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

CAPACITIES

PRACE

Partnership for Advanced Computing in Europe

Grant Agreement Number: RI-211528

D7.2
Systems compliant with user requirements

Final

Version:	1.0
Author(s):	G.Erbacci, C. Cavazzoni, CINECA
Date:	24.4.2008

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: D7.2	
	Deliverable Nature: Report	
	Deliverable Level: PU	**Contractual Date of Delivery:** 30 / April / 2008
		Actual Date of Delivery: 30 /April / 2008
	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: Systems compliant with user requirements: suitability of available architectures for application classes**	
	ID: D7.2	
	Version: 1.	**Status:** Draft / Final
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2003	
	File(s): PraceDeliverableTemplate.doc	
Authorship	**Written by:**	G. Erbacci, C. Cavazzoni
	Contributors:	D. Agudo Soto, M. Bull, S. Girona, E. Griffiths, J. Heikonen, V. Kolhinen, W.M. Lioen, J.P. Nominé, F. Robin, A. van der Steen, Stefan Wesner
	Reviewed by:	M. Schliephake, HLRS; D Erwin, FZJ
	Approved by:	Technical Board

Document Status Sheet

Version	Date	Status	Comments
0.1	03/April/2008	Draft	
0.2	10/April/2008	Draft	Revision of tables
0.3	15/April/2008	Final Draft	For PRACE QA
0.4	16/April 2008	Formal restructure	DE
1.0	24/April/2008	Final version	

Document Keywords and Abstract

Keywords:	PRACE, HPC, Research Infrastructure, Petaflop/s Systems, Challenge computational applications
Abstract:	This document presents an initial translation of the user requirements into architectures and configuration specifications for the European Petaflop/s systems. The analysis is based on the results of Deliverable D6.2.1 which presents the key requirements of grand challenge HPC applications across Europe. Deliverable D7.2 is organised into three main sections: first, the user requirements, coming from the applications, are analysed in terms of architecture specifications, trying to identify the ideal architectural characteristics for such applications. Then, the main architectural features of the HPC Systems that could represent production Petaflop/s systems in 2009/10 are briefly underlined. Finally, an initial match of the main applications analysed to the HPC architectural classes is attempted and some considerations and remarks are presented.

Copyright notices

Table of Contents

List of Tables

References and Applicable Documents

[1] PRACE - Grant Agreement N. RI-21528- Annex1: DoW. http://www.prace-project.eu.

[2] HET - European HPC initiative. *The Scientific Case for a European Super Computing Infrastructure*, HET Report, www.hpcineuropetaskforce.eu, 2007.

[3] PRACE, Deliverable 6.2.1. *Preliminary report on application requirements*, March 2008.

[4] PRACE, Deliverable 7.1.1. *Initial recommendation for the selection of prototypes*, March 2008.

[5] Aad van der Steen, *Overview of recent supercomputers*, July 2007.

[6] John Shalf et al., *Investigation of leading HPC I/O performance using a scientific-application derived benchmark*, SC07 proceedings, 2007.

List of Acronyms and Abbreviations

BQCD	QCD code from the Konrad-Zuse-Zentrum für Informationstechnik Berlin.
BW	Bandwidth
ccNUMA	cache coherent Non-Uniform Memory Access
CFD	Computational Fluid Dynamics.
CPMD	Ab-inito Car-Parinello MD code.
CPU	Central Processing Unit.
DEISA	Distributed European Infrastructure for Supercomputing Applications. EU project by leading national HPC centres.
DL_POLY	General purpose MD code, developed at Daresbury Laboratory.
ECHAM5	General circulation model for climate research.
Fenfloss	Code for the simulation of incompressible flows.
FPGA	Field Programmable Gate Array
GADGET2	Cosmological simulations code.
GÉANT	Collaboration between National Research and Education Networks to build a multi-gigabit pan-European network, managed by DANTE. GÉANT2 is the follow-up as of 2004.
GENE	Gyrokinetics code for the simulation of plasma turbulence.
GF	GigaFlops or GFlop/s, i.e. one billion - 10^9 - floating point operations per second (usually now understood for computation with 64 bits of precision)
Gig/E	Gigabit Ethernet
GPFS	General Parallel File System
GPU	Graphical Processing Unit
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
HT3	HyperTransport 3.0
IB	InfiniBand
I/O	Input /Output.
IQCS	Code for the simulation of a quantum computer.
MD	Molecular dynamics.

