



PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE

**Grands calculateurs pour le calcul scientifique
Quelques expériences technologiques et perspectives dans le
cadre du projet européen PRACE**

J.-Philippe Nominé, CEA-DAM-DIF-DSSI - J.-Marie Normand, CEA-DSM-IPhT



Grands calculateurs pour le calcul scientifique. Quelques expériences technologiques et perspectives dans le cadre du projet européen PRACE.

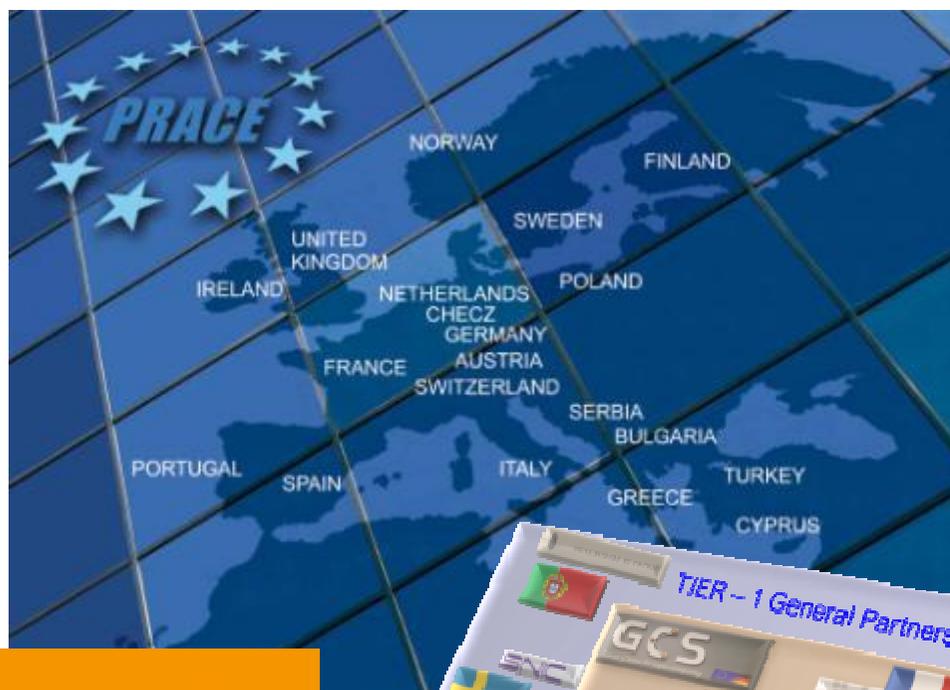
Plan

- **1. Quelques mots sur PRACE : objectifs et situation**
- **2. Quelques expériences technologiques : “Prototypes”**
- **3. Quelques remarques finales**

Grands calculateurs pour le calcul scientifique Quelques expériences technologiques et perspectives dans le cadre du projet européen PRACE

- PRACE, depuis 2008, prépare la mise en place d'un ensemble de supercalculateurs de puissance, de plus d'un petaflops (10^{15}), pour la simulation numérique à grande échelle, au profit des chercheurs européens.
- Cette infrastructure européenne va démarrer courant 2010 avec deux premiers supercalculateurs, en Allemagne puis en France, et lancement d'appels à projets pour les allocations d'heures de calcul.
- Ce projet a été et sera l'occasion de différents travaux techniques préparatoires et exploratoires, sur les architectures et les technologies matérielles et logicielles utiles au calcul intensif.
- Nous en donnons un aperçu dans cette présentation, avec les résultats obtenus collectivement depuis 2 ans, assorti de quelques remarques sur les tendances observées et les difficultés et incertitudes qui subsistent.

