SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

INFRA-2007-2.2.2.1 - Preparatory phase for 'Computer and Data Treatment' research infrastructures in the 2006 ESFRI Roadmap

PRACE

Partnership for Advanced Computing in Europe

Grant Agreement Number: RI-211528

D7.5.2
Technical Requirement for the second PetaFlop/s systems(s) in 2009/2010

Final

Version: 1.0
Author(s): Jonathan Evans, BSC
Date: 23.11.2009

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: RI-211528**	
	Project Title: Partnership for Advanced Computing in Europe	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: D7.5.2	
	Deliverable Nature: DOC_TYPE: Report	
	Deliverable Level: PU *	**Contractual Date of Delivery:** 30 / November / 2009
		Actual Date of Delivery:
	EC Project Officer: Maria Ramalho-Natario	

* - 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: Technical Requirement for the second PetaFlop/s systems(s) in 2009/2010**	
	ID: D7.5.2	
	Version: 1.0	**Status:** Final
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2003	
	File(s): D7.5.2.doc	
Authorship	**Written by:**	Jonathan Evans, BSC
	Contributors:	Giovanni Erbacci, CINECA Jean-Philippe Nominé, CEA François Robin, GENCI Norbert Meyer, PSNC Marek Zawadzki, PSNC Olli-Pekka Lehto, CSC Thomas Boenisch, HLRS Stefan Wesner, HLRS
	Reviewed by:	Michael Stephan, FZJ Dietmar Erwin, FZJ
	Approved by:	Technical Board

Document Status Sheet

Version	Date	Status	Comments
0.1	31/August/20009	Draft	First draft
0.2	31/October/2009	Draft	Enhanced requirements, first sizing values entered, application to architecture mapping added.
0.3	9/November/2009	Draft	Completion of full draft.
0.4	12/November/2009	Draft	Added comments from WP7 list.
1.0	23/November/2009	Final version	

Document Keywords and Abstract

Keywords:	PRACE, HPC, Research Infrastructure, Petaflop, Technical Requirement, Procurement
Abstract:	The PRACE project has the overall objective of preparing for the creation of a persistent pan-European HPC service. Work package 7 within PRACE is titled "Petaflop/s Systems for 2009/2010" and is responsible for providing technical information to the Management Board to facilitate selection of Petaflop/s production systems in 2009/2010. Task 7.5 in WP7 is responsible for drafting the technical requirements for Petaflop/s systems that will be used in the procurement process. The key objective of this deliverable is to provide a toolbox of technical elements that can be used flexibly in the procurement of systems for the PRACE research infrastructure. This work complements the procurement process template approach that is being developed by Task 7.6 in WP7. This document is the second iteration of the task.

Copyright notices

Table of Contents

List of Tables

References and Applicable Documents

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

[2] PRACE FP7-Infrastructures-2007-1 Construction of new infrastructures.

[3] Software Environment Management http://modules.sourceforge.net/

[4] GSI-SSH http://www.globus.org/toolkit/docs/4.0/security/openssh/

[5] lperf network monitoring http://dast.nlanr.net/projects/Iperf/.

[6] *GridFTP* http://www.globus.org/toolkit/docs/4.0/data/gridftp/.

[7] *Inca: user level grid monitoring* http://inca.sdsc.edu/drupal/

[8] Initial recommendation for the selection of prototypes and first estimates of costs of PetaFlop/s class systems, PRACE Deliverable D7.1.1, March 2008

[9] Report on systems compliant with user requirements, PRACE Deliverable D7.2, April 2008

[10] Technical component assessment and development report, PRACE Deliverable D8.3.1, September 2009

[11] Preliminary report on application requirements, PRACE Deliverable D6.2.1, March 2008

[12] Final report on application requirements, PRACE Deliverable D6.2.2, September 2008

[13] Final assessment of PetaFlop/s systems to be installed in 2009/2010, PRACE Deliverable D7.1.3, June 2009

[14] Report of installation requirements and availability at European sites, PRACE Deliverable D7.3, November 2008

[15] Report on available performance analysis and benchmark tools, PRACE Deliverable D6.3.1, November 2008

[16] Report on evaluation criteria and acceptance tests for procurement, PRACE Deliverable D7.6.3, December 2009

[17] Report on deployment of enhanced software stack to selected sites for distributed systems management, PRACE Deliverable D4.2.2, June 2009

[18] Procurement Strategy, D7.6.1, December 2008

[19] Final benchmark suite, PRACE Deliverable D6.3.2, December 2009

[20] J. Reetz, Th. Sodderman, B. Heupers, J. Wolfrat, “Accounting Facilities in the European Supercomputing Grid DEISA”, GeS2007. Available online at: http://www.ges2007.de/fileadmin/papers/jreetz/GES_paper105.pdf

[21] UNICORE, http://www.unicore.eu/

[22] www.top500.org

[23] http://www.accessgrid.org/

[24] Report on petascale software libraries and programming models, PRACE Deliverable D6.6, October 2009

[25] Final technical report and architecture proposal, PRACE Deliverable D8.3.2, December 2009

[26] Technical Requirement for the first PetaFlop/s system(s) in 2009/2010, PRACE Deliverable D7.5.1, November 2008

[27] Installation report prototype systems, PRACE Deliverable D5.1.2, December 2008

[28] Technical assessment report of prototype systems, PRACE Deliverable D5.2, December 2009

[29] Assessment report on communication and I/O infrastructure of prototype systems, PRACE Deliverable D5.3, December 2009

[30] Preliminary assessment of PetaFlop/s systems to be installed in 2009/2010, PRACE Deliverable D7.1.2, November 2009

List of Acronyms and Abbreviations

General terms

BSC	Barcelona Supercomputing Center (Spain)
CEA	Commissariat à l'Energie Atomique (represented in PRACE by GENCI, France)
CINECA	Consorzio Interuniversitario, the largest Italian computing centre (Italy)
CINES	Centre Informatique National de l'Enseignement Supérieur (represented in PRACE by GENCI, France)
CSC	Finnish IT Centre for Science (Finland)
CSCS	The Swiss National Supercomputing Centre (represented in PRACE by ETHZ, Switzerland)
EPCC	Edinburg Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
ESFRI	European Strategy Forum on Research Infrastructures; created roadmap for pan-European Research Infrastructure.
ETHZ	Eidgenössische Technische Hoschule Zuerich, ETH Zurich (Switzerland)
FZJ	Forschungszentrum Jülich (Germany)
HLRS	Höchstleistungsrechenzentrum Stuttgart, High Performance Computing Centre 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
IT	Information Technology
KTH	Kungliga Tekniska Högskolan (represented in PRACE by SNIC, Sweden)
LRZ	Leibniz Supercomputing Centre (Garching, Germany)
MEAT	Most Economically Advantageous Tender
PRACE	Partnership for Advanced Computing in Europe; Project Acronym.
PSNC	Poznan Supercomputing and Networking Centre (Poland)
RFI	Request for Information
RI	Research Infrastructure
SARA	Stichting Academisch Rekencentrum Amsterdam (Netherlands)
SLA	Service Level Agreement between vendor and IT equipment owner, covering level of support, time to fix, acceptable down time etc.
SNIC	Swedish National Infrastructure for Computing (Sweden)
STFC	Science and Technology Facilities Council (represented in PRACE by EPSRC, United Kingdom)
TCO	Total Cost of Ownership. Includes the costs (personnel, power, cooling, maintenance) in addition to the purchase cost of a system.
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.
WP2	PRACE Work Package 2 - Organisational Concepts of the Research Infrastructure
WP4	PRACE Work Package 4 - Distributed systems management
WP5	PRACE Work Package 5 - Deployment of prototype systems
WP6	PRACE Work Package 6 - Software Enabling for PetaFlop/s Systems
WP7	PRACE Work Package 7 - PetaFlop/s systems for 2009/2010
WP8	PRACE Work Package 8 - Future PetaFlop/s computer technologies beyond 2010

Technical terms

B	Byte = 8 bits
BLAS	Basic Linear Algebra Subprograms
CAF	Co-Array Fortran
Cell BE	Cell Broadband Engine
CFD	Computational Fluid Dynamics
CN	Compute Node. Refers to the collection of processing units controlled by one operating system instance or one Single System Image (SSI).
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture (NVIDIA)
CVS	Concurrent Version System
DFT	Density Functional Theory
DP	Double Precision, i.e. 64 bits
FFT	Fast Fourier Transform
Flop	Floating point operation (usually in 64 bits, i.e. DP)
Flop/s	Floating point operation per second (usually in 64 bits, i.e. DP)
FPGA	Field Programmable Gate Array
GByte	Giga (= 2^{30} ~10^{9}) Bytes (= 8 bits), also GByte
GByte/s	Giga (= 10^{9}) Bytes (= 8 bits) per second, also GByte/s
GCC	GNU Compiler Collection
GPGPU	General-Purpose computing on GPU
GPU	Graphical Processing Unit
HMPP	Hybrid Multi-core Parallel Programming (CAPS)
I/O	Input/Output
KByte	Kilo (= 2^{10} ~10^{3}) Bytes (= 8 bits), also KB
LAPACK	Linear Algebra PACKage
LDAP	Lightweight Directory Access Protocol
MByte	Mega (= 2^{20} ~10^{6}) Bytes (= 8 bits), also MB

MByte/s	Mega (= 10^6) Bytes (= 8 bits) per second, also MB/s
MD	Molecular Dynamics
MMT	Massively Multithreaded
MPI	Message Passing Interface
MPP	Massively Parallel Processing (or Processor)
MTBI	Minimum meantime Between Interrupts
NIC	Network Interface Controller
NUMA	Non-Uniform Memory Access or Architecture
OS	Operating System
PByte	Peta (= 2^{50} ~10^{15}) Bytes (= 8 bits), also PB
PetaFlop/s	Peta (= 10^{15}) Floating point operations (usually in 64 bits, i.e. DP) per second, also TF/s = 1000 TF/s
PGAS	Partitioned Global Address Space
Processing Unit	A processing core plus accelerators.
QCD	Quantum ChromoDynamics
RAID	Redundant Array of Inexpensive Disk
Rmax	Maximum performance of a computer in GFlop/s, as used in Top500
SDK	Software Development Kit
SHMEM	Share Memory access library (Cray)
SLA	Service Level Agreement
SMT	Simultaneous MultiThreading
TByte	Tera (= 2^{40} ~10^{12}) Bytes (= 8 bits), also TB
TCP/IP	Transmission Control Protocol/Internet Protocol
TDFDT	Time-Dependent Density Functional Theory
TFlop/s	Tera (= 10^{12}) Floating point operations (usually in 64 bits, i.e. DP)) per second, also TF/s
UPC	Unified Parallel C
UPS	Uninterruptible Power Supply

Executive Summary

The PRACE project has the overall objective of preparing for the creation of a persistent pan-European HPC service. Work package 7 within PRACE is titled "Petaflop/s Systems for 2009/2010" and is responsible for providing technical information to the Management Board to facilitate selection of the Petaflop/s production systems in 2009/2010.

This document is the second deliverable from WP7 task 5 *Drafting of Technical Requirements* and provides an update of D7.5.1 [26] based on the work undertaken in the technical work packages of PRACE during 2009. It is designed as a flexible toolbox of technical elements that can be utilised when preparing procurements where the elements used will be dependant on the procurement process and target system. The document supports the PRACE Management Board in selecting the production PetaFlop/s Tier-0 systems under different funding models.

The approach to presenting requirements has continued the method introduced in D7.5.1 [26]. The model for PRACE is to have an infrastructure of 3-5 Tier-0 Petaflop/s systems with a variety of system architectures to support the demands of the key application codes and research areas identified in WP6. This approach of application led procurement, adopted in PRACE, ensures that the investment in applications of the computational simulation research community is protected as they move to the Petascale regime.

A second key input into the document is the assessment of the WP7-WP5 PRACE prototypes using synthetic benchmarks in WP5. This allows the performance of separate parts of the systems to be measured and feeds directly into the technical requirements. The experience gained in using a standard set of benchmarks is advantageous as it can be supplied to vendors to allow realistic performance figures to be prepared and can be used in acceptance tests for delivered systems.

It is recommended that the work undertaken here be continued in the follow up project to support the implementation phase of the PRACE Research Infrastructure. The technical requirements in this document are a snapshot in time and will need to be updated as user needs evolve and vendor offerings change through new technologies and changes in market conditions. A process of continuous improvement to the document should be established to capture new best practice and requirement values, based on Tier-0 and Tier-1 procurements. The lessons learnt should be fed into the toolbox of technical elements started here to provide a valuable resource for the European HPC community.

1 Introduction

This chapter provides an introduction and sets out the objectives and scope of this document.

1.1 Background and Purpose

The PRACE project has the overall objective of preparing for the creation of a persistent pan-European HPC research infrastructure (RI). Work package 7 within PRACE is titled "PetaFlop/s Systems for 2009/2010" and is responsible for providing technical information to the Management Board to facilitate selection of the PetaFlop/s production systems in 2009/2010.

Task 7.5 in WP7 is responsible for drafting the technical requirements for PetaFlop/s system(s) that will be used in the procurement process. This task is iterated twice.

The first iteration provided technical requirements based on information gathered during the first 10 months of the PRACE project and documented in D7.5.1 [26]. It provided requirements for major HPC architectures, which were already available or considered likely to be available in 2009/2010, without giving preference to a particular architecture. The first Petaflop/s system in Europe, an upgrade to Jugene, has now been installed at FZJ and is likely to form one of the first Petaflop/s machines in the permanent Research Infrastructure. More details are available in Section 7.3.

The second iteration, which forms this document, updates these requirements with new information provided by HPC vendors and experience gained from the technical activities in PRACE Work Packages 4 to 8 including the application and prototype evaluations. This document will be available to support the procurement of the second (multi-) PetaFlop/s system(s).