MPI	Message Passing Interface. A library for message-passing programming.
MPP	Massively Parallel Processing (or Processor)
NAMD	MD code for simulation of large biomolecular systems.
NEMO	Nucleus for European Modelling of the Ocean. Oceanographic code.
NFS	Network File System
OpenMP	Open Multi-Processing. An API for shared-memory parallel programming.
NUMA	Non-Uniform Memory Access
OS	Operating System
OSS	Object Storage Servers
PRACE	Partnership for Advanced Computing in Europe; Project Acronym.
PEPC	Pretty Efficient Parallel Coulomb solver. Plasma physics code.
QCD	Quantum chromodynamics.
QDR	Quad Data Rate
QPI	Quick Path Interconnect
QuantumESPRESSO	Ab-inito MD code (also QE).
RAMSES	Adaptive mesh refinement code for astrophysical fluid dynamics.
RISC	Reduced Instruction Set Computer
SMP	Symmetric Multi-Processing
SU3_AHIGGS	Lattice QCD code for research into the conditions of the Early Universe.
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.
UMA	Uniform Memory Access
VLIW	Very Long Instruction Word

Executive Summary

The Partnership for Advanced Computing in Europe (PRACE) [1] has the overall objective to prepare for the creation of a persistent pan-European HPC service. PRACE is divided into a number of inter-linked work packages and WP7 focuses on Petaflop/s systems available by 2009/2010.

The primary goal of PRACE WP7 is: identify architectures and vendors capable of delivering Petaflop/s systems by 2009/2010; translate user requirements into architecture and configuration specifications; define technical requirements and evaluation criteria for Petaflop/s systems in 2009/2010 and define installation requirements for Petaflop/s systems and evaluate consistency with possible hosting sites.

Task 7.2 is in charge of translating user requirements into architecture and configuration specifications. This work is based on the analysis of the requirements of application codes across Europe, captured by WP 6 in Task 6.2.

Task 7.2 contributes to the definition of the main macro characteristics of the systems to be installed in 2009/2010 in order to achieve the performance required from challenging computational applications.

The initial result of Task 7.2 is this document (D7.2). Due to timing constraints the applications considered in Task 6.2 and consequently also here are based on the set of applications used within the DEISA project. This was necessary in order to deliver input for WP2 for the selection of prototypes and WP5 for their deployment already by Month 4. As a result this deliverable provides only the macro characteristics of the systems to be installed based on current applications and needs further input from WP6 about new applications in order to finalise the assessment at a later stage. The *intermediary* results derived from the current limited application basis are presented in a tabular form in Table 1. Based on the final analysis by Task 6.2 this document will be reviewed and updates will be incorporated into the technical specification that will be produced by month 11 as deliverable D7.5.1.

1. Introduction

Structure of the report

After this introduction, the report is organised in three sections: the next section extracts the user requirements coming from the fourteen applications analysed initially in Deliverable D6.2 and attempts a definition of the ideal or best architecture characteristics for these applications.

Section 3 outlines the main architectural features of the HPC systems that characterise the production Petaflop/s systems available in 2009/10. Finally, in Section 4, a mapping of the key applications analysed in Section 2 to HPC architectural classes is attempted. The results are presented in a table to provide an immediate view and understanding of the applications in terms of their architectural requirements.

2. Mapping User Requirements to Architecture Specifications

PRACE is working towards a pan-European HPC infrastructure with a number of Tier-0 Petascale systems, running from 2010 onwards and providing the computational resources for the major Grand Challenges in capability computing [1].

Advanced supercomputers supporting computational science are fundamental to the investigation of very complex scientific phenomena, permitting the development and testing of new, more quantitative and predictive theories, and enabling breakthrough science. Capability supercomputers will allow scientists to construct a hierarchy of models, where each is founded on the characteristics computed at the lower scale.

The computational disciplines [2] that will benefit most from capability computing are primarily, but not exclusively:

- weather, climatology and earth sciences;
- astrophysics, high-energy physics, and plasma physics;
- material sciences, chemistry and nano-sciences;
- life sciences;
- engineering.