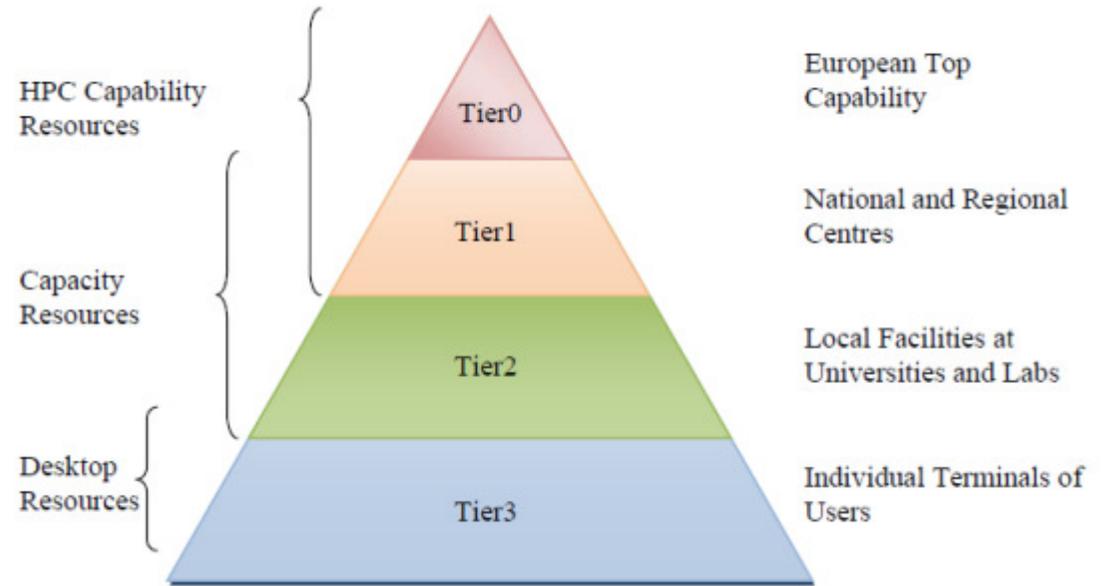
PARTNERSHIP
FOR ADVANCED COMPUTING
IN EUROPE



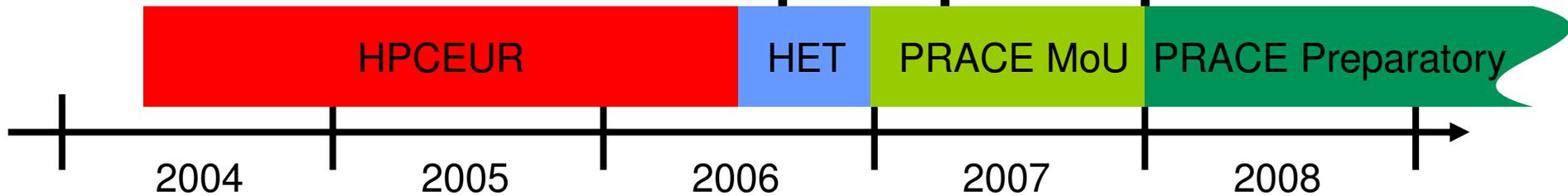
- Soutenu par FP7
DGINFSO
FP7 INFRA-2007-2.2.2.1
Grant RI-211528
- Objectif = une RI
Au sens feuille de route ESFRI
Calculateur ↔ Grand instrument
- 20 pays membres désormais
EU + partenaires 'proches'
- France représentée par GENCI
CEA, CNRS, CPU, INRIA



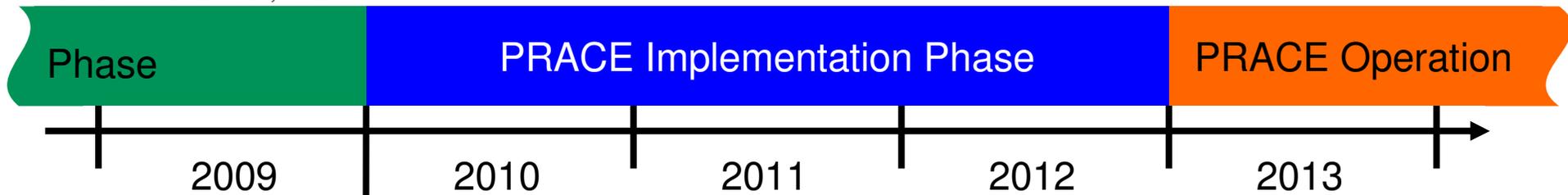
- **Objectif**
 - 4-6 centres de calcul « petascale
 - A partir de 2010
- **Candidats à l'accueil**
 - Allemagne (Gauss)
 - France (GENCI-CEA)
 - Espagne (BSC)
 - UK (EPSRC)
 - Pays-Bas (NCF-SARA)
 - Italie (CINECA)
- **France**
 - Site CEA-TGCC retenu par GENCI en accord avec CEA et CNRS
- **Articuler tier-0/tier-1**



PRACE History and first steps



EU-Grant: INFSO-RI-211528, 10 Mio. €



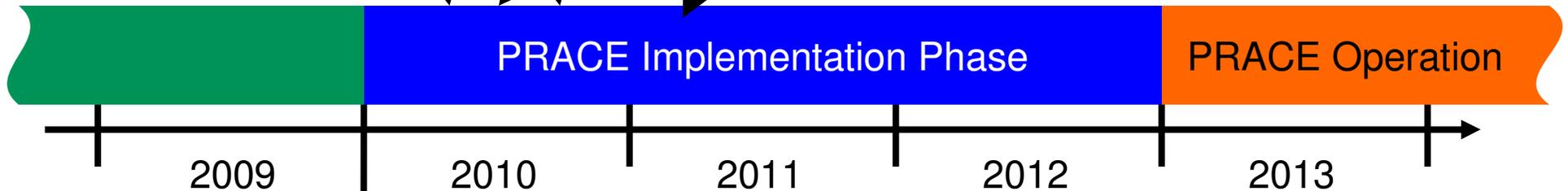
**Tier-0 centres providing HPC-capability
service in a legal entity**



Avril = Statuts signés
AISBL Bruxelles

2 appels à projets
BG/P FZJ (Mai, ...)

Machne No2
CEA TGCC
(GENCI)



2009

2010

2011

2012

2013

EU-Grant: PRACE 1IP 20 Mio. €



Modalités techniques d'accès prévues

GEANT (réseau 10 Gbit/s)

Middleware issu de DEISA

UNICORE, gsi-ssh, GridFTP, certificats X509...