1.2 Objectives

The objectives of this document are:

1. To support the PRACE Management Board in selecting the second production PetaFlop/s systems by providing a consistent approach for procuring PRACE PetaFlop/s system(s).
2. To provide a toolbox of technical elements which can be utilised when preparing procurements. The elements used will be dependant on the procurement process and target system.
3. To provide a suggested process for the technical assessment within a procurement.
4. To present the information in a format that will support the addition of new architectures as they become available.
5. To encourage continuous improvement of the document by distilling best practice from PRACE procurements.

1.3 Scope

The requirements are presented without limiting which procurement process is to be followed, for example open, restricted or pre-commercial [18]. Whilst a decision on the PRACE funding model is ongoing Tier-0 systems may be procured and hosted by national centres or by a permanent PRACE Research Infrastructure (PRACE RI).

Technical requirements are presented without preference for a particular architecture. It is for the management board to select an appropriate mix of architectures to support the needs of European researchers in the future PRACE RI.

Technical requirements are presented without preference for a specific vendor solution.

The following technical requirements are within the scope of this document:

1. hardware including systems architecture and sizing,
2. I/O performance and global storage sizing, internal and external to the system,
3. post processing and visualisation,
4. software including operating system, management and programming environment,
5. operational requirements including installation constraints,
6. maintenance and support requirements,
7. training and documentation requirements,
8. delivery requirements.

The technical requirements are presented in one of two groups:

1. Technical requirement ratios, such as memory per compute node, are designed to leave open the way future procurements are organised. This allows a procurement to start with a fixed budget and seek to acquire the best performance for the available budget or seek the lowest price for a fixed performance. These are provided on a per architecture basis.
2. All other technical requirements are presented as a checklist with example values for clarity. A specific procurement will decide which are relevant and where appropriate tailor the requirement values and target them at a specific installation site.

The document complements D7.6.3 [16] and includes processes for technical assessment.

The document demonstrates how it may be used in practice by applying to completed PRACE procurements.

1.4 Audience

The intended audience will be both technical (for example HPC researchers, operations staff and HPC vendors) and non-technical readers.

1.5 Document Structure

The document is split into 6 chapters plus an appendix:

Chapter 1 provides this introduction.

Chapter 2, *Requirement Methodology*, explains the requirements gathering process and makes connections to other PRACE technical tasks.

Chapter 3, *Procurement Elements*, provides information on procurement elements related to technical requirements including, evaluating vendor response, benchmarking rules and how total cost of ownership (TCO) is being calculated.

The *System Sizing Requirements* in Chapter 4 defines the key requirements and add values for each of the available architecture classes.

The *Check List of Technical Requirements* in Chapter 5 provides a comprehensive list of technical requirements to be considered when preparing procurements.

Chapter 6 summarises the document and indicates the next steps to be taken.

The Appendix provides supporting information relating to these requirements and covers:

- HPC architecture definitions,
- Application benchmark suite,
- PetaFlop/s procurements in Europe,
- Existing Systems Analysis.

2 Requirement Methodology

The gathering of requirements for the next PetaFlop/s systems has, not surprisingly, strong dependencies with the technical work packages in PRACE WP4 to WP8 and these are discussed in more detail below.

The key stakeholders in defining these requirements are the future users of the PetaFlop/s systems, the research scientists who are using computational simulation to advance their research. This group is represented by WP6, which amongst other activities is tasked with capturing application requirements for petascale systems and creating a representative benchmark suite of applications. The user needs drive procurement of HPC systems by identifying the types of HPC systems architecture most suited to the relevant mix of application codes. They also inform the decisions on where to spend money to get an appropriately balanced system, based on CPU, memory, message passing network and I/O. An overview of user needs is documented in Section 2.1.

When procuring leadership class systems it is important to assess and understand the alternatives offered by the market. The PRACE project has selected 6 prototypes for PetaFlop/s systems (WP7-WP5 prototypes) based on architectures and systems that are likely to form the first PRACE systems for procurement by the end of 2010. As well as being used within PRACE to assess the WP6 applications the separate parts of these systems are being assessed with a set of synthetic benchmarks as part of the WP5 activities. The results from these benchmarks feed into the technical requirements and are documented further in Section 2.2. This approach of assessing systems provides an alternative method for procurement appropriate to leadership class systems, where scientific applications are developed to take advantage of the system features.

Emerging technologies for the second wave of (multi-) PetaFlop/s systems are being assessed within work package 8. A set of prototypes (WP8 prototypes) have been selected and implemented to examine these promising technologies. Early results are discussed in Section 2.3.

Task 7.1 within WP7 has carried out three surveys of technologies, architectures and vendors for PetaFlop/s systems. The information gathered has provided the architecture classification used within this document as well as the classes of architecture that will be available to build a PetaFlop/s system in the 2009-2010 time frame. This work has been complemented with a survey of the top 15 systems in the Top500 list [22] to provide a comparison with the sizing ratios presented in this document. More information is available in Section 2.4.

There are other closely related PRACE activities that have helped to inform these requirements and these are listed in Section 2.5.

2.1 User Needs

2.1.1 *Methodology*

The user needs form a key part of the scientific and business case that are needed to justify the procurement of systems to form an improved HPC service for the research community. WP6 has assembled a representative set of application codes into a benchmark suite described in Deliverable D6.2.1 [11] then D6.2.2 [12]. This set of codes is being used to explore peta-scaling, porting and optimisation, as well as libraries and programming models. The codes are tested with runs of the benchmark suite on the PRACE WP7-WP5 prototypes and PRACE WP8 prototypes.

WP7 Task2 is in charge of the connection and translation work between WP6 and WP7. Deliverable D7.2 "Report on systems compliant with user requirements" [9], released in Month 4 (April 2008), presented an initial translation of the user requirements to architecture specifications that were derived from computational application classes for the European PetaFlop/s systems. An update to the final table presented in Deliverable 7.2 was provided in D7.1.3 [13] along with a first qualitative assessment of the main architectural parameters that constrain the benchmark applications.

This work of revisiting the applications requirements and how they can relate to computing systems architecture and sizing is evolving, based on the experience on the applications selected and the input received from the ongoing running of these applications on the WP7-WP5 and WP8 prototypes.

A new update to this table is now reported as part of this Deliverable D7.5.2, Table 1 in the next subsection. This updated version is based on the second set of runs for the Task 5.4 activity that is used to inform the deliverables D6.4 and D6.5 as well as the WP8 benchmarking in D8.3.1 [10].

In June 2009, following recommendations from the EU review in March 2009, the PRACE Application Benchmark Suite was updated.

The following applications were dropped: ECHAM5, VASP and SIESTA mainly for licensing problems. These applications were added: SPECFEM3D (earthquake simulation code), ELMER (multi-physics, engineering code), QuantumEspresso (electronic-structure calculations and materials modelling at nanoscale), Octopus (Density Functional Theory calculations) and WRF (Weather Research and Forecasting Model). In addition EUTERPE replaced TORB.

A total of 22 codes now form the PRACE Application Benchmark Suite and these are listed in Annex 7.2. As the work for enabling and assessing the scalability of the new codes (WRF, SPECFEM3D, ELMER, QuantumEspresso, Octopus) is still on going, for compatibility reasons we decided to update the original Table presenting only the codes selected originally.

As in the previous versions, Table 1 reports in each row a different benchmark code and in each column a different architecture class, using the architecture classification detailed in section 7.1. The crossing box between row and column represents the fitting of the given benchmark code on the specific architecture class.

To express how application codes are mapped into architecture classes, a colour code is used, in order to provide an immediate visual interpretation for the reader and facilitate the perception of clustering. Three different colours are used, with the following meaning:

Green box: the corresponding application has a high match with the corresponding architecture class. This architecture is a good choice for production runs of this application in terms of single-core efficiency, scalability, memory capacity and price-performance.

Yellow box: the application has a moderate match with the architecture class. This architecture is a reasonable choice for production runs of this application, but there may be some problems in terms of single-core efficiency, scalability, memory capacity or price-performance.

Grey box: the application has a low match with the architecture class. This architecture is a poor choice for production runs of this application. There are serious problems in terms of single-core efficiency, scalability, memory capacity or price-performance. The grey colour is used also in case it is not likely to port the application to that architecture class (not useful or too difficult).

White *box:* means that no information or insufficient information is available to classify on the mapping between the application and the architecture class.

In Table 1, the column named “Accelerators” refers to the porting activity of applications on Homogeneous Clusters equipped with specific accelerators, as represented by the WP8 prototypes. The extensive testing of these prototypes is on going as part of the WP8 and WP6 activities. Some performance results have been presented in Deliverables D6.6 [24] and D8.3.1 [10] with more to be reported in D8.3.2 [25] at Month 24. The activity of exploiting accelerators is complex and until now only numerical kernels representative of computational applications (like dense and sparse linear algebra and spectral methods) have been ported. For this reason, the values for the Accelerators column are related to expected behaviour of applications in the future extrapolated from the structure of the application rather than measured performance. More comment on the assessment of WP8 prototypes is included in Section 2.3.

2.1.2 *Application to PetaFlop/s systems architectures mapping update*

	Parallel cluster systems						
	Clusters				Custom built supercomputers		
	Homogeneous Clusters			Heterogeneous Clusters	MPP		Vector
	Small Memory	Large Memory	Accelerators	Small Memory	Small Memory	Large Memory	
NAMD	🟩	🟩	🟨	🟩	🟨	🟩	▨
CPMD	🟩	🟩	⬜	🟨	🟩	🟩	🟩
VASP	🟩	🟩	⬜	⬜	🟨	🟩	🟩
QCD	🟩	🟩	🟨	🟨	🟩	🟩	🟨
GADGET (1)	🟨	🟩	⬜	⬜	🟨	🟩	⬜
Code Saturne (2)	🟩	🟩	🟨	🟩	🟩	🟩	🟩
TORB	🟩	🟩	⬜	🟩	🟩	🟩	🟩
NEMO (3)	🟨	🟨	⬜	▨	🟨	🟨	🟩
ECHAM5 (4)	🟨	🟩	▨	▨	🟨	🟨	🟩
CP2K (5)	🟨	🟩	⬜	⬜	🟨	🟩	⬜
GROMACS (6)	🟩	🟩	🟩	🟩	🟩	🟩	⬜
NS3D (7)	▨	🟨	🟩	⬜	▨	🟨	🟩
AVBP	🟩	🟩	⬜	⬜	🟨	🟨	⬜
HELIUM	🟨	🟩	⬜	⬜	🟨	🟩	🟨
TRIPOLI 4	🟩	🟩	⬜	⬜	🟨	🟨	⬜
GPAW(10)	🟨	🟩	⬜	⬜	🟨	🟩	⬜
ALYA (8)	🟩	🟩	⬜	🟩	🟩	🟩	🟨
SIESTA (9)	🟩	🟩	🟨	🟨	🟩	🟩	⬜
BSIT	🟩	🟩	🟩	🟩	🟩	🟩	▨
PEPC	🟩	🟩	▨	▨	🟨	🟩	▨

Table 1: Applications to PetaFlop/s systems architectures mapping

In the following list, we report some comments related to the behaviour of specific application codes analysed (see Section 7.2 for details about each application):

(1) **GADGET**: In some cases the code has a heavy demand on memory (bandwidth and capacity). The use of accelerated systems is still on-going for Gadget, the outcome is not yet known.

(2) **Code Saturne**: The code scales well up to 1000s of cores on several MPP platforms (scalability up to 80 TFlops/s has been measured) but the percentage of peak performance obtained is still relatively low.

Platforms with small amount of memory and lack of high-performance parallel I/O sub-systems may present a reduced suitability.
The code should perform well on vector processors due to vectorisable loops.
Parts of the code could take advantage of accelerator facilities resulting in better serial node performance. Probably more suited for Cell platforms than GPUs/FPGAs.

(3) **NEMO**: NEMO has been optimized for NEC vector platforms.
Poor scalability on cluster systems (the code scales up to 5 TFlop/s partitions). The cache is not optimally used, because of the use of large loops. Blocking is needed, but it is not clear yet exactly how. I/O is an issue especially on the IBM BlueGene.
Porting to systems with accelerator facilities and to heterogeneous cluster systems would require a lot of work. In fact the code is not limited to a small kernel that uses most of the CPU time, but spreads over tens of routines.

(4) **ECHAM 5**: The code does not scale on clusters. Large amounts of physical memory and I/O bandwidth are needed to increase performance. ECHAM5 runs well on vector platforms but the price/performance ratio is much worse than that of MPP and clusters. There is presently no porting effort for accelerators or heterogeneous clusters (code GREY) due to application complexity.

(5) **CP2K**: Performance on IBM BlueGene: the scalability, as of today, has not been confirmed by WP6. But good scalability and performance has been reported for the Jaguar Cray XT5 system at Oak Ridge, as well as on the new CSCS system. Vector and accelerated systems (code WHITE) have not been evaluated. It is possible to port CP2K to vector systems but would require a big effort.

(6) **GROMACS**: On clusters and MPP with small amount of memory (i.e. IBM BlueGene), a large number of nodes is required to get decent performance. The parallelisation does not always scale far enough. GROMACS is being ported to GPUs and heterogeneous clusters (Cell) but there are no publicly available versions yet. Typically Molecular Dynamics applications have seen speedups of up to 30 when run on GPUs. The scalability depend very much on the technique used to compute long range forces, tables are filled up considering “RF” technique (less accurate than PME).

(7) **NS3D:** N3D has been substituted by NS3D offering better performance in terms of scalability. The code achieves good performance on vector systems. For MPP systems with a small amount of memory, it is not possible to reduce the problem size (load per process) below a specific amount to fit on a single core. The code runs on clusters with a small amount of memory with reasonable performance if the problem size fits on a single core.

(8) **ALYA**: The code suffers from poor locality (indirect and irregular memory access).

(9) **SIESTA**: SIESTA uses BLAS, Lapack and FFT. All of these computational methods achieve good performance on Cell chips but the code suffers from poor locality (indirect and irregular memory access).

(10) **GPAW**: the application can perform two kinds of computation, DFT and TDDFT, where DFT is much more communication and memory intensive. Tables have been compiled considering TDDFT computation.