All these disciplines traditionally use specific and representative application codes that are well known to the scientific communities. These applications have specific requirements in terms of architectures to meet the needs of European researchers.

The objective of this section is to map user requirements derived from computational application classes to architecture specifications, in the context of future Petascale systems.

User requirements are captured from the pool of the fourteen well known applications analyzed in deliverable D6.2.1 [3]. These applications codes cover the following scientific disciplines:

Life Sciences:

- DL_POLY: a general purpose molecular dynamics simulation package.
- NAMD: parallel molecular dynamics code for high-performance simulation of large biomolecular systems.

Material Sciences

- CPMD: a parallelized plane wave/pseudopotential implementation of Density Functional Theory, particularly designed for ab-initio molecular dynamics.
- QuantumESPRESSO: a material sciences code using ab-initio total energy and molecular dynamics calculations based on plane waves and pseudopotentials.

Earth Science

- NEMO: numerical platform for the ocean (dynamics and biochemistry) and the sea-ice simulations.
- ECHAM5: the 5th generation of the ECHAM general circulation model.

Plasma & Nuclear physics

- GENE: a gyrokinetics code for the simulation of plasma turbulence.
- PEPC: Pretty Efficient Parallel Coulomb-solver; based on a generic Barnes-Hut tree algorithm for computing long-range forces.

QCD

- BQCD: a quantum chromodynamics code from the Konrad-Zuse-Zentrum fuer Infomationstechnik Berlin.

- SU3_AHIGGS: a lattice quantum chromodynamics code intended for computing the conditions of the Early Universe.

Astrophysics

- RAMSES: an Adaptive Mesh Refinement code for astrophysical fluid dynamics.
- GADGET2: a code for cosmological N-body/SPH simulation.

CFD

- FENFLOSS: a code for the simulation of incompressible laminar and turbulent flows, using Reynolds-averaged Navier-Stokes-equations on unstructured grids.

Quantum Computing

- IQCS: part of a larger software package developed to simulate ideal operations of a quantum computer on a classical computer.

The architecture specifications onto which the applications will be mapped are derived from the survey of vendors and their offerings performed by Task 7.1 and documented in deliverable D7.1.1 [4]. Since the information obtained from the vendors is covered by an NDA, these specifications are therefore only expressed using a general classification. Nevertheless, the classes of specifications used here cover the essential architectural characteristics of forthcoming Petascale machines.

The mapping procedure has being carried out by first identifying the architecture specifications and then analysing the requirements of the user applications documented in D6.2.1. Architecture specifications that have been considered are divided into four main categories:

- CPU;
- Internal Network;
- I/O;
- Memory.

The CPU category is further subdivided into Commodity (Intel x86-64, AMD x86-64, PowerPC), Superscalar (Power6 and Itanium), Vector, and Accelerators (i.e: GPUs, Cell, FPGA).

Network specifications are subdivided into Low Latency, High Bandwidth, possibility to have optimized global communication, and possibility to overlap communications.

I/O is subdivided into Global low performance file system (ex: NFS), Global high performance file system (ex: GPFS, LUSTRE,…).

Finally, four subcategories for the memory subsystem have been identified: Memory subsystem with a Low Bandwidth to Flop/s ratio, Memory subsystem with a High Bandwidth to Flop/s ratio, Memory subsystem with low capacity (memory size) to number of CPUs ratio, and Memory subsystem with high capacity to number of CPUs ratio.

The results of the analysis are summarized in Table 1. Each row represents a different application while the columns represent the architecture specifications. Each cell represents a possible match between an application and an architecture specification. The match is represented by way of a colour scheme using three different colours, in order to provide an immediate visual interpretation for the reader:

- Green means that the application has an *high fit* or *requires the architecture specification*;
- Yellow means that application has a *moderate fit* or *may benefit from the architecture specification*;
- Grey means that the application has *low fit* or *does not require the architecture specification.*
- A blank cell means *no information is available.*

A comment field has been added for each of the four main categories to point out user requirements that cannot be captured with the colour scheme, or to add some more information about the mapping.