Approche par « prototypes »

- PRACE 2008/2009 = préparation de RI
 - Organisationnel, juridique, politique, financement, ...
 - Ne pas rester « inactif » techniquement pendant ce temps
 - **Applications et benchmarks, prototypes, lien entre les deux**
 - **Effort important également en formation (Ecoles, Porting Workshops etc.)**

Approche par « prototypes »

- Prototypes: une définition « étendue »
 - **Systèmes complets pré-production et « early implementation »**
 - Tous types de test hardware (hw), software (sw), opérations, système etc. en prémisses des choix de premières machines en 2010
 - Du temps sur de grandes machines de production déjà en place, de nouvelles machines dédiées plus petites ...
 - **Composants et autres systèmes avancés**
 - R&D, exploration/expérimentation à moyen terme hw et sw

Prototypes for Petaflop/s systems in 2009/2010 (pre-production)

PRACE Prototype Access

<http://www.prace-project.eu/prototype-access>



IBM BlueGene/P (GCS-FZJ)
01-2008 / 06-2009



IBM Power6 (SARA)
07-2008



Cray XT5 (CSC)
11-2008



IBM Cell/Power (BSC)
12-2008



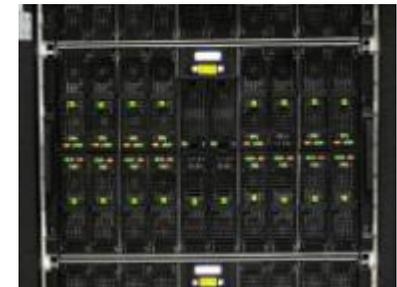
NEC SX9, vector part (HLRS)
02-2009



Intel Nehalem/Xeon (CEA/FZJ)
06-2009

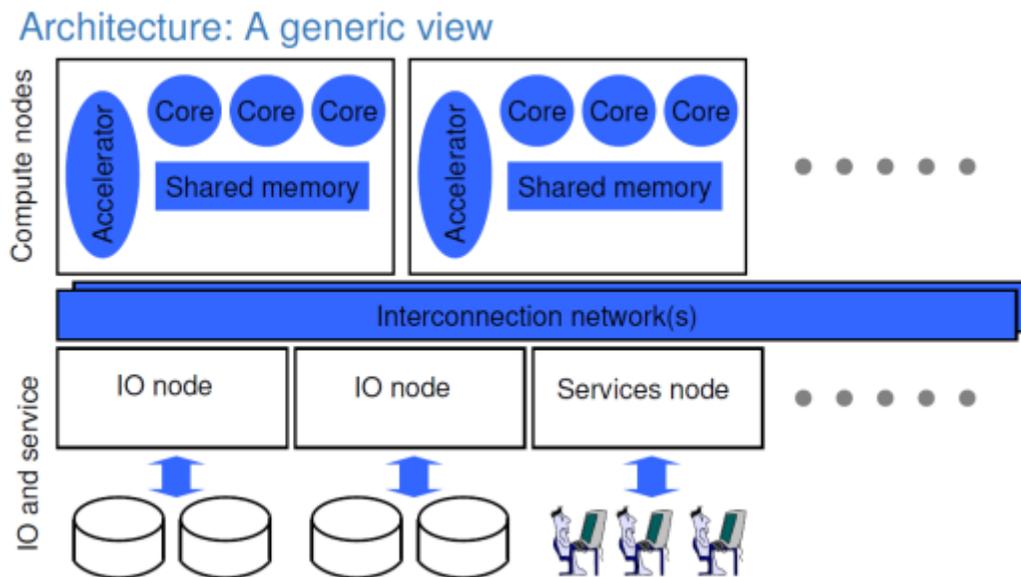
Au CEA: prototypes PRACE ouverts à la communauté scientifique dès fin 2008

- « **Petites machines** » dans une bulle « **expérimentale** »
 - CPU Nehalem (cluster Bull) ↔ Juropa @ FZJ/JSC
 - GPU Nvidia (serveurs TESLA)
- Installation préfigurant les futures connexions tier-0, « à la DEISA »
- Ont accueilli des activités expérimentales
 - internes à PRACE
 - et de quelques membres des communautés scientifiques européennes au sens large (futurs utilisateurs tier-0)



Le HPC est-il un long fleuve tranquille?

- **Court terme : petaflop/s (PF/s)**
 - **Equiper PRACE de machines dans l'état de l'art, 1+ PF/s**
 - **Architectures et composants existants ou imminents**
 - **Adapter les applications**
- **Cf. ce qui précède**



Crédit
F. Robin

Le HPC est-il un long fleuve tranquille?

- Court terme : petaflop/s (PF/s)
 - Equiper PRACE de machines dans l'état de l'art, 1+ PF/s
 - Architectures et composants existants ou imminents
 - Adapter les applications
- **Moyen terme : petascale**
 - **Systemes multi-petaflop/s (~5 à 10)**
 - **Consolider les applications, petaflop/s « soutenu »**
 - **Architectures et composants émergents**
- **Long terme: exascale???**
 - **Nécessaires ruptures : hw, sw**
 - **Efficacité énergétique, fiabilité, programmabilité à grande échelle**

➤ **Cf. ce qui suit et conclusions du jour ...**

Ce qui suit est principalement le fruit d'un sous-projet de PRACE PRACE/WP8

- Veille, R&D: 8% de l'effort projet 2008-2009
- ~20% de l'effort sur la phase IP 2010-2012
- Veille technologique, partage entre partenaires
- Evolution vers de plus en plus de prototypage 'actif' et collaboratif
- Structuration par grands thèmes transverses
 - **Efficacité énergétique**
 - **Programmabilité**
 - **Data management**
 - ...
- Création de STRATOS

QUELQUES EXPERIENCES TECHNOLOGIQUES : “PROTOTYPES”

A lot of various hardware and software information available and difficult to summarize. This is just a glimpse on what you will find in **public version of Deliverable D8.3.2** at:

<http://www.prace-project.eu/documents/d8-3-2.pdf>

A little bit more technical information available in the “long” version of this presentation

Hopefully this presentation makes you want to learn more about PRACE work around prototypes.