2.1.3 *Hints on the main parameters driving application requirements*

In order to better characterise the architectures with ontology parameters, it is important to analyse how the applications behave in terms of architecture constraints (memory-bound, CPU-bound, communication-bound). Some of these parameters were originally retrieved from D6.2.2 [12] and have now been updated with the information from the benchmark results that are available on the different architectures. In the following Table 2 we provide an updated list of the most critical parameters (memory usage, CPU usage, communication usage; I/O usage also is indicated when some information is available). Moreover we provide some updated information on the *scalability* of the codes: the scalability is expressed in terms of TFlops/; a code scales up to k TFlop/s if, running from a system configuration with k/2 TFlop/s performance, the code reaches a speed up of at least 1.6 on the k TFlop/s configuration. This measure corresponds to the upper limit to which the code can be considered scaling well, so far, on PRACE prototypes – we only keep the best result if different prototypes were tested.

It is important to underline that this table still provides very coarse qualitative information, based on the general characteristics of the codes. Most of the parameters can change as a function of the input data set selected.

For each application the Memory Usage is classified as high, medium or low. These values are related to the scalability reported in the last column of Table 2.

- *High* means that Memory is a limiting factor for the scalability. Data are a function of the problem size but the data cannot be distributed among the different processors (replicated on each processor). A high memory usage can have a negative impact on scalability as a higher scalability could require local data structures not fitting in the memory available on each node.
- *Medium* means that memory is not a constraint for the scalability of the code, up to the scalability reported in the table, but can represent a threshold for a higher scalability, This threshold can change from application to application.
- *Low* means that there are no limitations at all. The data structure can be distributed easily among the different processors

The CPU usage is classified as high, medium or low.

- *High* means a code CPU-bound. In other words, if we double the power of the processor we can also observe a doubling in performance.
- *Medium* identifies a code where the floating-point arithmetic does not represent a dominant part of the code. In other words the code is mainly characterised by integer arithmetic.
- *Low* identifies an IO-bound application.

The communication usage is *high* when the communication scheme of the application is mainly based on collective communication, so an architecture with a performant communication network is required to sustain the communication activity.

Low communication usage identifies applications with a moderate communication activity. Applications characterised by *point to point communication* schemes are typical of this class.

Medium communication usage identifies applications which need a collective communication scheme but this communication is not an issue as other limiting factors are influencing the scalability, (i.e. memory).

CODE	Memory usage	CPU usage	Communication usage	I/O usage	Scalability
NAMD	medium	high	medium (p2p)	low	20
CPMD	medium	high	high (coll.)	low	10
VASP	high	high	high (coll.)	medium	▨
QCD	low	high	low	low	160
GADGET	high	high	medium (coll+p2p)	medium	40
Code_Saturne	medium	high	medium	medium	80
TORB	▨	high	high	▨	80
NEMO	high	high	medium	high	5
ECHAM5	high	low	low (p2p)	high	▨
CP2K	high	high	high (coll.)	low	5
GROMACS	low	high	medium (p2p)	low	40
NS3D	high	high	medium	medium	20
AVBP	medium	medium	medium	low	40
HELIUM	high	medium	high (p2p)	low	20
TRIPOLI 4	▨	▨	▨	▨	
GPAW	medium	high	low	low	40
ALYA	medium	high	high (coll.)	medium	20
SIESTA	medium	high	medium	low	▨
BSIT	high	medium	medium	high	10
PEPC	medium	high	medium	medium	10

▨ ***Information unknown at this stage***

**** p2p = point to point communications; coll. = collectives***

Table 2: Qualitative behaviour of applications in terms of ontology parameters

2.1.4 *General comments on applications*

This still qualitative but updated study confirms that general-purpose architectures (MPP and homogeneous clusters) can satisfy most application requirements, in their current status.

Vector systems are suitable for a subset of applications, and a good match for applications that can suffer from limitations on MPP or cluster systems. Interestingly, we also observe an increased interest in "accelerated" systems for a significant subset of applications, still mostly estimated, since porting efforts are certainly important and will require significant time.

Due to the important scaling efforts still required for most of the applications, MPP and homogeneous cluster supercomputers already offer a wide variety of system designs for the foreseen types of applications. The objective is to tune and balance design parameters such as memory size and organization, interconnect options, compute node configuration possibly including attached processors (accelerators) etc. This already offers many degrees of freedom and seems to be the direction to take for the first round of PRACE acquisitions, while fostering more application adjustment or re-writing in the mid term in order to benefit from other future or emerging architectural options.

2.2 WP7 Prototype Assessment

Work Package 7 has been responsible for identifying PRACE prototypes to be assessed for their suitability as future systems in the PRACE permanent research infrastructure. The prototypes chosen are detailed in Table 3.

System	Location	Architecture Classification
IBM Blue Gene/P “Jugene”	FZJ, Germany	MPP (small memory, few cores, low-power)
Cray XT5 “Loviatar” and “Louhi” production system	CSC, Finland (joint proposal with CSCS, Switzerland)	MPP (larger memory, more cores)
IBM POWER6 “Huygens”	SARA, Netherlands	Homogeneous multi-core (more memory, more cores)
IBM Cell at BSC “MariCel”	BSC, Spain	Heterogeneous multi-core
NEC SX9/Nehalem x86	HLRS, Germany	Hybrid System
Bull Intel Nehalem/Xeon IB “INTI”, “Juropa”	CEA, France FZJ, Germany	Homogeneous multi-core (small memory, few cores)

Table 3: WP7 PRACE prototypes

Where as WP6 (along with WP5 Task 4) has focussed on user needs and application assessment, WP5 Tasks 2 and 3 have focussed on assessing the capability of the PRACE prototypes. These assessments have been undertaken with a set of synthetic benchmarks selected in conjunction with WP6 and detailed in D6.3.2 [19]. A synthetic benchmark is defined as a set of programs that do not represent a real application, but rather attempt to assess a particular property of a computer system in order to understand its performance.

The technical assessment report D5.2 [28] assesses the performance in these categories:

1. System performance (CPU, memory),
2. System balance (bandwidth/flops ratios),
3. Operating system performance,
4. Reliability,
5. Manageability,
6. Total cost of ownership.

The technical assessment report D5.3 [29] assesses the performance in these categories:

1. Message passing,

2. Internal I/O,
3. External I/O.

Where appropriate the scalability of the prototypes has been assessed by running benchmarks on different partition sizes.

The resulting measurements are being used within PRACE to help application code developers understand the strengths and weaknesses of the prototype architectures and to help inform the technical requirements in Chapter 4. The main ratios are presented in Table 4 to Table 6.

Prototype	Memory / processing unit (GByte)	Memory / Linpack Flop/s (Byte / Flop/s)	Memory bandwidth / Linpack Flop/s (Byte / Flop)
INTI (Homogeneous cluster, lower memory)	2 – 4	0.21 – 0.43	0.30
Huygens (Homogeneous cluster, higher memory)	4 – 8	0.39 – 0.78	0.008
MariCel (Heterogeneous cluster)	4 – 16	0.058 – 0.23	0.098
Jugene (MPP lower memory)	0.5 – 1	0.195 – 0.39	0.429
Louhi (MPP higher memory)	1 – 2	0.14 – 0.28	0.053
Baku (Vector part)	32	0.36	0.122
Baku (Homogeneous cluster, lower memory)	1.5 - 6	0.16 - 0.65	0.377

Table 4: WP5 synthetic benchmarking assessments 1 of 3

Prototype	All to all MPI bandwidth / processing unit (MByte/s)	All to all MPI bandwidth / Linpack Flop/s (Byte / Flop)
INTI (Homogeneous cluster, lower memory)	288 – 18	0.0314 – 0.0019
Huygens (Homogeneous cluster, higher memory)	144 – 77	0.0117 – 0.0076
MariCel (Heterogeneous cluster)	73 – 47	0.0007 – 0.0011
Jugene (MPP lower memory)	119 – 35	0.0457 – 0.0134
Louhi (MPP higher memory)	40 – 11	0.0057 – 0.0015
Baku (Vector Part)	1128 - 1015	0.0120 - 0.0108
Baku (Homogeneous cluster, lower memory)	18 – 5.4	0.0018 - 0.00057

Table 5: WP5 synthetic benchmarking assessments 2 of 3

Prototype	Read I/O bandwidth / processing unit (MByte/s)	Write I/O bandwidth / processing unit (MByte/s)	Linpack Flop/s / MW (TFlop/s / MW)	Linpack Flop/s / footprint (TFlop/s / m^2)
INTI (Homogeneous cluster, lower memory)	264 – 14	40 – 10	318	3.0
Huygens (Homogeneous cluster, higher memory)	55 – 6	45 - 1	80	0.39
MariCel (Heterogeneous cluster)	17 – 2	13 – 1	455	5.0
Jugene (MPP lower memory)	50 – 0.3	38 – 0.04	300	2.0
Louhi (MPP higher memory)	17 – 1.5	13 – 1.2	165	3.8
Baku (Vector Part)			48.71	
Baku (Homogeneous cluster, lower memory)	46 - 0.5	25 - 0.6	273.06	

Table 6: WP5 synthetic benchmarking assessments 3 of 3

Care needs to be taken in making unqualified comparisons between these prototypes because:

- Some are in daily production use, others are in a near production state,
- They are at different scales (values as ranges are dependant on scale, i.e. the size of the system),
- The I/O systems are designed for different levels of performance and in the case of MariCel not designed to be representative of a full scale system,
- They represent different generations of technology.

The two homogeneous cluster prototypes show good overall system balance although the new technology in the INTI and Baku (Nehalem) prototype shows better power efficiency and packaging.

The heterogeneous cluster stands out as a more experimental system with lower bandwidth / Flop ratios owing to the use of accelerators in the processing unit.

The MPP systems show good overall system balance.

The vector system shows high memory bandwidth per Linpack Flop/s.

2.3 WP8 Prototype Assessment

Work Package 8 has a goal of evaluating future multi-petascale-technology and has chosen 12 prototypes to examine the following features of HPC systems: computational accelerators, multi threaded processors, interconnects, I/O, memory, programming models and energy efficiency. The prototypes that have been selected are listed in Table 7.

Number	Prototypes	Installation Site	Targeted Components
1	eQPACE	JSC, Germany	Interconnects, Cell node performance, Low Power Consumption and high density packaging
2	RapidMind	BAdW-LRZ, Germany	Programming Models, Accelerators, Multi threaded processors

Number	Prototypes	Installation Site	Targeted Components
3	LRZ-CINES (Phase 1)	CINES, France BAdW-LRZ, Germany	SGI Altix XE, Intel Nehalem-EP,ClearSpeed-Petapath and QDR Infiniband SGI Altix ICE 8200LX, Intel Nehalem-EP, Intel Larrabee, ClearSpeed-Petapath and DDR Infiniband
4	LRZ-CINES (Phase 2)	BAdW-LRZ, Germany	SGI UV, Intel Nehalem-EX, Numalink5, Intel Larrabee and Clearspeed-Petapath
5	Hybrid Technology Demonstrator	CEA, France	GPGPU, HMPP
6	Maxwell FPGA	EPCC, United Kingdom	FPGA, Low Power Consumption and Programming Model
7	PGAS Language Compiler	CSCS, Switzerland	PGAS Programming Model
8	ClearSpeed-Petapath	SARA, Netherlands	ClearSpeed-Petapath
9	XC4-IO	CINECA, Italy	I/O and performance , File System, SSD for metadata,
10	Research on Power Efficiency	PSNC, Poland, SFTC, United Kingdom	Power consumption, porting of applications
11	PGAS Programming	CSC, Finland	Performance of two PGAS languages: UPC and CAF
12	Parallel GPU	CSC, Finland	Parallelizing CUDA, porting CUDA to OpenCL, GPGPU and their performance
13	SNIC-KTH	KTH, Sweden	Energy efficient computing

Table 7: WP8 PRACE prototypes

Task 8.3 has now produced deliverable D8.3.1 [10], which provides some preliminary findings on the benefits and drawbacks of using these technologies based on a set of benchmarks. The WP8 benchmark set is drawn from existing synthetic benchmarks, modified to use new architectures or programming models and some micro-kernels extracted from the WP6 application benchmarks.

The evaluations show that the performance observed with the accelerator based approach can be substantial but it is very much dependent on the actual match of the characteristics of the algorithm and the device. The programmability is still not ideal, especially when aiming at very high performance and non-trivial codes. However, many of the techniques to be developed for accelerated systems will be useful in optimizing the performance of these new homogeneous multi-cores in the future. We can also expect that processor vendors can eliminate one of the major bottlenecks in the use of accelerators today - the bandwidth limits between host processor and memory, and accelerator - leading to future systems that will have tightly coupled heterogeneous cores. Finding the right balance in node architecture and programming models requires significant further research and development efforts. Due to the high Linpack performance of accelerator based approaches, it is very likely that the processing cores used in future systems will be heterogeneous e.g., high performance many-core processors and one accelerator or heterogeneous many-core processors will constitute the processing elements in these systems. Hence, the programming models should be able to handle the heterogeneity that these two types of resources constitute and intelligently decide where to run the different tasks based on their bandwidth and computation needs.

Based on responses to the Task 7.1 vendor survey the first and second-generation procurements of PetaFlop/s systems are unlikely to achieve the peak performance by relying on off-chip accelerators. However there may be a requirement, for research purposes, to specify a smaller partition of the machine with accelerators. For this reason where relevant the technical requirements contain a sub-heading of optional extensions, which includes those requirements derived from the WP8 future technologies work.

Task 8.3 is due to produce deliverable D8.3.2 [25] at the end of the PRACE project which will update the findings from the evaluation of the WP8 prototypes and indicate potential architectural approaches and relevant components for next generation systems.

2.4 Vendor Survey and Existing Comparable Systems

WP7 Task 7.1 *Survey of technologies, architectures and vendors for PetaFlop/s systems to be delivered in 2009/2010* has provided deliverable D7.1.3 [13]. This document updates the market survey in D7.1.1 [8] providing bounds and ranges for potential PetaFlop/s systems in 2009/2010 and feeding into the hardware requirements including system sizing.