	CPU					Network					I/O			Memory				
	Commodity	Superscalar	Vector	Accelerators	Comments	Low Latency	High Bbw	Global op	Overlap comm	Comments	Slow global filesystem	Fast global filesystem	Comments	Low bw/flops	High bw/flops	Low GB/procs	High GB/procs	Comments
DL_POLY	🟨	🟩	▒	▒	(1)	🟩	🟨	🟩	🟨		🟨	🟩		🟨	🟨	🟩	🟨	(2)
NAMD	🟨	🟩	🟨	🟨		🟨	🟨	🟩	🟨		🟩	▒		🟩	▒	🟨	🟨	
IQCS	🟩	🟨	▒	▒	(3)	🟩	🟩	▒	🟩	(4)	🟩	▒	(5)	▒	🟩	🟩	▒	
CPMD	🟨	🟩	🟨	🟨		🟩	🟨	🟩	🟨	(6)	🟨	🟩	(7)	▒	🟩	🟩	🟨	(8)
QE	🟨	🟩	🟨	▒	(9)	🟩	🟨	🟩	🟨		▒	🟩		▒	🟩	🟨	🟩	
GENE	🟨	🟩	▒	🟨	(10)	🟨	🟨	🟩	🟨		🟩	▒	(11)	▒	🟩	🟩	▒	
PEPC	🟩	🟨	▒	▒	(12)	🟩	▒	🟩	🟩	(13)	🟩	▒	(14)	▒	🟨	🟩	▒	
BQCD	🟩	🟨	▒	▒	(15)	🟩	🟩	🟩	▒	(16)	🟩	🟨		▒	🟩	🟩	▒	(17)
SU_AHIGGS	🟨	🟩	🟨	▒	(18)	🟨	🟨	🟨	🟨		▒	🟩		▒	🟩	🟨	🟩	
NEMO	🟨	🟨	🟩	▒	(19)	🟩	🟨	🟨	🟨		▒	🟩	(20)	▒	🟩	🟩	🟨	
ECHAM5	🟨	🟩	🟩	▒		🟨	🟩	🟩	🟨		▒	🟨		▒	🟨	🟨	🟩	
RAMSES	🟨	🟩	🟨	🟨		🟩	🟩	🟩	🟨		▒	🟩		🟨	🟩	🟨	🟩	
GADGET2	🟩	🟩	▒	🟨	(21)	🟨	🟨	🟨	🟩	(22)	🟩	▒	(23)	🟨	🟨	🟨	▒	
FENFLOSS	🟨	🟩	🟩	▒	(24)	🟩	🟨	🟩	🟨	(25)	🟨	🟩		▒	🟩	🟩	🟨	

Table 1 : Mapping of user requirements to architecture specifications

Comments:

DL-POLY

(1) CPU: Not amenable to vectorisation
(2) Memory: Low memory requirement typical of MD codes

IQCS

(3) Memory: No experience on vector machines / or accelerator usage:
(4) Network: Ability to overlap possibly beneficial
(5) I/O: I/O not critical for this code therefore all are "not required"

CPMD

(6) Network: Global operations for 3D FFTW
(7) I/O: Required for restart files
(8) Memory: Scaling is problem dependent. Using "taskgroup parallelization", it scales to 16-32K processors on a BlueGene System.

QE

(9) CPU: No experience on accelerator usage

GENE

(10) CPU: Peak flops is the main issue, use of accelerator could help
(11) I/O: I/O not critical for this code therefore all are "not required"

PEPC

(!2) CPU No experience on accelerator usage
(13) Network: Main issue is synchronization
(14) I/O: I/O not critical for this code therefore all are "not required"

BQCD

(15) CPU: Peak flops is not an issue
(16) Network: Bandwidth and latency are the main issues
(17) Memory: Memory bandwidth, latency and cache are important

SU_AHIGGS

(18) CPU: No experience on vector machines

NEMO

(19): CPU: Tuned for vector systems
(20) I/O: I/O performance critical for this application

GADGET 2

(21) CPU: Peak flops is the main issue, use of accelerator could help
(22) Network: Ability to overlap possibly beneficial
(23) I/O: I/O not critical for this code therefore all are "not required"

FENFLOSS

(24) CPU: Vectorised version exists
(25) Network: Requires fast global reductions

3. Overview of HPC Architectures

In this section, we give a quick overview of the HPC Systems that could become production Petaflop/s solutions in 2009/10. We use some material from Deliverable D7.1.1 (Initial recommendation for the selection of prototypes and first estimates of costs of Petaflop/s class systems) [4]. The information provided below is mostly public or globalized in ranges, so that we do not reveal any explicitly confidential vendor information.

