have been performed by the PI on the JUQUEEN supercomputer, but the initially assigned computer time has already been consumed by these tests. Still, the submission of new jobs on JUQUEEN is allowed though with lower priority.

Limited contacts with the PI have so far been an obstacle for the progress of this project. The first extensive discussion with the PI on the project objectives took place on July 3rd, 2013. Final conclusions on the work plan have not been yet decided upon.

Outlook

The current work plan (which is still under discussion) includes:

- the optimization of the code with the usage of OpenMP threads,
- the execution of benchmark tests on JUQUEEN and
- gathering of profiling and tuning performance metrics.

4 Summary

This deliverable outlines the work of Task 7.1 in PRACE-2IP. The task successfully performed five cut-offs for preparatory access including the associated review process and the support for the approved projects.

In total 24 Type C Preparatory Access projects have been supported by T7.1. The timeline of these projects is shown in the Gantt-chart in Figure 15. The chart shows the run time of each project with start date and end date in PRACE-2IP. PRACE-2IP took over responsibility for running PA C projects from PRACE-1IP in June 2012, the hand-over to PRACE 3IP will take place on August 31, 2013 (for details on both hand-over times see also section 2.5). Usually the projects run for six months. Some projects received a prolongation; this is possible on special request, mainly in case of technical system problems. During PRACE-2IP it has been decided that a prolongation should not exceed one month. The reason for the slightly different start dates within one cut-off is that each hosting member finally decides on the exact start date of the projects at their local site. Also the PI can choose a slightly later start date.

For the finalized PA C projects eight white papers were created and published. The detailed optimization work has been described in this deliverable in section 3.

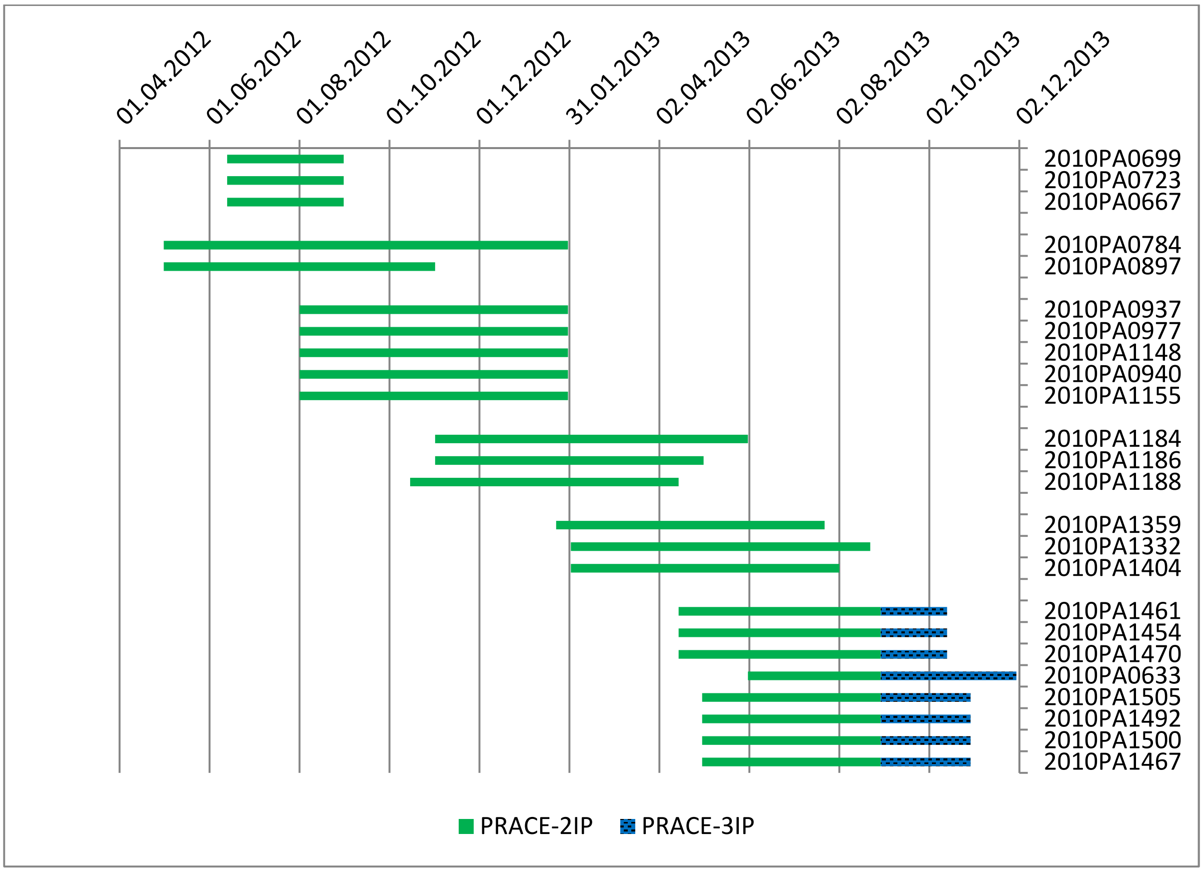

Figure 15: Timeline of the PA C projects

Table 6 gives a short overview of the results of the finalized PA C project in terms of suitability for Tier-0 after project end. Only one project was delayed as described in section 3.4.1. Two projects were mainly supported in code profiling and detection of bottlenecks so that a strategy for further improvements could be identified. For all remaining projects the code performance could be improved, half of them are considered to be able to run on Tier-0 systems. In the third column it is listed which project announced to plan an application for regular access now or already got successfully access.

A further goal which could be reached besides Tier-0 suitability is also the porting and optimizing for new system architectures, like GPUs.

PA project number, section in the document	**Result**	**PRACE Tier-0 regular access**
2010PA0699, 3.1.1	Tier-0 suitable	Regular project running
2010PA0723, 3.1.2	Tier-0 suitable	Regular project running
2010PA0667, 3.1.3	Performance improved	
2010PA0784, 3.2.1	Performance improved	
2010PA0897, 3.2.2	Tier-0 suitable	
2010PA0937, 3.3.1	Tier-0 suitable	Regular project running
2010PA0977, 3.3.2	Tier-0 suitable	Regular project running
2010PA1148, 3.3.3	Performance improved	
2010PA0940, 3.3.4	Performance improved	
2010PA1155, 3.3.5	Performance improved	Regular proposal planned
2010PA1184, 3.4.1	Delayed due to compiler bug	
2010PA1186, 3.4.2	Tier-0 suitable	Regular project running
2010PA1188, 3.4.3	Tier-0 suitable	Regular proposal planned
2010PA1332, 3.5.1	Successful code profiling	
2010PA1359, 3.5.2	Tier-0 suitable	Regular proposal planned
2010PA1404, 3.5.3	Successful code profiling	
2010PA1500, 3.6.1	Tier-0 suitable	

Table 6: Results of finalized PA C projects

In summary, T7.1 successfully achieved the envisaged objectives and enhanced the performance of the applications under consideration. This can also be concluded from the fact that from 13 fully finalized PA C projects in PRACE-2IP (projects from the cut-offs December 2012 and March 2013 so far could not yet apply for regular access, because they just ended in July 2013 or are still running) up to now seven projects announced that they plan to apply for regular access now. From these projects five have already been able to apply successfully for regular PRACE access and are now running their production projects on the Tier-0 systems.

The Preparatory Access to Tier-0 services and especially the support by PRACE experts are invaluable for the scientists to use and effectively exploit the resources of PRACE.