Programming Models



CSCS “UPC/CAF”

PGAS language compilers



LRZ “RapidMind”

RapidMind Multi-Core Development Platform



CEA “GPU/CAPS”

Tesla GPU Server (CUDA, HMPP, DDT)



CINES-LRZ “LRB/CS”

Hybrid SGI ICE/UV/Nehalem-EP&EX/ClearSpeed/(Larrabee)



NCF “ClearSpeed”

ClearSpeed & Petapath



FZJ “Cell & FPGA IC”

eQPACE (PowerXCell 8i)



EPCC “FPGA”

Maxwell FPGA

**Accelerators
Interconnect
Compute node
Architecture
(+Programming Models)**

**I/O, Storage
File Systems**



CINECA

I/O Subsystem (SSD, Lustre, pNFS)

**Energy Efficiency
H. density packaging**



SNIC-KTH + PSNC & STFC + CSC

Air cooled blade Supermicro with AMD Istanbul proc. & QDR IB

A large spectrum of issues covered by PRACE prototype set: holistic view

Std Proc. AMD (Istanbul), IBM POWER6->7), Intel (Nehalem)

	Vector	IBM-Cell	FPGA	GPU	Intel-Larrabee	ClearSpeed
Accelerators	WP7-HLRS	WP7-BSC WP8-FZJ	WP8-EPSRC-EPCC	WP8-GENCI-CEA	WP8-LRZ-1	WP8-NCF WP8-GENCI-CINES WP8-LRZ-1

Collaborate effort
among ClearSpeed
proposals

	PGAS	HMPP	RapidMind	DARPA	OpenMP/MPI	CUDA
Languages	WP8-ETHZ-CSCS	WP8-GENCI-CEA	WP8-LRZ-2	WP7-NCF (X10) WP8-ETHZ-CSCS (Chapel)	WP7-CSC/CSCS	WP8-EPSRC-EPCC WP8-GENCI-CEA

VHDL, Harvest

	I/O	Network	Power Efficiency	Large nodes	Very Large nodes
Hardware	WP8-CINECA ? WP8-LRZ-1	WP8-FZJ	(WP8-FZJ) (WP8-EPSRC-EPCC) WP8-SNIC-KTH-TN ?	WP8-SNIC-KTH-FN ? WP7-NCF	WP8-LRZ-1

	BULL	DELL/SuperMicro ?	IBM	WP7-FZJ SUN	NEC	HP	SGI	CRAY
HW Vendors	WP7-GENCI-CEA WP8-GENCI-CEA	WP8-SNIC-KTH ?	WP7-FZJ WP7-NCF WP7-BSC		WP7-HRLS	WP8-CINECA	WP8-GENCI-CINES WP8-LRZ-1	WP7-CSC/CSCS WP8-ETHZ-CSCS

PRACE Benchmark Suite

- For (pre) production systems: a set of user representative applications
- See: <http://www.prace-project.eu/news/prace-benchmark-suite-finalised>

Synthetic Benchmarks

- For technology prototypes (+ Specific applications)
- To allow comparisons (performance, effort to port): 4 representative synthetic benchmarks of the EuroBen (<http://www.euroben.nl>):
 - *mod2am* dense matrix multiplication $C=AxB$
 - *mod2as* sparse CSR matrix vector multiplication $c=Axb$
 - *mod2f* 1-D radix-4 Fast Fourier Transform (FFT)
 - *mod2h* random number generator

Programming Models



CSCS “UPC/CAF”

PGAS language compilers



LRZ “RapidMind”

RapidMind Multi-Core Development Platform



CEA “GPU/CAPS”

Tesla GPU Server (CUDA, HMPP, DDT)



CINES-LRZ “LRB/CS”

Hybrid SGI ICE/UV/Nehalem-EP&EX/ClearSpeed/(Larrabee)



NCF “ClearSpeed”

ClearSpeed & Petapath



FZJ “Cell & FPGA IC”

eQPACE (PowerXCell 8i)



EPCC “FPGA”

Maxwell FPGA

Accelerators Interconnect Compute node Architecture (+ Programming Models)

I/O, Storage File Systems



CINECA

I/O Subsystem (SSD, Lustre, pNFS)