The deliverable produced a revised HPC architecture classification scheme, elaborated further in Section 7.1, which is used in this document to group the sizing requirements for different architectures.

The deliverable also included a chapter on the roadmaps for Petascale projects worldwide that examined both trends in the latest Top500 list [22] and different region’s roadmaps to help size future PRACE Petascale systems, so that they will have the appropriate visibility in the Top500 list. The methodology is explained in Section 5.8 of D7.1.3 [13] and involves plotting the Top 5 and Top 10 plus announced projects and then extrapolating to estimate the peak performance for the range of positions 1 to 5 and 1 to 10 in the lists. The conclusions that help to inform the Chapter 4 sizing values are included in Table 8:

Peak performance (PetaFlop/s)	TOP 5	TOP 10
November 2009	[0.5, 2]	[0.3, 2]
June 2010	[0.8, 2.7]	[0.5, 2.7]
November 2010	[1.1, 3.4]	[0.7, 3.4]
June 2011	[1.7, 4.8]	[1, 4.8]

Table 8: Estimated peak performance to be in TOP5/10 - November 2009 to June 2011

This work has been adapted for this document to examine some of the hardware sizing values and ratios, where available, of the top 15 machines from the June 2009 Top500 list [22]. Planned systems have not been included because of scarcity of sizing information. These values are compared and contrasted with the PRACE prototype benchmarking and vendor survey to help inform the sizing values entered in Chapter 4. The detailed figures are available in Section 7.4 with the following interesting results found:

1) The top 2 systems have an I/O bandwidth of approximately 200 GByte/s on 2 – 10 PByte Global storage. This is consistent with the requirement of the PRACE PetaFlop/s systems.

2) The average memory size to Flop/s rate is 0.22 Byte / Flop/s and this is a decreasing trend with increased system size. The PRACE sizing requirements are ranges that encompass this value for some of the systems. Homogeneous clusters with large memory are higher and heterogeneous clusters with a high Flop/s rate are lower.

3) The average peak performance compared to power consumption is 256 TFlop/s / MW with an increasing trend for larger machines. The sizing requirements for the PRACE machines are consistent with this figure containing a values both above and below, and an average of 228 TFlop/s / MW.

This work will have to be updated, for instance at the time of writing this report, November 2009 Top500 list had not yet been published.

2.5 Additional Dependencies

The following tasks and their associated deliverables are also related to this document.

WP7 Task 7.3 *Installation requirements for PetaFlop/s systems* provided deliverable D7.3 [14] which assesses the capability of the PRACE consortium to host and operate Peta-scale computing facilities based on vendor input of installation parameters and site surveys of existing, and planned, PRACE partner HPC sites. The parameters identified have helped to inform the checklist of technical requirements in Section 5.3. Part of this process involved a workshop with PRACE partners, large international installations and vendors and this has being repeated in September 2009, where an update on possible PRACE installation sites was discussed. It confirmed the findings in D7.3 [14].

WP7 Task 7.6 *Procurement Process Template* is responsible for defining the PRACE procurement strategy including the vendor selection criteria and acceptance criteria for tender responses. As this document is one of the key inputs into the procurement process the two tasks are closely linked. More information relating technical requirements to the procurement process are discussed in Chapter 3.

Work Package 4 (WP4) is concerned with Distributed System Management and providing a consistent user experience when accessing the proposed Tier-0 PRACE systems and collaborating Tier-1 national systems. Deliverable D4.2.2 [17] is used to provide technical requirements for the connectivity between systems and the software stack required to be supported.

3 Procurement Elements

This chapter provides procurement elements that are needed to support the technical requirements in a tender document.

Task 7.6 “Procurement process template” is responsible for defining a procurement process for the PRACE RI and this document feeds into that process. One of the deliverables from this task, D7.6.3 [16], documents the acceptance tests and evaluation criteria that form the final part of the procurement process. The evaluation criteria that directly relate to the technical requirements are discussed further in this chapter, with the remaining evaluation criteria and acceptance tests left to D7.6.3 [16].

3.1 Presentation of Technical Requirements

The key part of any procurement is an accurate description of the requirements that need to be met by a vendor and the technical requirements form the main procurement element in this document. The approach taken is one of flexibility and adaptability as the PRACE funding model is still under discussion and the procurement process may vary according to the class of architecture being procured. Flexibility is also important as this document is intended to be subject to continuous improvement in the PRACE RI as it should be updated with new information from vendor surveys as well as with feedback from procurement experience.

Technical requirements are split into two groups. The first group, presented in Chapter 4, avoids specifying final machine sizing by using minimum values and ratios, such as memory per compute node. This is designed to leave open the way future procurements are organised so, for example, will allow a procurement to start with a fixed budget and seek to acquire the best performance for the available budget or seek the lowest price for a fixed performance. These requirements are provided on a per architecture basis.

The second group of technical requirements in Chapter 5 is presented as a checklist with example values for clarity. A specific procurement will decide which requirements are relevant and where appropriate tailor the requirement values and target them at a specific installation site.

3.2 Guidelines and Constraints for Vendor Response

A procurement process needs to set out ground rules for vendor responses. These rules are intended to help remove ambiguity in the response to requirements and so improve the quality of answers received. This allows responses to be more meaningfully compared and provides transparency in the selection process.

Requirement values, unless otherwise stated, are minimum values to be met and the vendor can offer better values. Where a value is presented as a ratio the value in the vendor response will depend on the solution offered, typically the number of computing nodes.

Desirable requirements are so categorised to give vendors the option of meeting them or not and to provide the opportunity for vendors to differentiate themselves over the competition.

The procurement process will compare and contrast additional vendor offerings depending on the evaluation scheme selected; see Section 3.4 for more information.

Questions are included to ensure comparable information is provided in vendor responses.

It is important not to over constrain the specification of requirements and one way to introduce flexibility into the process is to allow vendors to propose technology changes to save money (for example through reduced energy consumption) or to improve performance. This flexibility needs to be made explicit in the vendor response constraints.

3.3 Rules for running benchmarks

This is aimed at vendors who will use the PRACE benchmark suite to provide metrics for comparison of solutions and improve confidence that the solution will meet the required system performance. The benchmark suite is also available to be used as part of an acceptance test of an acquired system, but this is not considered further in this document.

Task 6.3 is responsible for producing the benchmark suite incorporating the Peta-scaling and optimisation tasks of WP6 in preparation for PetaFlop/s systems [19].

The following rules are to be applied to running the benchmark suite:

1) Ideally the test machine configuration should be equivalent to the proposed final system but as this may not be possible when procuring leadership class systems the vendor may run benchmarks on a reduced system partition and explain how the results can be projected on to the full proposed system.
2) The test machine software stack should represent a production system. All system services, which are running during the benchmarking, must be listed.
3) The test machine size, in terms of processing units, should be at least n% (*value to be defined during procurement*) of the proposed final system.
4) The test machine network interfaces should be the same as the proposed final system.
5) The disk space size should be at least m% (*value to be defined during procurement*) of the proposed final system.
6) The kernel configuration should remain the same during all benchmark runs.
7) The following benchmark optimisations are allowed. Separate benchmark runs may be made with one of a) or b) mandatory and the remaining benchmark runs optional:
    a) no modifications in the code and the same compiler and compiler options for all benchmarks,
    b) no modifications in the code (except for library changes) and:
        i) dedicated compilers provided by the hardware vendor,
        ii) benchmark specific compiler optimisations, with flags generally available to HPC community,
        iii) code changes to call optimised libraries performing the same algorithms are allowed as long as the libraries used are reported along with version and library provider,
        iv) library calling sequences and parameter types must be unchanged.
    c) modifications to the code, if the results without modification are given and:
        i) code changes are not allowed to alter the algorithm used,
        ii) calculations should be run in the same precision as the unmodified version,
        iii) code changes should be achievable by the average user,
        iv) all changes must be supplied with the results,
        v) the time and effort to make changes must be reported,

vi) knowledge of the output of the benchmark can not be used to skip parts of the code,

vii) optimisations should not require super user privileges.

8) Where multiple runs are made please provide all results so that the variance across runs can be determined.

9) Report the power consumption during the benchmark tests, both total Watts and Flops / Watt if a benchmark gives a Flops result. Power consumption should include processing units, I/O units, management servers and network switches.

The evaluation of benchmarking by vendors, such as scoring the benchmark results, is not included in this document.

3.4 Evaluating Vendor Response to Technical Requirements

A quantitative method for evaluating and comparing vendor responses to technical requirements is to score the responses with a weighted points system. The following method, along with points for each relevant requirement, may be provided as part of the tender document, dependant on the type of procurement.

A desirable requirement is scored as one of:

1) A fixed number of points if the requirement is met.

2) A number of points per improved value over a base value up to a maximum.

3) A fixed number of points for the best performer and a reduced pro-rata value for worse values using the formula:

   (1 – (abs (best value - value) / best value)) x points

   So for a maximum of 10 points and power usage of 18kW (best value) and 20kW (value) the former would earn 10 points, the latter 9 points. A power usage of 36kW would earn 0 points.

   Points cannot be less than 0.

The number of points are summed for each response and normalised so that the response with the highest number of points is assigned a value of 100. Normal rounding to a whole number is used.

For example:

Respondent	Technical Assessment Points scored	Technical Assessment Normalised points
A	34	81
B	42	100
C	28	67

This method allows other technical elements of the procurement, such as scoring benchmarking results, as well as non-technical elements such as capital costs and support costs to be combined into a final score. Refer to Section 3.5 to see a list of all the elements that form the TCO although how these are combined is not discussed in this document. This is typical of the *most economically advantageous tender* (MEAT) as introduced in D7.6.1 [18] and best value procurement approach as opposed to lowest initial cost.

3.5 Total Cost of Ownership

The Total Cost of Ownership (TCO) for the system is an important figure that will need to be derived during a procurement process and matched to the available budget. The TCO methodology along with indicative costs is defined in D7.1.3 [13]. Some of the elements that make up the TCO are related to the technical requirements in this document and the list from D7.1.3 is reproduced here to indicate where the TCO calculation has input from these requirements (the relevant requirement section is added in italics):

- Supercomputer including installation – *hardware requirements including system sizing*,
- Related IT equipments needed for the operation of the supercomputer: storage system (including back-up), internal computer centre networks (including connection point for external network connection), including installation – *supporting systems*,
- Maintenance of the supercomputer and related IT equipments and software licences, including vendor support for hardware and software – *maintenance and support requirements*,
- Building (floor space for the IT equipments, the technical facilities, offices for computer centre team) – *installation constraints,*
- Technical facilities including cooling, power supply (transformers, UPS, distribution, etc.) - *operational requirements*,
- Maintenance of the building and of the technical facilities – *no dependency,*
- Electricity charge including the cost of the power line and main substation if needed and not shared with other facilities – *operational requirements and installation constraints,*
- The staff including management, computer centre operation, application support, building and technical infrastructure support. Application support may actually be considered as including development and job submission tools support, code profiling, optimization, porting and scaling. – *no dependency,*
- Training (users and administrators) – *documentation and training requirements,*
- Some (minor) evolutions and upgrades necessary within the 5 years of operation (most likely within the 2 or 3 first years) – *hardware requirements including system sizing.*

4 Technical Requirements for System Sizing

This chapter contains a list of the key technical requirements for sizing a PetaFlop/s system. These requirements are presented as maximum or minimum global values, values per computational node or system ratios. As the number of nodes increases (or decreases) the total value will change and so is scalable with number of nodes. The intention is to avoid specifying a solution and so this approach provides more flexibility for a specific procurement where the procurement process may vary and where the budget is yet to be defined.

In Section 2.1 the latest analysis of user application requirements for system architectures is detailed and this determines the key architectural features required by systems in the PRACE RI. The quantitative values entered in this chapter have been informed by the assessment of the WP7 PRACE prototypes using the synthetic benchmark suite and provide performance values that may be measured both by vendors and installation sites during acceptance tests. These values represent a snapshot of the systems and architectures that have been available to PRACE during the 2008 and 2009 timeframe and will inevitably change as the vendor offerings develop. Some system sizing values are dependant on the scale of the system and this is indicated within the tables.

As a general point, care needs to be taken when specifying system-sizing values so as not to over constrain vendor responses. They may also depend on the type of procurement, for example with novel solutions or large-scale machines a collaborative design process may require a few initial sizing values that are refined as the design progresses.

In the first section of this chapter the requirements are defined with a numbering scheme and in the second section values are specified for the hardware architectures being considered within the PRACE project. Note that not all requirements may be applicable to a specific procurement. These requirements together with the checklist of requirements listed in Chapter 5 will be the basis for the information to be sent to vendors in the calls for tender.

Each technical requirement includes the following information:

a. A requirement category, which is one of two types,

   1. R (required) is a fixed requirement that a vendor must meet for the PetaFlop/s systems,
   2. D (desirable) is a feature which a vendor does not have to meet, but would be advantageous and may be used to differentiate similar vendor bids,

b. A unique number to allow unambiguous referencing in this and other documents, where the number has the format n.m where n is 1 for sizing requirements and m is an increasing integer,

c. A descriptive title,

d. Optional notes to provide a fuller description of the requirement and to help remove ambiguity.

4.1 Requirement Definitions

Ref	Title	Notes
CPU		
R1.1	Minimum Peak Flop/s [2] in PetaFlop/s.	Calculated as the theoretical maximum double precision floating point operations per second for the system. A sum of the peak Flop/s for all processing units. The calculation processors must support floating point calculations using IEEE-754 representation.
Memory		
R1.2	Minimum memory per processing unit (GByte).	Needed to set a lower limit on the memory available to an application task. To be set based on the prototype configurations D5.1.2 [27] and the application benchmarking experience. Task 7.2.
R1.3	Calculation node memory compared to Flop/s performance (Byte per Flop/s).	Memory from R1.2, Flops is Rmax from D5.2 T1.1 [28] Linpack runs.
R1.4	Memory to processor bandwidth (Byte/s) compared to Flop/s performance (Byte per Flop).	Ratio calculated in D5.2 [28] as Stream sustained bandwidth / Linpack Flop/s.
Message Passing Network		
R1.5	Minimum network bandwidth per processing unit (GByte/s)	Based on the SkaMPI Alltoall benchmark. Derived as P*(P-1)*M/T / number of nodes. Enter a range and the scales for which values are valid. Minimum scale must be 2 nodes or greater.
R1.6	Ratio of minimum network bandwidth (Byte/s) per calculation node to Flop/s (Byte/Flop).	Network performance for applications needs to scale with increased calculation rates, hence this ratio. Informed by T2.2 ratios in D5.2 [28]. Enter a range and the scales for which values are valid. Minimum scale must be 2 nodes or greater.