Architectures – general classification

All systems likely to scale up to Petaflop/s are composed of a large number of interconnected processing units (consisting of cores and memory). They are themselves connected to some file system for data handling. We mostly distinguish the possible architectures by the nature of these computing elements – often called nodes:

- MPP systems consist of a very high number of small elements. They are mostly characterised by the fact that the compute nodes run a reduced dedicated kernel to minimise OS jitter and by having interactive access and I/O handled through dedicated interactive and I/O nodes with full OS kernels. Examples are the Cray XT4/5 and the IBM BlueGene/L and BlueGene/P.
- Thin node clusters use elements with a small amount of computing, memory and I/O resources - typically one or two processors with a shared memory. The processors themselves follow the general trend and tend to become more and more multi-core. The difference between thin node clusters and MPP systems is sometimes blurred; thin node clusters are typically based on commodity components, whereas MPP systems generally rely on more customised integration. This holds particularly for the interconnection networks. Examples of thin node machines are the Bull Novascale (and following generation) and SGI ICE systems.
- Fat-node clusters have nodes with a large amount of computing, memory, and I/O resources; although there is no formal definition or threshold, 16 cores is a typical size beyond which a node is considered fat; fat nodes can exhibit different kinds of memory access behaviour~~s~~, e.g. SMP, NUMA, ccNUMA. There are examples of ccNUMA machines among the Bull and SGI next generation systems while the IBM POWER7 cluster is an example of a system with SMP based fat nodes.
- Vector systems use vector processors, able to stream - or pipeline - operations on arrays, with high memory-to-processor bandwidth. It is expected that no more than two vendors will continue to offer such systems: Cray (the X2 system) and NEC (the SX-9 follow-on system).
- Hybrid systems use a combination of two or more types of units (scalar nodes, vector units, and possibly units with accelerators such as for instance GPUs – Graphical Processing Units - or FPGAs – Field Programmable Gate Arrays). Several vendors expect to have such hybrid systems available by 2010. Cray XT5h is an available current intermediate hybrid system, combining FPGA and vector processors.

Components: CPUs, computing nodes, network, I/O and file systems

For this initial matching of application requirements to architectures and technology, we limit ourselves to components and systems that are already existing or considered to be likely available by 2009/2010. Further extrapolations are not in our current scope.

In the following tables we give indications on important features and parameters for different categories of technological components, with ranges for figures whenever possible:

- CPU: clock cycle, floating point performance, CPU/memory bandwidth (BW) in Table 2;
- Computing nodes/servers: number of cores, peak performance, memory size, communication BW in Table 3;
- Network: latency, BW in Table 4.

File systems also deserve some specific comments which are given in a dedicated paragraph.

All figures or ranges should be considered as estimates and/or orders of magnitude, since they may vary – upon time, depending on the technology used by providers or system vendors, or depending on the precise characteristics of specific element (e.g. host bus adapter of a network).

For CPUs we distinguish:

- Commodity processors (x86-64) , PowerPC)
- True 64-bit RISC, VLIW processors (Power6, IA64)
- Vector processors
- Accelerators such as GPUs, FPGAs, Cell BE

Important parameters are:

- The clock cycle
- The performance (abbreviation: GF = Gigaflop/s)
- The bandwidth between memory and processor.

CPU (socket)	**Clock cycle**	**No. of cores/node**	**Bandwidth (GB/s/core)**
Commodity	1 to 3+ GHz	≤ 8	HT3: 16; QPI: 34
Superscalar	Up to 5 GHz	≤ 64	POWER6: 37.5; IA64: 34
Vector	Cray X2: 1.6 GHz; NEC SX-9: 3.2 GHz	Cray X2: 4 NEC SX-9: ≤ 16	Cray X2: 25.6 NEC SX-9: 256
Accelerators	210 MHz to 3 GHz	96 - 128	25.6 - 6.8

Table 2 : Types and characteristics of CPUs

For computing units (“servers” or “nodes”), we try to give indications on typical characteristics:

- Number of cores (abbreviation: *x*S*y*C means *x* sockets with *y* cores each)
- Peak performance
- Maximum memory per node (regardless of the organization UMA, NUMA…)
- Memory bandwidth

Type of node	**Number of cores**	**Peak performance (Gflop/s)**	**Max memory (GB)**	**Communication between Nodes BW (GB/s)**
MPP	4 - 8	13.6 - 55 GF	≤ 16 GB	≤ 7.6
Thin node	8 - 16 2S4C - 2S8C	100 - 200 GF	64 GB	Same range as MPP
Fat node	16-128 (4 - 16 sockets)	256 GF	1 - 4 TB	Up to 230 (aggregated)
Vector	Up to 16	Up to 1.6 TF	Up to 1 TB	Up to 128 bi-directional
Hybrid	not known/ configuration dependent	not known/ configuration dependent	not known/ configuration dependent	not known/ configuration dependent

Table 3 : Types and characteristics of computing nodes

It is noticeable that hybrid systems are difficult to characterise. The number of cores is not necessarily a relevant parameter here. Memory and communication bandwidth may vary considerably.

Only the peak performance can be estimated in some cases: GPUs are now delivering some 100 GF in single precision and should soon deliver the same in double precision. The same holds for IBM’s Cell/Opteron Triblade units (4 Opteron cores + 4 Cell chips resulting in 400 GF).

Networks/interconnect

In a large distributed memory machine the interconnection between many hundreds or thousands of compute nodes has two facets: network topology and network technology.

Network technologies range from inexpensive but not very effective – to more high-end, proprietary but more expensive networks. The first category is typically Gigabit Ethernet (Gig/E) which is still widespread if not dominant in Top500 machines: We will not consider Gig/E further. Examples for the second category are: QsNet and Myrinet which are vendor independent, whereas the SeaStar2 network is Cray proprietary. Somewhere in between lies InfiniBand (IB) which can be regarded as quite cost effective with good (high) bandwidth and good (low) latency. Proprietary networks can further decrease latency.

In the following table, we give some figures taken from Overview of recent supercomputers, Aad van der Steen, July 2007 [5], based on measures performed in 2007, they provide an idea about the possibilities of the different technologies. Looking in more detail, actual performance can also be affected by many factors such as exact technical characteristics of the components (e.g. host bus adapter). These figures are only indicative.

	Bandwidth (GB/s)	Latency (µs)
Cray SeaStar2*	2.1	4.5
IBM Infiniband**	1.2	4.5
Infiniband**	1.2-1.3	4.0-4.5
Myrinet 10G	1.2	2.1
Quadric QsNet^{II}	0.9	2.7
SGI NumaLink4	2.7	1.2
NEC IXS (SX9)***	16	2

* Cray SeaStar2+ is supposed to improve performance by 30%.

** Infiniband quad-data rate should be able to double the bandwidth. Links can be aggregated in units of 4, 8, or 12, called 4X, 8X, or 12X. The speed of 4X QDR IB is supposed to be around 3.2 GB/s.

*** Can be extended up to 128 GB/s. The chosen value is seen as a typical installation choice.

Table 4 : Network technologies characteristics

It is not expected that networks latency will significantly decrease in a near future, at least for the present technologies.

The three main classes of networks topologies are hypercube, torus, and fat-tree. For an introduction to these concepts see [5].

A priori, topology and technology are orthogonal notions, but in practice IB is often proposed in a fat-tree topology whereas the other topologies are often implemented through proprietary networks which are typically found in MPP or vector systems. But there is no absolute rule.

Fat-tree has a priori the advantage of cost, but tends to be less flexible with increasing number of nodes (if the connections are saturated, adding top-level switches will be required resulting in a possibly significant re-cabling). By contrast, a torus topology is more flexible, nodes can be locally added with a linear cost function.

I/O and file systems

For operation of capability Petaflops/s systems we consider only the following parallel global file systems to be relevant: Lustre (by SUN/CFS), GPFS (by IBM), pNFS (an extension of NFS V4 for clusters), or PanFs (by Panasas).

These file systems are either proprietary (GPFS, PanFS) or open source (Lustre, pNFS) but supercomputer vendors tend to be agnostic with respect to the file system choice. All seem to be open to integrating any file system or at least are able or willing to propose other options when they have a proprietary preferred solution (e.g. there are Top50 IBM machines with Lustre instead of GPFS).