Energy Efficiency H. Density Storage



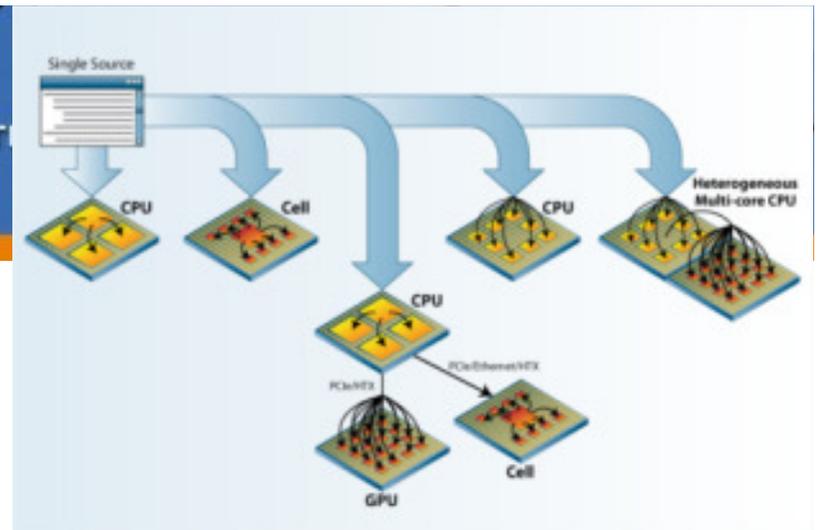
SNIC-KTH + PSNC & STFC + CSC

Air cooled blade Supermicro with AMD Istanbul proc. & QDR IB



PGAS Languages - ETHZ-CSCS

- **Partitioned Global Address Space (PGAS) a parallel program model (pm)**
 - Combine program advantages of shared mem pm + ctrl data layout & perf of message passing pm
 - incorporate data parallelism by smallest language changes providing mechanisms for data distribution with: global mem add space + logical partitioned mem + local mem/proc and **mem management for sharable entities - explicitly parallel execution SPMD**
- Assessments
 - **PGAS languages: Unified Parallel C (UPC, “thread”), Co-array Fortran (CAF “image”)**
 - **overlap computation and** (asynchronous 1-sided) **communication** in data-parallel applis
 - **productivity improvements** (also **HPCS-Chapel + task parallelism**)
- Configuration
 - Cray XT5 (CSCS) + Cray Compiler Environment (CCE)
 - SGI Altix ccNUMA shared mem. (LRZ) + UPC Berkeley U. and g95-compiler with CAF
- Some results
 - Scalability for **fine-grain parallelism in distributed mem will require special interconnect hw features**
 - **UPC more mature - CAF will be part of Fortran 2008 Std (targeted August 2010?)**



RapidMind - BAdW-LRZ

- **RapidMind, Automatic code generation for various backends:**
 - GPU (NVIDIA, ATI), IBM Cells and x86-Multicores (Intel, AMD)
 - adds special types and functions to C++, **express data parallelism** (not task) allowing
 - **data stream programming** (~ Fortran's array operations + freely programmable), **SPMD**
- Assessments
 - Parallelization **on a higher abstraction level than MPI or OpenMP.**
 - **Portable code for various hardware accelerators**
- Some results.
 - **easily describe data dependencies & workflows**, few lines needed, but
 - perf among platforms is very different and need different implementations
 - **RapidMind has lately been acquired by Intel and their product will dissolve in Intel's new language Ct ("C for throughput computing")**
 - Similar approach: **PGI Accelerator compiler** uses directives (C pragmas, Fortran comments) to offload compute-intensive code to NVIDIA CUDA GPU

Programming Models



CSCS “UPC/CAF”

PGAS language compilers



LRZ “RapidMind”

RapidMind Multi-Core Development Platform



CEA “GPU/CAPS”

Tesla GPU Server (CUDA, HMPP, DDT)



CINES-LRZ “LRB/CS”

Hybrid SGI ICE/UV/Nehalem-EP&EX/ClearSpeed/Larrabee



NCF “ClearSpeed”

ClearSpeed & Petapath



FZJ “Cell & FPGA IC”

eQPACE (PowerXCell 8i)



EPCC “FPGA”

Maxwell FPGA

*Accelerators
Interconnect
Compute node
Architecture
(+ Programming Models)*

*I/O, Storage
File Systems*



CINECA

I/O Subsystem (SSD, Lustre, pNFS)

*Energy Efficiency
H. Density Storage*



SNIC-KTH + PSNC & STFC + CSC

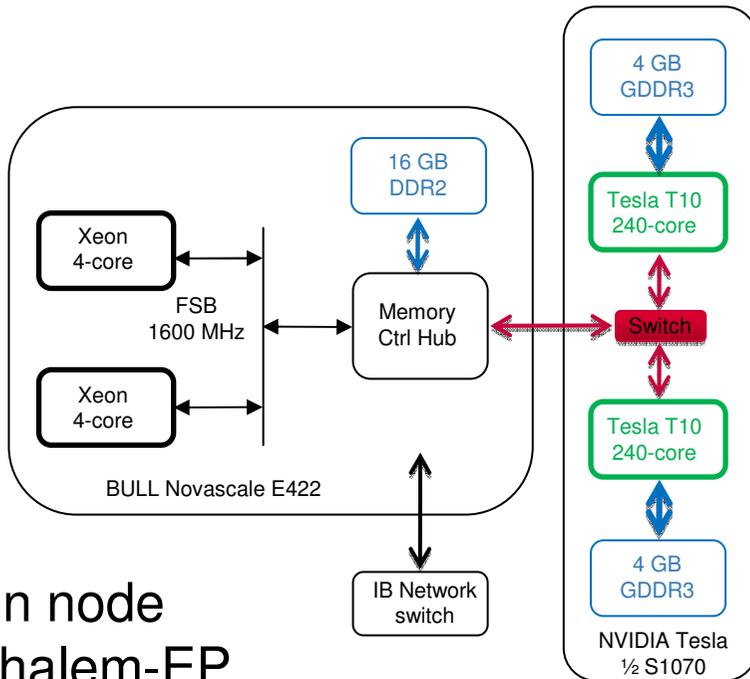
Air cooled blade Supermicro with AMD Istanbul proc. & QDR IB