Ref	Title	Notes
R1.7	Point to point timing (µs).	Use figures from D5.3 [29] where the SKAMPI point to point latency has been measured. Enter a range and the scales for which values are valid. Minimum scale must be 2 nodes or greater.
R1.8	Barrier timing (µs).	Use figures from D5.3 [29] where the SKAMPI barrier time has been measured. Enter a range and the scales for which values are valid. Minimum scale must be 2 nodes or greater.
Storage and I/O – Global		
R1.9	Minimum global disk storage (PByte) per PetaFlop/s performance.	This is defined as fast hard drive disk space for use in running applications, as opposed to slower disks or tape for archive storage. This is global to a single Tier-0 site. This figure only includes storage space available to running applications i.e. it excludes redundant disk storage.
R1.10	Maximum size for a global file system partition (PByte/s).	
R1.11	Peak read bandwidth required by a processing unit (MByte/s).	The bandwidth is informed by the POSIX IOR read separated benchmark in D5.3 [29]. Enter a range and the scales for which values are valid.
R1.12	Peak write bandwidth required by a processing unit (MByte/s).	The bandwidth is informed by the POSIX IOR write separated benchmark in D5.3 [29]. Enter a range and the scales for which values are valid.
R1.13	Percentage of processing units requiring concurrent access at peak bandwidth.	Peak bandwidth which needs to be supported by the I/O subsystem is calculated as number of processing units * peak bandwidth per processing unit * % requiring concurrent access.
Storage and I/O – Local		
R1.14	Minimum user scratch space specified as multiple of calculation node memory size.	Scratch space is working space required by a running job and does not need to persist between jobs. This may be located on local storage or on global storage depending on the system solution.

Ref	Title	Notes
R1.15	Peak read bandwidth required by a calculation node to scratch space (MByte/s).	The bandwidth is informed by the POSIX IOR read separated benchmark in D5.3 [29].
R1.16	Peak write bandwidth required by a calculation node to scratch space (MByte/s).	The bandwidth is informed by the POSIX IOR write separated benchmark in D5.3 [29].
Installation Constraints		
D1.17	Performance compared to power consumption (TFlop/s / MW).	Specified for compute nodes and not ancillary equipment. Excludes cooling power. Informed by D5.2 T6.2 [28]. Flop/s rate is Linpack Rmax.
D1.18	Performance compared to floor space (TFlop/s / m^2).	Excludes cooling components. Informed by D5.2 [28] and D5.1.2. Flop/s rate is Linpack Rmax.

Table 9: Technical requirements for system sizing

4.2 Requirement Values

The following tables provide system-sizing values for all hardware architectures considered by PRACE [13]. Different classes of supercomputing architecture are included here to support procurement of a complementary set of computing facilities that will form the PRACE research infrastructure.

Task 7.1 issued a request for information (RFI) to vendors of potential PetaFlop/s systems asking for proposals for a PetaFlop/s system in the 2009/2010 timeframe. Based on the vendor responses the following hardware architectures are being considered, as one or more systems are likely to be available in this time frame:

1. Homogeneous cluster (few cores, small memory)
2. Homogeneous cluster (more cores, larger memory)
3. Heterogeneous cluster (small memory)
4. Massively-parallel system (small memory)
5. Massively-parallel system (larger memory)

These architectures are defined in Section 7.1. Note that with no vector systems or massively multithreaded systems proposed by vendors it appears that there will be no solutions available; so, for this iteration of the document no requirement values have been entered for these classes of system. More information on the background to these decisions is available in the market survey update in D7.1.3 [12].

The requirement sizing values are presented below in a table per architecture-class with the following information:

a. Reference number, as detailed in the requirement definitions,

b. A sizing value which may be one of:

    1. a fixed value indicating no uncertainty, e.g. 5MByte,
    2. a range providing upper and lower limits, e.g. 5MByte-10MByte,
    3. an upper limit, with no lower bound, e.g. <10ms,
    4. a lower limit, with no upper bound, e.g. >5MByte,
    5. a value of "not available", if information is not available,
    6. a value of "not applicable", if the requirement is not applicable for a particular architecture class,
    7. a blank value indicating no decision has been made about this requirement.

c. A justification commentary to allow references and/or a commentary on how the value was produced.

4.2.1 *Homogeneous Cluster 1*

This class is based on clusters of nodes containing CPUs without on-chip vector units, with no or only moderate multi-threading, without off-chip accelerators, with an industry-standard interconnect, for which packet processing is done on the CPU or on the NIC, and with a full standard OS on the compute nodes. This first set of sizing values is targeted at nodes with few cores and small memory per core.

Ref	Value	Justification
CPU		
R1.1	1 – 3 PetaFlop/s	Peak Flop/s based on analysis in D7.1.3 with the range needed to achieve a Top 5 position in Nov 2010.
Memory		
R1.2	2 - 4 GByte memory per processing unit	Based on prototypes configuration. (both INTI at CEA and JUROPA at FZJ have 3 GByte/core). Recommended to equip 5% to 10% of nodes of a large cluster with more memory (4 to 8 GByte/core).
R1.3	0.22 – 0.45 Byte memory / Flop/s	Based on 2 – 4 GByte memory per processing unit and 1024 processing units Linpack result of 9.1 TFlop/s on INTI.
R1.4	0.30 Byte/s memory bandwidth / Flop/s	Based on assessment in D5.2 [28] for Nehalem prototypes with Stream bandwidth and Linpack result .
Message Passing Network		
R1.5	288 – 18 MByte/s per processing unit	Measured for 16 – 512 processing units.
R1.6	0.0314 – 0.0019 Byte/s / Flop /s	Measured for 16 – 512 processing units. Based on INTI prototype in D5.2 [28].
R1.7	3.1 – 4.0 µs	Measured for 16 – 512 processing units.
R1.8	6.8 – 79.3 µs	Measured for 16 – 512 processing units.
Storage and I/O – Global		
R1.9	20 PByte	Based on sizing values in D7.1.3. Global disk space for each PetaFlop/s of peak performance.

Ref	Value	Justification
R1.10	5 PByte	Based on 4 file systems on Global storage
R1.11	264 – 14 MByte/s per processing unit	Measured for 16 – 64 processing units. See Note 1.
R1.12	40 – 10 MByte/s per processing unit	Measured for 16 – 64 processing units.
R1.13	10%	This is an average estimate but will vary according to each application code.
Storage and I/O – Local		
R1.14	10-15	This is an average estimate but will vary according to each application code.
R1.15		No additional requirements are offered as the “local” storage is expected to be a partition on the global file system.
R1.16		No additional requirements are offered as the “local” storage is expected to be a partition on the global file system.
Installation constraints		
D1.17	318 TFlop/s per MW	Based on measurement of Linpack at 9.1 TFlop/s on 1024 core INTI cluster (29 KW).
D1.18	Approx 3 TFlop/s per m^2	Based on 1.45m^2 for INTI prototype of 64 nodes in 2 racks, but not including storage and network components. This area is doubled to allow for other components. Note however that extrapolating from this small scale prototype may not be accurate.

Table 10: System sizing values for Homogeneous clusters 1

4.2.2 *Homogeneous Cluster 2*

This class is a homogeneous cluster but with more cores and larger memory than the first set of Homogeneous clusters. Typically CPUs on the compute nodes can be shared between jobs, the nodes have many I/O slots and other infrastructure and there is lots of shared memory with small bandwidth per core. This class of system architectures is also known as a "fat node".

Ref	Value	Justification
CPU		
R1.1	1 – 3 PetaFlop/s	Peak Flop/s based on analysis in D7.1.3 with the range needed to achieve a Top 5 position in Nov 2010.
Memory		
R1.2	4 - 8 GByte memory per processing unit	Based on prototype configuration.
R1.3	0.39 – 0.78 Byte memory / Flop/s	Based on 4 – 8 GByte memory per processing unit and 2048 processing units Linpack result of 21 TFlop/s from Huygens prototype in D5.2 [28].
R1.4	0.008 Byte/s memory bandwidth / Flop/s	Based on assessment in D5.2 [28] for Huygens prototype with Stream bandwidth and Linpack result.
Message Passing Network		
R1.5	144 – 77 MByte/s per processing unit	Measured for 64 – 2048 processing units.
R1.6	0.0117 – 0.0076 Byte/s / Flop /s	Measured for 64 – 2048 processing units. Based on Huygens prototype in D5.2 [28].
R1.7	9.6 – 10.6 µs	Measured for 64 – 2048 processing units.
R1.8	15 – 38.9 µs	Measured for 64 – 2048 processing units.
Storage and I/O – Global		
R1.9	20 PByte	Based on sizing values in D7.1.3. Global disk space for each PetaFlop/s of peak performance.
R1.10	5 PByte	Based on 4 file systems on Global storage

Ref	Value	Justification
R1.11	55 – 6 MByte/s per processing unit	Measured for 64 – 2048 processing units. See Note 1.
R1.12	45 – 1 MByte/s per processing unit	Measured for 64 – 2048 processing units.
R1.13	30%	This is an average estimate but will vary according to each application code.
Storage and I/O – Local		
R1.14	10-15	This is an average estimate but will vary according to each application code.
R1.15		No additional requirements are offered as the "local" storage is expected to be a partition on the global file system.
R1.16		No additional requirements are offered as the "local" storage is expected to be a partition on the global file system.
Installation constraints		
D1.17	80 TFlop/s per MW	Based on measurements in D5.2 [28] for Huygens prototype with 3328 processing units.
D1.18	0.39 TFlop/s per m^2	Based on $125m^2$ including storage and network components for Huygens prototype.

Table 11: System sizing values for Homogeneous clusters 2

4.2.3 *Heterogeneous Cluster*

This architecture class is similar to a homogeneous cluster but with a heterogeneous multi-core CPU.

Ref	Value	Justification
CPU		
R1.1	1 – 3 PetaFlop/s	Peak Flop/s based on analysis in D7.1.3 with the range needed to achieve a Top 5 position in Nov 2010.
Memory		
R1.2	4 – 16 GByte memory per processing unit	Based on prototype configuration.
R1.3	0.058 – 0.23 Byte memory / Flop/s	Based on 4 – 16 GByte memory per processing unit and 144 processing units Linpack result of 10 TFlop/s from MariCel prototype in D5.2 [28].
R1.4	0.098 Byte/s memory bandwidth / Flop/s	Based on assessment in D5.2 [28] for MariCel prototype with modified Stream bandwidth and Linpack result.
Message Passing Network		
R1.5	73 – 47 MByte/s per processing unit	Measured for 4 –128 processing units.
R1.6	0.00067 – 0.00106 Byte/s / Flop/s	Measured for 4 – 128 processing units. Based on MariCel prototype in D5.2 [28].
R1.7	19.0 – 24.9 µs	Measured for 4 – 128 processing units.
R1.8	20 – 181 µs	Measured for 4 – 128 processing units.
Storage and I/O – Global		
R1.9	20 PByte	Based on sizing values in D7.1.3. Global disk space for each PetaFlop/s of peak performance.
R1.10	5 PByte	Based on 4 file systems on Global storage
R1.11	16.8 – 2 MByte/s per processing unit	Measured for 4 – 64 processing units. See Note 1.
R1.12	13 – 1 MByte/s per processing unit	Measured for 4 – 64 processing units.

Ref	Value	Justification
R1.13	30%	This is an average estimate but will vary according to each application code.
Storage and I/O – Local		
R1.14	10-15	This is an average estimate but will vary according to each application code.
R1.15	300 MByte/s	Estimated. If satisfied by direct attached disks then there will be no scaling adjustments.
R1.16	300 MByte/s	Estimated. If satisfied by direct attached disks then there will be no scaling adjustments.
Installation constraints		
D1.17	455 TFlop/s per MW	Based on measurements in D5.2 [28] for MariCel prototype with 128 processing units.
D1.18	5 TFlop/s per m^2	Based on $2m^2$ for the 2 racks of the MariCel prototype. Note that this is a small-scale prototype and that extrapolating to a full scale system may not be accurate.

Table 12: System sizing values for Heterogeneous clusters

4.2.4 *Massively-Parallel System 1*

This architecture class is characterised as a custom-built supercomputer using a custom interconnect with very high bandwidth and low latency that is able to handle the packet processing. These systems use a customised operating system on the compute nodes. These classes use a super-scalar CPU with moderate multi-threading. This first group of requirements is aimed at machines with a small number of cores and memory.

Ref	Value	Justification
CPU		
R1.1	1 – 3 PetaFlop/s	Peak Flop/s based on analysis in D7.1.3 with the range needed to achieve a Top 5 position in Nov 2010.
Memory		
R1.2	0.5 – 1 GByte memory per processing unit	Based on prototype configuration.
R1.3	0.195 – 0.39 Byte memory / Flop/s	Based on 0.5 – 1 GByte memory per processing unit and 65536 processing units Linpack result of 168.5 TFlop/s from Jugene prototype in D5.2 [28].
R1.4	0.43 Byte/s memory bandwidth / Flop/s	Based on assessment in D5.2 [28] for Jugene prototype with Stream bandwidth and Linpack result.
Message Passing Network		
R1.5	119 – 35 MByte/s per processing unit	Measured for 8 – 4096 processing units.
R1.6	0.0457 – 0.0134 Byte/s / Flop/s	Measured for 8 – 4096 processing units. Based on Jugene prototype in D5.2 [28].
R1.7	6.2 – 438 µs	Measured for 8 – 4096 processing units.
R1.8	3.6 – 4.1 µs	Measured for 8 – 4096 processing units.
Storage and I/O – Global		
R1.9	20 PByte	Based on sizing values in D7.1.3. Global disk space for each PetaFlop/s of peak performance.
R1.10	5 PByte	Based on 4 file systems on Global storage

Ref	Value	Justification
R1.11	50 – 0.3 MByte/s per processing unit	Measured for 8 – 16384 processing units. See Note 1.
R1.12	37.5 – 0.035 MByte/s per processing unit	Measured for 8 – 16384 processing units.
R1.13	30%	This is an average estimate but will vary according to each application code.
Storage and I/O – Local		
R1.14	10-15	This is an average estimate but will vary according to each application code.
R1.15		No additional requirements are offered as the "local" storage is expected to be a partition on the global file system.
R1.16		No additional requirements are offered as the "local" storage is expected to be a partition on the global file system.
Installation constraints		
D1.17	300 TFlop/s per MW	Based on measurements in D5.2 [28] for Jugene prototype with 65536 processing units.
D1.18	2 TFlop/s per m^2	Based on 85m^2 for the Jugene prototype.