It is difficult to rank file systems features since their behaviour is highly dependent on the machine architecture and on the type and number of components dedicated to I/O, especially the number of I/O nodes and the interconnect. A good example of how to try to quantify this is shown in “Investigation of leading HPC I/O performance using a scientific-application derived benchmark” by John Shalf et al., SC07 proceedings [6].

As of today, several existing Top500 machines have an aggregated disk bandwidth of several 10 GB/s, and we can consider nearly-existing ones (2008/2009) to have bandwidth of several 100 GB/s.

The disk storage size by itself is not really an issue, the total capacity is mainly limited by costs.

So file systems seem not to be a very distinctive issue by themselves, since their most important parameter, the overall bandwidth to storage, is mainly determined by the number of nodes dedicated to running the data management processes (e.g. Lustre OSS). Together with costs and the overall balance of the machine architecture, network bandwidth and software scalability – to be able to exploit several thousands of I/O nodes - seem to be the real bottlenecks to scaling file systems much further. For instance, Lustre roadmaps indicate the objective of 10 TB/s bandwidth (with V3.0 of the software) in 2009.

Summary of architectures

Table 5 gives a consolidated schematic view of architectures with their main features ranked qualitatively, with the following classification. Caution: low network latency is marked as High/Good!

▨	Low (poor) to moderate
🟨	Medium to high
🟩	High (good)
⬜	Not know/configuration dependent

The table tries to provide average ranking or trends. Hybrid systems are especially difficult to qualify. What can be said about them is that they exhibit a very high CPU potential, as long as applications can be adjusted to using the specialized units.

Accelerators used as co-processors deserve a special mention. Using accelerators can be seriously compromised by memory-CPU bandwidth and I/O bandwidth. Their use is relevant if applications can be organized with large amounts of computations localized on the acceleration units with little data movement to main memory. We can certainly consider it is worth using accelerators if the application speedup is more than a factor of ten compared with CPUs.

	MPP	**Thin node cluster**	**Fat node cluster**	**Vector**	**Hybrid**
Example systems	IBM BlueGene Cray XT5	SGI ICE Bull Novascale and successor	IBM Power6,7 Bull/SGI next generation clusters	Cray X2 NEC SX8/9	Cray XT5h Cray Baker IBM Cell/Opteron NEC SX9/x86
Flops/CPU	▨	🟨	🟨	🟩	🟩
Memory/CPU	▨	🟨	🟩	🟩	⬜
BW/Flops	🟨	🟨	🟩	🟩	🟨
Network latency	🟩	🟨	🟨	🟩	🟨
Network bandwidth	🟩	🟨	🟨	🟩	🟨
I/O	⬜	⬜	⬜	⬜	⬜

Table 5 : Characteristics of architectures

4. Translate User Requirements into Architectures

In this section the intermediate results of the analysis of user requirements based on the application set, investigated in Section 2, and the characterisation of the architectures, performed in Section 3, are combined to obtain a mapping between the user requirements and the architectural classes.

Again, user requirements are expressed through user applications, which are currently representative for the load of HPC systems installed in different countries. The results are summarized in Table 6.

The rows of the table represent the same user applications described in Section 2 and the columns represent the classes of architectures, identified in Section 3.

An additional column contains comments useful to clarify the choice of the mapping or to express application requirement not captured by this simplified schema, but useful for the prototype selection.

To express how user requirements are mapped into architecture classes the same colour code of the previous sections is used. In particular the colour of the cells means:

Green: the corresponding application has a high fit with the corresponding architecture class,

Yellow: the application has a moderate fit with the architecture class,

Grey: the application has a low fit with the architecture class,

Blank cell: means that no information on the mapping between the application and the architecture class is available.

This assessment has to be handled with care as it has been already mentioned above the assessment is based on the limited set of applications analysed within WP6 at PM3 and reflecting more or less the current status of deployment and do not intend to fully predict nature and requirements of applications in the future. WP7 will re-perform this assessment based on future input from WP6. In particular coupled applications following paradigms such as the Virtual Physiological Human, Virtual Airplanes or similar are not considered at all that would naturally also demand hybrid computing systems.