Hybrid technology demonstrator - CEA

- Hardware

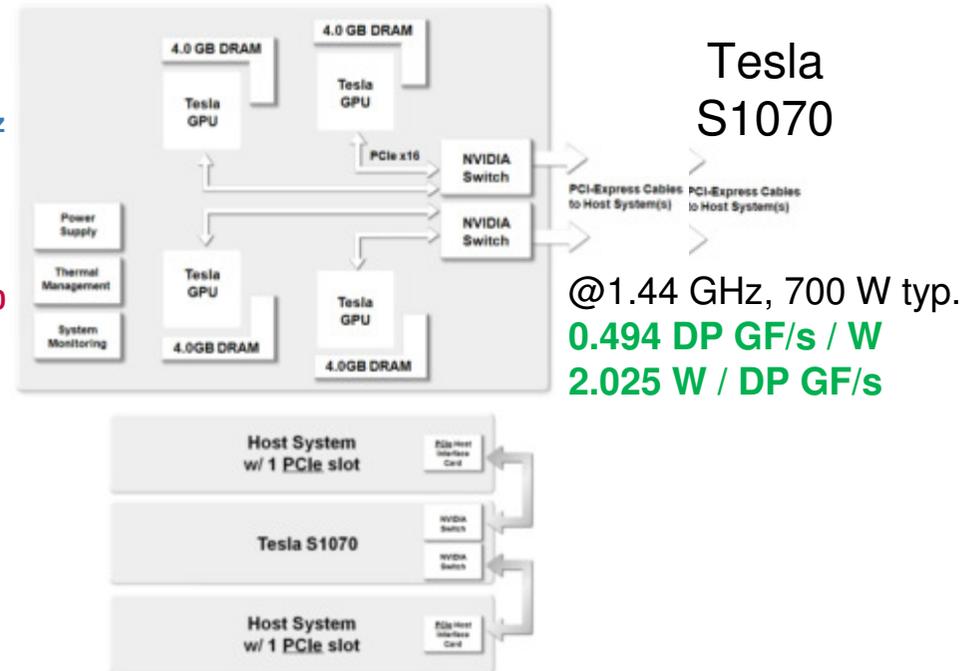
- 2 NVIDIA Tesla S1070-500 (4xT10 @1.44 GHz + 4x4 GB GDDR3 in 1U)
- 4 BULL NovaScale E422 (2xXeon E5462 4-core @ 2.8 GHz + 16 GB)
- 1/2 S1070 – E422 via PCIe x16 Gen 2.0 / E422 – E422 via IB DDR .

4 x

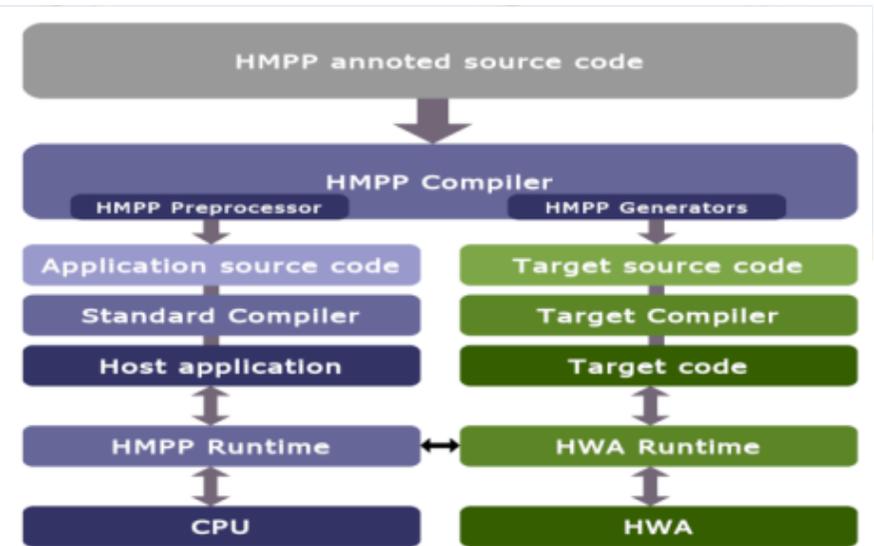
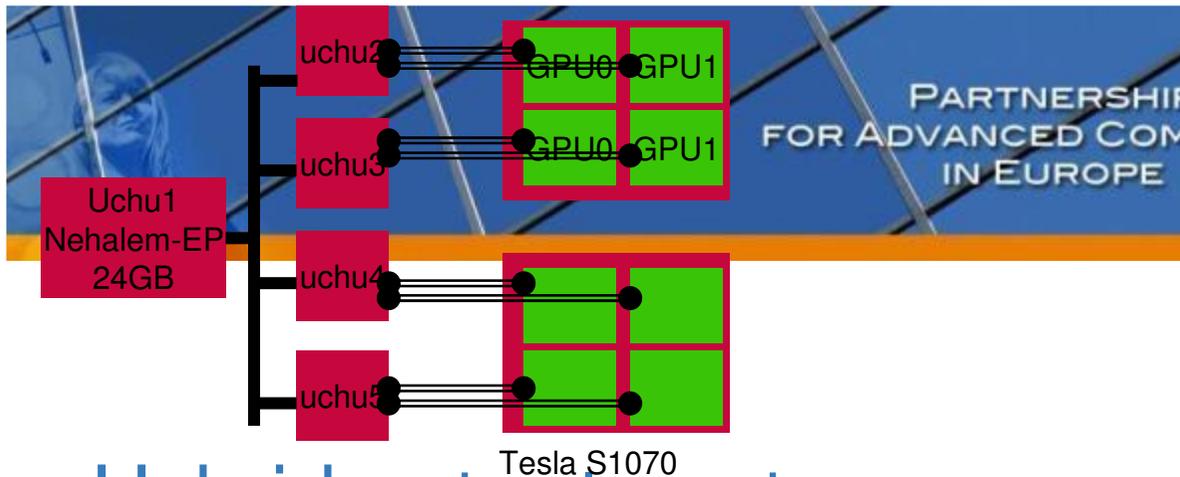