Table 13: System sizing values for Massively-Parallel system 1

4.2.5 *Massively-Parallel System 2*

This second group of requirements for MPP systems is aimed at machines with a larger number of cores and more memory.

Ref	Value	Justification
CPU		
R1.1	1 – 3 PetaFlop/s	Peak Flop/s based on analysis in D7.1.3 with the range needed to achieve a Top 5 position in Nov 2010.
Memory		
R1.2	1 – 2 GByte memory per processing unit	Based on prototype configuration.
R1.3	0.14 – 0.28 Byte memory / Flop/s	Based on 1 – 2 GByte memory per processing unit and 9360 processing units Linpack result of 66.25 TFlop/s from Louhi/Loviatar prototype in D5.2 [28].
R1.4	0.05 Byte/s memory bandwidth / Flop/s	Based on assessment in D5.2 [28] for Louhi/Loviatar prototype with Stream bandwidth and Linpack result.
Message Passing Network		
R1.5	40 – 11 MByte/s per processing unit	Measured for 16 – 2048 processing units.
R1.6	0.0057 – 0.0015 Byte/s / Flop/s	Measured for 16 – 2048 processing units. Based on Louhi/Loviatar prototype in D5.2 [28].
R1.7	17.6 – 21.1 µs	Measured for 16 – 2048 processing units.
R1.8	14.1 – 508 µs	Measured for 16 – 2048 processing units.
Storage and I/O – Global		
R1.9	20 PByte	Based on sizing values in D7.1.3. Global disk space for each PetaFlop/s of peak performance.
R1.10	5 PByte	Based on 4 file systems on Global storage
R1.11	17 – 1.5 MByte/s per processing unit	Measured for 128 – 1440 processing units. See Note 1.

Ref	Value	Justification
R1.12	13 – 1.2 MByte/s per processing unit	Measured for 128 – 1440 processing units.
R1.13	30%	This is an average estimate but will vary according to each application code.
Storage and I/O – Local		
R1.14	10-15	This is an average estimate but will vary according to each application code.
R1.15		No additional requirements are offered as the “local” storage is expected to be a partition on the global file system.
R1.16		No additional requirements are offered as the “local” storage is expected to be a partition on the global file system.
Installation constraints		
D1.17	165 TFlop/s per MW	Based on measurements in D5.2 [28] for Louhi/Loviatar prototype with 9360 processing units.
D1.18	3.8 TFlop/s per m^2	Based on $17.5m^2$ floor area for Louhi/Loviatar prototype, including compute nodes and disk storage.

Table 14: System sizing values for Massively-Parallel system 2

Note 1: The original sizing estimate for I/O bandwidth in Task 7.1 was 200 GByte/s per 20 PByte of global storage. For 100,000 processing units this equals 2 MByte/s per processing unit. If 30% of processing units require concurrent access at peak bandwidth a value of approx 7 MByte/s is achievable for these processing units.

5 Check List of Technical Requirements

This chapter provides a checklist of typical technical requirements used in HPC procurements, excluding those that have already been listed in Chapter 4. It forms one of the tools in preparing procurements and is presented as a descriptive list with values added to clarify meaning. Site-specific procurements are free to select those requirements that are relevant and add values meaningful to the type of system being procured.

Each technical requirement includes the following information:

a. A requirement category, which is one of three types,

   1. R (required) is a fixed requirement that a vendor must meet for the PetaFlop/s systems,
   2. D (desirable) is a feature which a vendor does not have to meet, but would be advantageous and may be used to differentiate similar vendor bids,
   3. Q (question) is a question to the vendor, where information is needed to evaluate offers,

b. A unique number to allow unambiguous referencing in this and other documents, classified into similar requirement groups, n.m where m provides a unique number within each group and n is,

   2. hardware (complementary to system sizing in Chapter 4),
   3. software,
   4. operational,
   5. maintenance and support,
   6. supporting systems,
   7. documentation and training,
   8. delivery requirements,

c. A descriptive title,

d. Optional notes to provide a fuller description of the requirement and to help remove ambiguity.

5.1 Hardware Requirements

Ref	Title	Notes
CPU		
R2.1	A capability system is required [2].	This is defined as a system with the ability to run a single MPI application on all calculation nodes requiring fast inter process communication.
R2.2	Calculation core bit size.	For example 32 or 64. The calculation processors must support floating-point calculations using IEEE-754 representation.
Q2.3	Vendor to provide information on an upgrade path for processing units after 2 to 3 years of production use.	Vendor to specify what upgrades are available or planned for release in this timeframe.
Q2.4	Vendor to provide information on options for cache levels; location, size, associativity.	This is defined as a question for vendors as usually there are very limited user choices for cache sizing.
Q2.5	Vendor to specify the number of Simultaneous Multi Threading (SMT) threads supported and the restrictions on how instructions from different threads are scheduled together.	SMT provides CPU efficiency improvements and can help to hide memory latency.
Memory		
R2.6	Mechanism for error detection and correction in main memory.	This is defined as the use of error correcting codes in memory controllers to automatically reconstruct memory contents using parity bits. As the total amount of memory used by a capability job reaches higher levels failures may become significant.
Q2.7	Vendor to provide peak memory bandwidth.	For NUMA architectures specify these values for varying memory hops, local, 1st up to nth in GByte/s.
Q2.8	Vendor to provide minimum memory latency.	For NUMA architectures specify these values for varying memory hops, local, 1st up to nth in nanoseconds.
Q2.9	Vendor to provide available memory configurations, including free slots for upgrade and memory unit sizes.	This includes information for memory upgrades during the system lifetime.

Ref	Title	Notes
Network These can be repeated for each internal network required. For example message passing, I/O, operating system mounting, management.		
R2.10	All calculation nodes are required to be connected to the network.	
Q2.11	Vendor to specify network technology.	For example InfiniBand, MyriNet, Ethernet.
Q2.12	Vendor to specify network topology.	This defines the physical and virtual connectivity of the computation and other nodes, such as storage servers.
Q2.13	Vendor to provide information on an upgrade path for network components.	
I/O and Storage		
R2.14	Minimum number of file system partitions to be supported by the global storage system.	
Q2.15	Vendor to specify the types of file system supported on the global storage system.	
Q2.16	Maximum global file system size supported.	PByte.
Q2.17	Maximum individual file size supported by the global file system.	TByte.
Q2.18	Maximum number of concurrent clients (computing nodes) which may connect to the global file system.	
D2.19	A long term upgrade path for global storage is required. Factor by which I/O bandwidth and storage may need to be increased in the future.	For example, after adding more processing nodes an increase in I/O and/or storage may be required.
D2.20	Maximum latency of local storage reads and writes.	Microseconds.

Ref	Title	Notes
D2.21	Minimum swap space specified as multiple of calculation node memory size.	This is optional (swap space may not be desired for an HPC application) and may be located on local storage or on global storage depending on the system solution.
Optional Extensions (vector and accelerators)		
D2.22	Ability to install full-sized, non-proprietary, 3rd party full length PCIe extensions,	Solutions based on additional nodes or chassis extensions are acceptable.

Table 15: Hardware requirements checklist

5.2 Software Requirements including system software and programming environment

The requirements listed in Table 16 concentrate on the functionality required from the software for the system. Performance requirements such as timings are listed under the Operational Requirements section. Distributed systems management software requirements are related to information that is published in D4.2.2 [17].

Ref	Title	Notes
Operating System		
R3.1	The operating system should be UNIX like.	It should be compatible with the X/Open Standard POSIX 1003 (ISO/IEC 9945).
Q3.2	Vendor to provide details of supported operating systems.	
R3.3	Nodes are able to be booted from multiple system images.	
R3.4	The maximum node CPU usage (%) by the operating system, with no applications running.	
R3.5	The maximum node memory usage (%) by the operating system, with no applications running.	
D3.6	Mechanisms to prevent uncoordinated interruption of user processes by O/S tasks to reduce operating system jitter.	
R3.7	Support for large page sizes.	Page sizes much larger than 4 KByte.
R3.8	Ability to dynamically (on a per process or job basis) alter the number of large pages available on a node, depending on user demands.	No reboot of the system is required to accomplish this task.
R3.9	The file systems to be supported on local disk storage.	
R3.10	Minimum number of open files for each process.	Used to ensure the maximum file descriptor table size can cater with Petaflop/s scale applications

Ref	Title	Notes
Programming Environment – Languages, Programming Models, Compilers		
R3.11	The programming environment should support C, C++ and Fortran.	
D3.12	The programming environment should support Java.	
R3.13	Interoperability between Fortran, C, and C++.	
R3.14	C compiler for compute nodes must at least support a full implementation of the standard ANSI/ISO 9899-1990 („C90“)	
D3.15	C compiler for compute nodes, supporting ANSI C99 standard.	
R3.16	C++ Compiler for compute nodes must at least support a full implementation of the standard ANSI/ISO 14882-1998 („C++98“), including the C++ standard library	
R3.17	Fortran compiler for compute nodes supports a full implementation of the language specifications of Fortran 95 (ANSI X3J3/96-007)	
D3.18	Fortran compiler for compute nodes supports a full implementation of the language specification of the Fortran 2003 standard.	
R3.19	A recent version of the GCC is available which supports the system hardware.	Note that some accelerators are delivered with dedicated compilers and are not supported by the GCC directly so this may be limited to CPUs and not accelerators. The same is true for vector systems.
Q3.20	Vendor to provide details of vendor optimised compilers and libraries including operating system support.	
R3.21	Compilers support 32 and 64-bit mode.	
D3.22	Support for PGAS programming model with support for emerging compilers, tools, and libraries.	For example Co-array Fortran, UPC (Unified Parallel C) and vendor-specific constructs for global data addressing such as SHMEM.

Ref	Title	Notes
R3.23	Ability to run different versions of compilers, linkers, libraries, applications, etc. alternatively or additionally to the standard programming environment	
Q3.24	Vendor to provide details of supported programming interfaces for any accelerator devices.	
D3.	Accelerated processors are required to use a standard programming model.	Where defined as add on units, either boards on a node or separate nodes.
Programming Environment – MPI and OpenMP		
R3.25	MPI library for compute nodes, fully supporting MPI Version 1.2	
D3.26	MPI library for compute nodes, supporting MPI Version 2.1.	With the exception of dynamic process spawning routines.
Q3.27	Vendor to provide details of MPI implementations (version 1 and version 2) supported and level of support (for example which parts of MPI 2 specification supported).	Include details on any MPI optimisations.
D3.28	Where compute nodes support threading, the MPI library must implement the highest level of thread safety (MPI_THREAD_MULTIPLE).	
R3.29	Shared memory applications are able to use POSIX threads.	The implementation should be compatible with the X/Open Standard POSIX 1003 (ISO/IEC 9945).
R3.30	If compute nodes have hardware-shared memory the Fortran, C and C++ compilers must fully support the OpenMP Version 2.5 standard.	
Programming Environment – Tools		
R3.31	The following debugging tools should be usable on the system.	To be populated at procurement time.
R3.32	The following profiling and optimisation tools should be useable on the system.	To be populated at procurement time.

Ref	Title	Notes
R3.33	A parallel debugger is available for the compute nodes.	Vendor to specify.
R3.34	A sequential performance analysis tool is available for the compute nodes.	Vendor to specify.
R3.35	A parallel performance analysis tool is available for the compute nodes with profiling capability.	Vendor to specify.
D3.36	Parallel performance analysis tools are available for the compute nodes with MPI tracing and hardware counter capability.	Vendor to specify.
Programming Environment – Libraries		
R3.37	BLAS and PBLAS libraries optimised for the compute nodes	
R3.38	FFTW version 2&3 libraries optimised for the compute nodes	
R3.39	LAPACK library for the compute nodes	
R3.40	ScaLAPACK library for the compute nodes	
D3.41	LAPACK Version 3.1 library for compute nodes	
Programming Environment – Front End/Login Nodes		
R3.42	A version of the Java SDK ≥ 5.0 is available for the front end nodes.	To run developer tools, software editors, also management tools.
R3.43	Front-end nodes have access to the global file system.	
D3.44	C, C++, Fortran cross compilers are available.	Cross compilers are applicable for use by developers outside the system or in the case of heterogeneous systems to build code for different target hardware from a single front-end node.
D3.45	Perl for front-end nodes.	
D3.46	Python for front end nodes.	
D3.47	Emacs editor for login/front-end nodes.	

Ref	Title	Notes
D3.48	Revision control system for login/front end nodes (e.g. CVS, Subversion).	
Scheduling, Batch and Resource Management Software		
R3.49	Ability to efficiently manage different workloads of the system.	For example, dynamically grant resources to system tasks depending on a classification scheme decided by the system administrator.
R3.50	The resource management software allows global control of nodes, processing units, interconnection networks.	
R3.51	The maximum time to start a job on all calculation nodes.	Seconds.
R3.52	The batch and resource management software is compatible with all supported parallel programming models.	
D3.53	The scheduling software supports backfill scheduling.	To avoid underutilisation of reserved nodes.
D3.54	The scheduling software provides a means for co-scheduling and/or resource reservation.	
Q3.55	Vendor to list supported scheduling, batch and resource management software.	
D3.56	It is possible to drain any compute node after the end of running jobs to block its re-use.	
Administration Software		
D3.57	The monitoring software should be able to suspend unused nodes and power them up only when required.	As long as the operation of the rest of the system is unaffected.
R3.58	Tools to change system parameters without system interruption.	
R3.59	Software monitors to measure important system characteristics such as; I/O behaviour, disk access behaviour, CPU load, memory load, paging rate.	An easy-to-interpret output is required.