	MPP	Thin node cluster	Fat node cluster	Vector	Hybrid …	Comments
DL_POLY	🟨	🟩	🟩	▨	🟨	Code has not been ported to vector machines. Hybrid architectures potentially useful, but low interest from code authors: they want to keep source portable.
NAMD	🟩	🟩	🟨	▨	🟨	Code requires good network performance
IQCS	🟩	🟩	🟨	▨	▨	Code requires good network memory performance, but not memory capacity
CPMD	🟩	🟨	🟩	🟨	▨	MPP fit is for "task group parallelization" CPMD has recently been ported to Cell, performance not jet well known
QE	🟨	🟩	🟩	🟨	▨	If memory per core is too low QE does not fit well
GENE	🟩	🟩	🟨		🟨	Better with peak core performances: accelerators may help
PEPC	🟩	🟨	🟨	🟨		Code requires not much memory, but does require network performance
BQCD	🟩	🟨	🟨			Code requires very good network rather than peak flops
SU_AHIGGS	🟩	🟨	🟨	▨	▨	
NEMO	🟨	🟩	🟩	🟩	▨	Vectorised version exists. Not likely to be suitable for hybrid architectures, as it lacks a compact kernel.
ECHAM5	🟩	🟨	🟨	🟩	▨	A vectorized version has been presented recentely
RAMSES	🟩	🟩	🟨	▨		Peak flop performance and good memory access is required
GADGET2	🟨	🟩	🟩			Code requires peak flop performances, good memory performance and large memory capacity
FENFLOSS	🟨	🟨	🟨	🟩	🟨	Vectorised version exists. Kernel is CG, so there is potential for accelerator use.

Table 6 : Preliminary mapping of application user requirements on architectures

5. Conclusions and final remarks

The objective of PRACE Task 7.2 is to contribute to defining the main macro characteristics of the systems to be installed in 2009/2010, in order to achieve the performance required from grand challenge computational applications. Deliverable D7.2 has the aim of translating an initial set of user requirements based on well established applications into architecture and configuration specifications.

This preliminary set of requirements and their mapping to architectures are synthesized in Table 6.

It is important to remark that many applications have a huge range of possible execution modes, from scalar mode to hyper parallel mode. Here we consider the application in the execution regime currently typical for European top HPC centres, as investigated in [3].

To ensure that requirements can be considered in a timely fashion in WP5 and WP2 for the selection of prototypes, deliverable D7.2 has been produced by Month 4, when the activity of Task 6.2 is still in an early stage. For this reason the deliverable provides only the macro characteristics of the systems to be installed, presented in a tabular form.

Based on the final analysis by Task 6.2 this document will be reviewed and updates will be incorporated into the technical specification that will be produced by month 11 as deliverable D7.5.1.

Some preliminary remarks can nevertheless be made about Table 6, which are quite qualitative at this stage and that would require further investigation and quantification, very likely by WP6 and WP7 together.

Hybrid systems are a quite new approach compared to other type of technologies and it is not surprising that the considered applications are not prepared for their immediate exploitation. However as PRACE is supposed to look into potential future types of architectures they must not be discarded. It was not easy to compare hybrid systems with homogeneous approaches and potential applications for hybrid systems need more time and effort in order to allow a proper analysis and quantification of the potential benefits of porting them to one or more kinds of accelerators. WP8 is expected to further evaluate technologies for hybrid computing.

Vector and fat-node architectures are suitable for several codes.

Thin-node systems can be regarded as near-to-MPP and even candidates for hybridisation, with GPU or FPGA units for instance, which shows the difficulty to clearly differentiate between the chosen categories. These two categories are clearly priorities for the selection of prototypes to be achieved by PRACE by mid-2008, but the other categories must also be tentatively represented to further refine technology and application behaviours.

In summary, the analysis of applications shows that there is no single architecture that is perfectly suited for all classes of applications. In particular the results must be seen as preliminary as the deliverable from WP6 used as input is only the preliminary report on application requirements. Considering a potentially larger scope for the applications including for example also eHealth applications a futher diversification of application requirements can be expected. Consequently PRACE should offer a variety of promising architectures and explore novel systems even at an early stage as the current development in hardware architectures is seen as very volatile.