512-b, 800 MHz
102.4 GB/s

PCIe x16 Gen2.0
8 GB/s/dir FD



+ login node
4xNehalem-EP



Hybrid technology demonstrator

- Assessments: learn how to program efficiently those machines
 - Program environments: NVIDIA CUDA , CAPS-entreprise HMPP, RapidMind
 - Debugging hybrid machine using DDT from Allinea
- Hybrid Multicore Parallel Programming **HMPP**
 - **High-level abstraction for many-core program.** Annotated source code (comments for Fortran, pragma for C) **generate code for different targets, target oriented optimisations (eg NVIDIA CUDA, AMD CAL/IL, OpenCL)**
 - **Aims to Keep code portability:** C & Fortran (e.g., in legacy codes)
- Some results
 - Port to HMPP is easy yet can **reach significant performances**, difficulty lies in data mvt
 - Key high level features as **Asynchronous data transfers** (not yet in PGI) and kernel execution
 - mod2am vs. CUBLAS lib), mod2as and some real codes (**if kernels are important, only full applications can exercise a language**)

Hybrid System Architecture - LRZ-CINES

- Specific features
 - **Hybrid general purpose Multi-Petaflop system architecture with high performance/power efficiency**
 - SGI UV, SGI ICE, Nehalem EP/EX, ClearSpeed (+ NVIDIA Tesla) accelerator boards (originally also Intel Larrabee)
- Assessments
 - Test **SGI's next generation large scale ccNUMA (fat node UV) and thin nodes (ICE) systems**
 - Applicability of **hybrid architecture** approaches in a common programming environment within a **shared file system** with components from **SGI and ClearSpeed-Petapath**
 - Evaluate management software for hybrid systems
 - Scalability tests with different accelerator cards
- About **ClearSpeed processor** and boards...



ClearSpeed CSX700 processor and boards

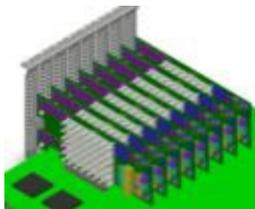


2x MTAP / chip
2x48x PE / MTAP
So 192 PE (core) / chip

96 DP GF/s IEEE 754-1985 @ 250 MHz
9 / 12 W typ / max
10.667 / 8 GF/W typ / min
0.094 / 0.125 W/GF typ / max
SECDED ECC on all internal Mem



Advance e710 board
Low profile PCI x8 slot
(vertically in 2U server)



Advance e720 board
HP blade Type II mezzanine slot

1x CSX700 + 1x FPGA ctrl data traffic
2 GB 533 MHz ECC DDR2
PCI e x8 Gen 1.1, 2 GB/s/dir (FD)
96 DP GF/s IEEE 754 @ 250 MHz
15 / 25 W typ / max
6.4 / 3.84 GF/W typ / min
0.156 / 0.26 W/GF typ / max