Ref	Title	Notes
R3.60	Facilities for the on-line detection of hardware errors.	For example faulty memory modules, processors, fans, network links, switches.
D3.61	Tools to extract CPU performance information like number of floating point operations, number of integer operations, main memory and cache references, etc. per second from hardware performance counters	Information should be available to the system administrators on a per CPU-core basis without any impact on user codes and without the necessity of any specific changes to those codes.
Distributed Systems Management Software		
R3.62	The system will support the DEISA user administration system [20].	This uses X.509 certificates and LDAP for user authentication and authorisation.
R3.63	The system can run Java on a node that can be accessed externally to the system.	An accounting transformation and information service is required to run on the system for central reporting of account usage.
R3.64	The system can run the UNICORE [21] version 6 services.	Required by D4.2.2 [17] for Grid access and services such as job monitoring.
D3.65	The system supports the GSI-SSH software [4] or X.509 for SSH.	For interactive command line access.
R3.66	The system can run the modules software (TCL implementation) [3].	This is used to create a standard PRACE environment for grid users.
R3.67	The system can run the GridFTP data transfer tool [6].	Allows users to exchange information between PRACE and external sites.
R3.68	The system can run the lperf network-monitoring tool [5].	This is used to measure external network performance.
R3.69	The system supports the monitoring tool Inca [7].	To monitor the software versions available to users and for system and service status monitoring.

Table 16: Software requirements checklist

5.3 Operational Requirements including installation constraints

The requirements listed in Table 17 relate to the operational use of the system, including reliability and operational management, as well as installation constraints which are linked to information in D7.3 [14].

Ref	Title	Notes
Users and Jobs		
R4.1	Number of concurrent users logged in to the system.	
R4.2	Number of concurrent batch jobs to be supported by the system.	
Reliability and Availability		
R4.3	Minimum mean time between interrupts (MTBI).	This is defined as the mean time between job interrupts, where a job does not complete because of component failure. In hours.
R4.4	Minimum mean time between failures for system components.	Either one value for mean time across all components or separate times for each component type (calculation node, network and disk components) can be specified. In hours.
R4.5	Seamless degradation of the system in case of a failure of a compute or I/O node.	
Q4.6	Vendor to demonstrate how hardware redundancy provides continuity of service to users for different types of component failure.	
R4.7	Calculation nodes provide temperature monitoring and automated shut down if configurable limits are exceeded.	
R4.8	Calculation nodes can be stopped and started without interrupting applications running on other calculation nodes.	
R4.9	Type of hardware redundancy required for global disk storage.	This includes the disk storage hardware topology and disk controllers RAID level.

Ref	Title	Notes
R4.10	File systems support journaling of meta- and/or user data or some equivalent mechanism	
R4.11	Ratio of backup storage space to global storage system, including all levels of any hierarchical storage.	Relates to number of full backups that need to be retained.
R4.12	Backup of data can be achieved in parallel with and without interrupting running jobs.	
R4.13	Maximum time to run full global file system backup.	In hours.
R4.14	Recovery of data from backup can be achieved without interrupting running jobs.	
R4.15	Maximum time to recover full global file system.	In hours.
R4.16	Means to ensure end-to-end data consistency for global file system.	
R4.17	Facilities for the on-line detection and correction of media errors in global file system.	
R4.18	Maximum time to check file system after media errors in global file system.	In hours.
R4.19	Hardware redundancy of local disk storage.	
D4.20	Automated check pointing at application level and restart capabilities.	
Manageability		
R4.21	All calculation nodes can be started and stopped from a single administration computer	
R4.22	Software can be installed on all calculation nodes from a single administration computer	

Ref	Title	Notes
R4.23	Time required for an operating system upgrade or complete installation of a new operating system on all nodes.	In hours.
R4.24	Maximum time to add a file to the software stack across all nodes.	In minutes.
R4.25	Ability to verify correct installation of software changes.	
R4.26	Maximum time for controlled shut down of system with file system contents preserved.	In minutes.
R4.27	Maximum start-up time per node after a controlled shut down.	In minutes.
R4.28	Maximum start-up time for whole system after a controlled shut down.	In minutes.
R4.29	Maximum start-up time per node after a forced shut down.	In minutes.
R4.30	Maximum start-up time for whole system after a forced shut down.	In minutes.
R4.31	Monitoring the interconnect of the system is possible.	For example, number and size of packets, amount of data sent or received.
R4.32	Error tracking and reporting mechanism in cases of operating system errors. (e.g., system dumps) available and officially supported by the vendor	
D4.33	Modifications of all parts of the operating system (except the kernel) possible at any time, without interrupting the operation, immediately taking effect, and possibility to undo single modifications separately	
R4.34	Possibility to carry out all tasks of user administration via scriptable interfaces including setting of new passwords without interactive editing of files by the system administrator	

Ref	Title	Notes
R4.35	The ability to monitor power consumption of the system components.	
System security		
R4.36	Access control to files on the global file system is managed with Unix groups.	
R4.37	Minimum number of Unix groups required to be supported.	
D4.38	Privileges isolation between nodes is required.	Gaining administrative privileges on one node does not automatically mean these privileges are available on other nodes.
Power		
R4.39	The maximum power consumed by a calculation node whilst idling.	In kW.
R4.40	The maximum power consumed by a calculation node whilst under full load.	In kW.
Installation Constraints		
Q4.41	Vendor to specify the system cooling system type.	
Q4.42	Vendor to specify the system cooling capacity required.	
Q4.43	Vendor to specify the cooling system parameters. Direction of airflow, airflow rate, inlet and outlet temperatures.	
R4.44	Maximum floor space available to system.	In m^2.
R4.45	Access constraints for physical access during installation.	
R4.46	Maximum floor loading available to system.	In kg/m^2.
Q4.47	Vendor to supply any restrictions on cabling distances between components.	

Ref	Title	Notes
Q4.48	Vendor to specify electricity supply requirements (current, voltage, phase).	
R4.49	Maximum heat dissipation per rack.	In kW.

Table 17: Operational requirements checklist

5.4 Maintenance and Support Requirements

Any large HPC cluster is expected to require frequent maintenance and this is particularly important for Petaflop scale machines. The requirements listed in Table 18 address the needs of a high availability Petaflop/s machine with respect to vendor service level agreements (SLA).

Ref	Title	Notes
R5.1	Warranty duration.	In years.
R5.2	Percentage availability of calculation nodes during working hours.	Working hours are 8am-6pm, Monday - Friday excluding public holidays.
R5.3	Percentage availability of calculation nodes during non-working hours.	
R5.4	Response time to reported problem for redundant hardware (compute nodes).	In hours.
R5.5	Response time to reported problem for non-redundant hardware (network, administration nodes).	In hours.
R5.6	Support cover hours for redundant hardware (compute nodes).	
R5.7	Support cover hours for non-redundant hardware (network, administration nodes).	
R5.8	Target repair time for redundant hardware (compute nodes).	In hours.
R5.9	Target repair time for non-redundant hardware (network, administration nodes).	In hours.
R5.10	Maximum time between response to the problem report and full availability of the entire system.	This is added because repair time of the component may not include re-cabling, software reconfiguration etc. that is required for the system to fully enter production mode. In hours.

Ref	Title	Notes
D5.11	Owner can install third party extensions cards and attach third party network devices without warranty void as long as they are compatible and properly installed. Owner can physically relocate the system without warranty lost as long as it is done according to vendor's procedures included within the documentation.	
D5.12	Vendor will provide the owner (within the warranty) with access to the problem reporting system where each problem report is identified by a distinct problem id.	
R5.13	Vendor will provide the owner (within the warranty) with all software upgrades (including operating system and firmware) available for the delivered software and hardware.	
D5.14	Vendor to provide proactive support by notifying recommended firmware and software updates as they become available.	

Table 18: Maintenance and support requirements checklist

5.5 Supporting Systems

This section includes requirements in Table 19 to be considered for the supporting systems that form the infrastructure of any large HPC system. The volumes of data required for petascale systems, both in pre-processing the input data sets and post-processing the output data sets require sufficient resources that must be considered when procuring such a system. Typically, for smaller scale systems, user data has been prepared and analysed by downloaded to a local machine but this approach is no longer appropriate. Either the HPC system must provide sufficient local storage along with compute and visualisation resources or high-speed networks must allow the data to be moved to other centres where these facilities are available.

Pre and post-processing can take advantage of a partition of the petascale system, for example to run pre-processing parallel meshing steps, or use a smaller dedicated distributed memory or shared memory cluster which shares the storage space with the Tier-0 system. Partitions of the main system can be temporarily made available through the batch system or dedicated as interactive nodes.

Visualisation of results from a remote (to the user) system requires streaming of graphical output.

Note that global storage is included in the system sizing and hardware requirements and so is not included here.

Ref	Title	Notes
External Access Network		
R6.1	There should be fast external access to the global file system for user file transfer.	
R6.2	There should be fast external access to the global file system for data backup and recovery.	
R6.3	Minimum external network bandwidth.	Gigabit/s.
D6.4	The infrastructure is required to have access to the DEISA network.	This provides consistent and secure access by researchers to the PRACE infrastructure facilities and is specified in D4.2.2 [17].
Archive and Backup		
D6.5	Hierarchical/tiered storage management is required.	This provides storage space for running applications using fast disk, with archive data moved to slower disk or tape as defined by site-specific policies.

Ref	Title	Notes
D6.6	Percentage of storage space (related to online fast disk) for each level of a hierarchical/tiered storage system.	
Pre-processing, post-processing and Visualisation		
D6.7	Partitions of the Tier-0 cluster can be allocated as interactive nodes.	This may be complimentary to dedicated pre/post-processing resources.
D6.8	A dedicated shared memory machine is required.	Alternative solutions with distributed memory clusters and accelerators such as GPUs should be considered.
D6.9	Minimum memory in GByte.	For a dedicated pre/post-processing system.
D6.10	Minimum TFlop/s peak performance.	For a dedicated pre/post-processing system.
D6.11	Minimum local disk storage in GByte.	For a dedicated pre/post-processing system.
D6.12	The machine must be connected to the Global file system.	For a dedicated pre/post-processing system.
D6.13	The system can run GridFTP [6] for file transfers.	Allow transfers of pre and post processing data to external systems and is specified in D4.2.2 [17].
D6.14	An interactive resources reservation system is available.	Shared systems for pre/post-processing and visualisation require a simple way to make reservations, for example through a web portal.
D6.15	A remote display system is required (software), compliant with current visualisation packages, and with some collaborative features.	Collaborative features should include as a minimum the ability to share mouse control between different remote users.
D6.16	A room with enhanced display (large size, high definition) at the Tier0 site, connected to the post-processing sub-system.	To be used for scientific events or local communication actions.
D6.17	Video conferencing facilities to enable meetings between remote parties and the sharing of screens of visualisation data.	For example Access Grid [23] type features.

Table 19: Supporting Systems checklist

5.6 Documentation and Training Requirements

Adequate documentation and training are necessary for successful installation and configuration of a large HPC system and these requirements are listed in Table 20.

Ref	Title	Notes
Documentation		
R7.1	System is provided with an electronic and optional paper version of a complete list of components.	List includes physical location, model name, serial number and network settings (if applicable).
R7.2	Documentation includes system general description, graphical diagrams of all interconnects (including communication and management network) and configuration values (including network settings).	
R7.3	Documentation describes procedures required for complete disaster recovery.	Physically relocating the system, reconfiguring all components from scratch and reinstalling all provided software.
R7.4	Electronic and optional paper documentation for all supplied software.	
Training		
R7.5	Advanced training at the Owner's location will be provided by the vendor and will cover: system and technology introduction, management tools and procedures, system monitoring and optimization, security, applications and programming. The actual system will be used in the training during the labs.	

Table 20: Documentation and training requirements checklist

5.7 Delivery Requirements

The requirements in Table 21 relate to the procedures for delivery of the system.

Ref	Title	Notes
R8.1	Vendor will provide the owner with complete delivery schedule including starting and ending times of every delivery phases.	
R8.2	Vendor will adjust to the owner’s facility regulations.	Includes safety laws, waste disposal procedures, floor space necessary for packages and system assembly.

Table 21: Delivery requirements checklist

6 Conclusions

This document is the second deliverable from WP7 task 5 *Drafting of Technical Requirements* and provides an update of D7.5.1 [26] based on the work undertaken in the technical work packages of PRACE during 2009. It is designed as a flexible toolbox of technical elements that can be utilised when preparing procurements where the elements used will be dependant on the procurement process and target system. The document supports the PRACE Management Board in selecting the second production PetaFlop/s Tier-0 systems under different funding models, for example they may be procured and hosted by national centres or by a PRACE permanent research infrastructure. The technical requirements listed here complement the work done under WP7 task 6 to develop a procurement process template.

The approach to presenting requirements has continued the method introduced in D7.5.1 [26]. The model for PRACE is to have an infrastructure of 3-5 Tier-0 Petaflop/s systems with a variety of system architectures to support the demands of the key application codes and research areas identified in WP6. This approach of application led procurement, adopted in PRACE, ensures that the investment in applications of the computational simulation research community is protected as they move to the Petascale regime. These technical requirements are focussed on the HPC system architectures identified as most suitable by the WP6 application to architecture mapping exercise that has been updated for this document along with a preliminary indication of which part of the system balance is more important. The identified architectures are homogeneous cluster, heterogeneous cluster and MPP system.

A second key input into the document is the assessment of the WP7-WP5 PRACE prototypes using synthetic benchmarks in WP5. This allows the performance of separate parts of the systems to be measured and feeds directly into the technical requirements. The experience gained in using a standard set of benchmarks is advantageous as it can be supplied to vendors to allow realistic performance figures to be prepared and can be used in acceptance tests for delivered systems.

It is worth noting here that an alternative approach to procuring leadership class systems needs to be considered, particularly for novel systems. The jump into the Exascale regime and beyond will require new architectures, most likely based on many core and heterogeneous CPUs. New approaches to application code design and development will need to take advantage of these new architectural features, so even though the starting point will continue to be the scientific need for higher performance systems, the approach to procuring systems will be led by the HPC system architectures offered.

It is recommended that the work undertaken here be continued in the follow up project to support the implementation phase of the PRACE RI. The technical requirements in this document are a snapshot in time and will need to be updated as user needs evolve and vendor offerings change through new technologies and changes to market conditions. A process of continuous improvement to the document should be established to capture new best practice and requirement values, based on Tier-0 and Tier-1 procurements. The lessons learnt should be fed into the toolbox of technical elements started here to provide a valuable resource for the European HPC community.

7 Annex

7.1 HPC Architecture Classification

Task 7.1 has produced an updated architecture classification [13] that is being adopted by this document. This classification builds on and refines the HPC architectures description used in 2008. The original system was based on a flat table of architecture names and this has proved too rigid because it did not take into account that definitions overlap and that certain supercomputer classes are only transitory because they mark a generation of system that introduces a new major feature. The new classification is based on a system, which consists of an ontology of parameters. An additional set of numerical values can help to identify architectures, which try to optimize a certain feature beyond the current standard, e.g. power consumption. The classification provides a set of commonly used architecture names and their definition based on the provided ontology. Thus, the new classification does not eliminate current language practice but helps to define these terms more precisely.

Homogeneous Cluster

Homogeneous multi-core CPUs, without on-chip vector units, with no or only moderate multi-threading, without off-chip accelerators, with an industry-standard interconnect, for which packet processing is done on the CPU or on the NIC, and with a full standard OS on the compute nodes.

Some examples follow:

IBM POWER-5/6/7	*Large Memory Size* (>= 2 GByte/core)
Bull MESCA	*Large Memory Size*
Bull INCA	*Small Memory Size*
SGI Altix	*Large Memory Size*
IBM MareNostrum	*Small Memory Size*

Heterogeneous Cluster

A variation on a homogeneous cluster with heterogeneous multi-core CPUs (2 or more types of cores).

An examples is:

IBM MariCel	*Small Memory Size*

Massively parallel system (MPP)

Variation on homogeneous cluster by deploying a custom interconnect with very high bandwidth, low latency and often specialised topology, which is able to handle the packet processing, and by using a customised operating system on the compute nodes.

Some examples follow:

IBM BlueGene/P	*Small Memory Size*
Cray XT5	*Large Memory Size*

Hybrid Systems

Examples are:

NEC SX-9 integrated with x86 processors *Integration of different properties within one single system such as very high memory bandwidth and very high per processor sustained performed (Vector) with general purpose processors*
X86 based system with accelerators Integration of general purpose processors with accelerators as “co-processors“

Massively multithreaded system MMT

MPP like vector systems that deploy a massively multithreaded CPU instead of a vector CPU.

7.2 Benchmark Suite

Deliverable D6.3.1 [15] has identified a set of representative applications for the scientific community that has considered the following aspects:

- Coverage of relevant application areas,
- Representative applications within the covered application areas,
- Coverage of (the range of) hardware platforms (prototypes) which are relevant for PRACE,
- Petascaling opportunities of benchmark codes with relevant datasets,
- Optimisation opportunities of benchmark codes.

These applications have been integrated into a benchmark suite, to help improve the benchmarking of PetaFlop/s systems. The list has been modified since D7.5.1 [26] was published following comments from the EU review of March 2009.

There follows a short summary of each of the application benchmarks:

ALYA: Finite element code for Large Eddy Simulation of compressible and incompressible flows.
Application area: Computational Fluid Dynamics
Languages: Fortran 90
Libraries: Metis
Programming model: MPI / OpenMP
I/O characteristics: read at start, write periodically

AVBP : Turbulent combustion + CFD code.
Application area: Computational Fluid Dynamics
Languages: Fortran 90
Libraries: HDF5, SZIP, METIS
Programming model: MPI
I/O characteristics: read at start, write periodically

BSIT: Computational geophysics code.
Application area: Computational Geophysics
Languages: Fortran 95, C
Libraries: Compression lib
Programming model: MPI /OpenMP
I/O characteristics: read at start, write periodically

Code_Saturne: General purpose CFD code, used for nuclear thermal-hydraulics process, coal and gas combustion, aeraulics, etc.
Application area: Computational Fluid Dynamics
Languages: Fortran 77, C99, python
Libraries: BLAS
Programming model: MPI
I/O characteristics: read at start, write periodically

CP2K: Package to perform atomistic and molecular simulations of solid state, liquid, molecular and biological systems. It consists of several components for classical molecular dynamics, ab-initio density functional theory. etc.
Application area: Computational Chemistry and Condensed Matter Physics
Languages: Fortran 95
Libraries: FFTW, LAPACK, ACML
Programming model: MPI

I/O characteristics: checkpoints and output, intense

CPMD: Parallelized plane wave/pseudopotential implementation of Density Functional Theory particularly designed for ab-initio molecular dynamics.
Application areas: Computational Chemistry and Condensed Matter Physics
Languages: Fortran 77
Libraries: BLAS, LAPACK
Programming model: MPI
I/O characteristics: no special

Elmer: Multi-physics engineering code.
Application area:
Languages:
Libraries:
Programming model:
I/O characteristics:

EUTERPE: A gyro kinetic particle-in-cell (PIC) code in a three dimensional domain in coordinates and two in velocities plus time. Its main target is to simulate the micro-turbulences in the Plasma core.
Application area: Plasma physics
Languages: Fortran 90 / C
Libraries: ESSL, PETC
Programming model: MPI
I/O characteristics:

GADGET: Code for cosmological N-body/SPH simulations on massively parallel computers with distributed memory.
Application area: Astronomy and Cosmology
Languages: C 99
Libraries: FFTW, GSL, HDF5
Programming model: MPI
I/O characteristics: no special

GPAW: Density-functional theory (DFT) code based on the projector-augmented wave (PAW) method. It uses real-space uniform grids and multi-grid methods.
Application area: Computational Chemistry and Condensed Matter Physics
Languages: C, Python
Libraries: BLAS, LAPACK
Programming model: MPI
I/O characteristics: read at start, write at end

GROMACS: Package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles.
Application area: Computational Chemistry and Life Sciences
Languages: C, assembler
Libraries: BLAS, FFTW, LAPACK
Programming model: MPI
I/O characteristics: read at start, write periodically, relaxed

HELIUM: Code to simulate the behaviour of helium atoms using time-dependent solutions of the full-dimensional Schrödinger equation.
Application area: Atomic Physics
Languages: Fortran 90
Libraries:
Programming model: MPI
I/O characteristics: read at start, write periodically

NAMD: Parallel molecular dynamics code for high-performance simulation of large biomolecular systems.
Application area: Computational Chemistry, Condensed Matter Physics, Life Sciences
Languages: C++
Libraries: Charm++, FFTW, TCL
Programming model: Charm++, MPI, master-slave
I/O characteristics: no special

NEMO: Numerical platform for simulating ocean dynamics and biochemistry, and sea-ice.
Application area: Earth and climate science.
Languages: Fortran 90
Libraries: NetCDF
Programming model: MPI
I/O characteristics: read at start, write periodically

NS3D: Code to solve the incompressible Navier-Stokes equations by Direct Numerical Simulation (DNS).
Application area: Computational Fluid Dynamics
Languages: Fortran 90
Libraries: EAS3, Netlib, (FFT)
Programming model: MPI + NEC-microtasking
I/O characteristics: read at start, write periodically

Octopus: A is a computer code to calculate excitations of electronic systems. The code relies on Density Functional Theory (DFT) to accurately describe the electronic structure of finite 1-, 2- and 3-dimensional systems, like e.g. quantum dots, molecules and clusters.
Application area: Electronic systems
Languages: Fortran 90 / C
Libraries: FFTW, BLAS, LAPACK, GSL
Programming model: MPI
I/O characteristics:

PEPC: Parallel tree-code for computation of long-range Coulomb forces. The forces are calculated based on the Barnes-Hut algorithm.
Application areas: Plasma Physics
Languages: Fortran 90
Libraries:
Programming model: MPI
I/O characteristics: read at start, write periodically

QCD: Particle physics multi-kernel QCD code.
Application area: Particle Physics
Languages: Fortran 90, C
Libraries:
Programming model: MPI
I/O characteristics: no special

Quantum ESPRESSO: is an integrated suite of computer codes for electronic-structure calculations and materials modelling at the nanoscale, based on density-functional theory, plane waves, and pseudopotentials..
Application area:
Languages: Fortran 90, Fortran 77, C
Libraries:
Programming model: MPI
I/O characteristics:

SPECFEM3D: Earthquake simulation code.
Application area:

Languages:
Libraries:
Programming model:
I/O characteristics:

TRIPOLI-4:
Application area:
Languages:
Libraries:
Programming model:
I/O characteristics:

WRF: Weather Research and Forecasting Model
Application area:
Languages:
Libraries:
Programming model:
I/O characteristics:

7.3 PetaFlop/s procurements in Europe

Since the last version of this document was released the first European Petaflop/s system has been installed, the IBM BlueGene/P extension at FZJ. It appears at number 3 in the June 2009 top500 list [22] with a theoretical peak of just over 1 Petaflop and a Linpack performance of 825 TFlop/s.

This system is fully consistent with the work done in WP7 during the PRACE project:

- MPP was recommended by D7.1.2 [30] as a suitable architecture for the representative applications selected by PRACE.
- The original Jugene system, with a peak performance of 222 TFlop/s was used as a PRACE prototype and informed the technical requirements and system sizing values of D7.5.1 [26].

7.4 Existing Systems Analysis

A survey of sizing values for the first 15 entries in the June 2009 Top500 list [22] is included in Table 22.

Top500 # June2009	Architecture	System	Manufacturer	Type	Country	Organization	Year
1	Hybrid cluster	RoadRunner	IBM	Opteron 2C+PowerXcell 8i	USA	LANL	2008
2	MPP	Jaguar	Cray Inc.	XT5 Opteron 4C	USA	ORNL	2008
3	MPP	Jugene	IBM	BG/P	Germany	FZJ	2009
4	Cluster / MPP	Pleiades	SGI	Altix ICE Xeon 54xx 4C	USA	NASA/Ames	2008
5	MPP	bgl	IBM	BG/L	USA	LLNL	2007
6	MPP	kraken	Cray Inc.	XT5 Opteron 4C	USA	NICS/Utennessee	2008
7	MPP	Intrepid	IBM	BG/P	USA	ANL	2007
8	Cluster	Ranger	Sun	Opteron 4C	USA	TACC/Utexas	2008
9	MPP	Dawn	IBM	BG/P	USA	LLNL	2009
10	Cluster	Juropa	Bull SA	Xeon 55xx	Germany	FZJ	2009
11	MPP	Franklin	Cray Inc.	XT4 4C	USA	LBNL	2008
12	MPP	Jaguar	Cray Inc.	XT4 4C	USA	ORNL	2008
13	MPP	RedStorm	Cray Inc.	XT3/XT4 2C/4C	USA	SNL	2008
14	BG	Shaheen	IBM	BG/P	Saudia Arabia	KAUST	2009
15	Cluster	Magic Cube	Dawning	Dawning 5000A Opteron 4C	China	Shangai SC	2008

Top500 # June 2009	System	Peak perf RPEAK TFlop/s	Linpack RMAX TFlop/s	RMAX/ RPEAK	Total RAM TByte	Cores	Local storage PByte	I/O throughput GByte/s	IC BW TByte/ s	Power MW	Linpack TFlop/s / MW	Memory / Peak perf (Byte / Flop/s)	Memory / Core (GByte)	Flops / Core (GFlop/s)
1	RoadRunner	1457.000	1105.000	0.758	98.000	129600	2.0	216.000	384	2.483	445	0.067	0.76	11.24
2	Jaguar	1381.000	1059.000	0.767	362.000	150152	10	284.000	532	6.951	152	0.262	2.41	9.20
3	Jugene	1002.700	825.500	0.823	144.000	294912	6	66.000		2.268	364	0.144	0.49	3.40
4	Pleiades	608.829	487.005	0.800	51.000	51200	0.3			2.090	233	0.084	1.00	11.89
5	bgl	596.378	478.200	0.802	49.100	212992	1.89			2.330	205	0.082	0.23	2.80
6	kraken	607.200	463.300	0.763	129.000	66000	3.3					0.212	1.95	9.20
7	Intrepid	557.056	458.611	0.823	80.000	163840	7.6	88.000		1.260	364	0.144	0.49	3.40
8	Ranger	579.379	433.2	0.748	123.000	62976	1.73	40.000		2.000	217	0.212	1.95	9.20
9	Dawn	501.350	415.700	0.829	147.500	147456	2.2			1.134	367	0.294	1.00	3.40
10	Juropa	308.283	274.800	0.891	79.000	26304	0.86	20.000		1.549	177	0.256	3.00	11.72
11	Franklin	355.506	266.300	0.749	76.560	38642	0.44	32.000		1.150	232	0.215	1.98	9.20
12	Jaguar	260.200	205.000	0.788	62.000	30976	0.6			1.581	130	0.238	2.00	8.40
13	RedStorm	284.000	204.200	0.719	78.750	38208	1.75	50.000	120	2.506	81	0.277	2.06	7.43
14	Shaheen	222.822	185.171	0.831	65.536	65536	1.9	16.000		0.504	367	0.294	1.00	3.40
15	Magic Cube	233.472	180.600	0.774	122.880	30720				0.700	258	0.526	4.00	7.60

Table 22: First 15 in Top500 June2009 – sizing values