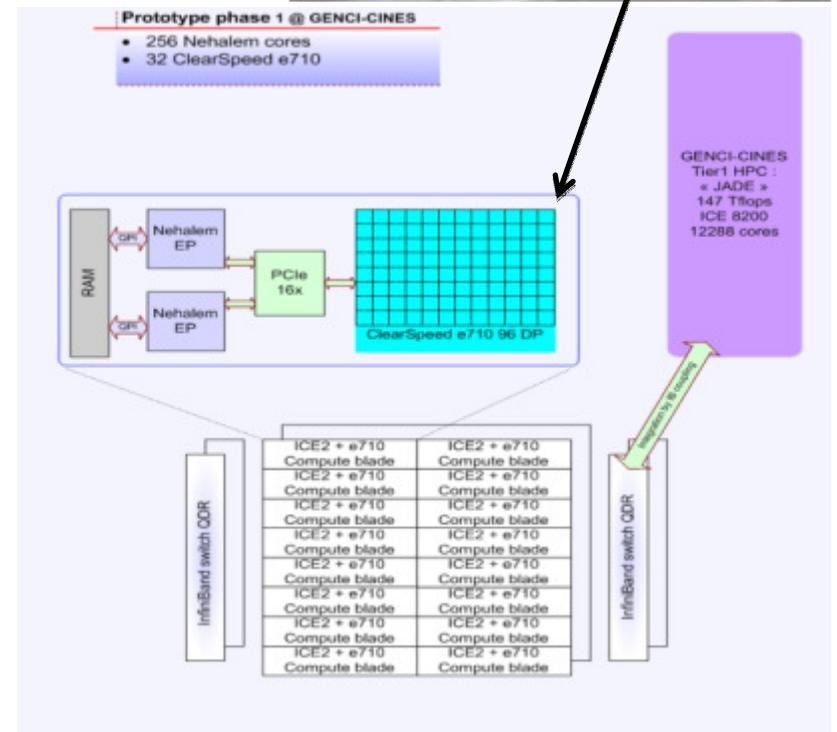


CTS 700
12x e710 in 1U, 19"
1.152 DP TF/s, 24 GB
400 W typ

Hybrid System Architecture LRZ-CINES

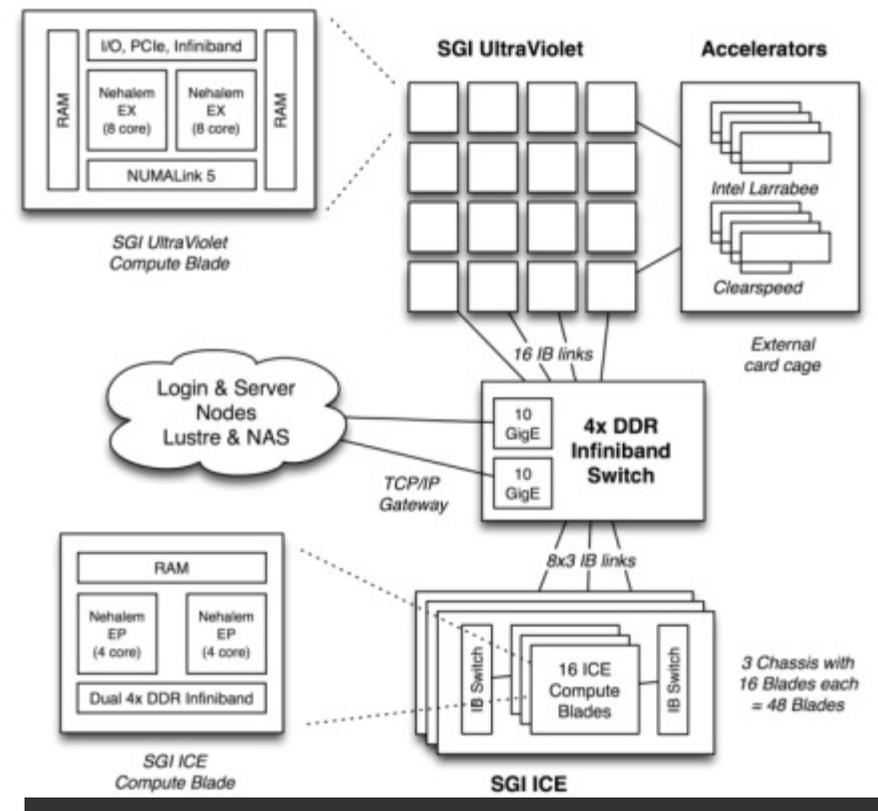


- CINES focuses on assessing **ClearSpeed as accelerator cards**
- Hardware
 - SGI-ICE hosting platform: 32 blades XE
 - Blade: 2 Intel Nehalem-EP (4-core), 4 GB/core
 - IB QDR
 - **Estimated peak perf 2.53 TF/s**
 - ClearSpeed-Petapath accelerators: 32 e710
 - 1 e710 per ICE blade connected by PCIe
 - **Estimated peak perf 3 TF/s**



Hybrid System Architecture LRZ-CINES

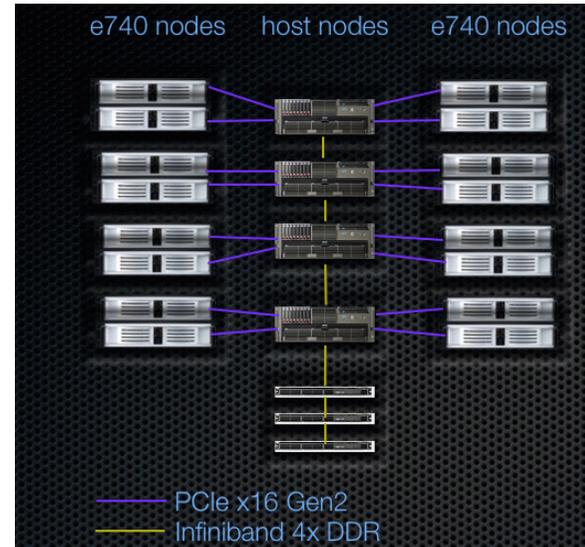
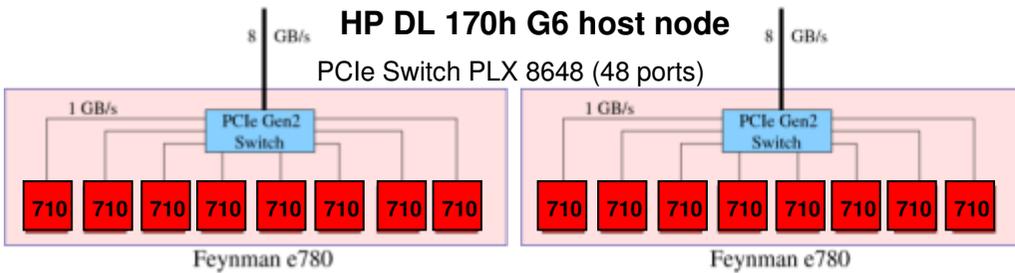
- LRZ focuses on the Evaluation of a **hybrid system architecture containing thin nodes, fat nodes and compute accelerators with a shared file system**
 - The prototype is a test vehicle for hybrid general purpose HPC architectures which will be available in the 2010 and 2011 timeframe: besides the ClearSpeed accelerators, there is **no breakpoint in the programming environment**
- Hardware
 - SGI Altix ICE 48 blades
 - Blade: 2 Nehalem-EP (4-core) @ 2.53 GHz, 3 GB/core
 - 4-socket Nehalem-EX white box
 - SGI UV proto 16 compute blades
 - 2 Nehalem-EX (8-core) @ 2.0 GHz, 2 GB/core
 - 4x DDR IB (44 IB ports + 2x 10 GigE)
 - SGI UV extension unit with 4 e710



ClearSpeed-Petapath - NCF

112 CSX700 in 2 racks
10.752 DP TF/s IEEE 754-1985

Nehalem-EP Intel X5550 @ 2.67 GHz	24 GB DDR3	Nehalem-EP Intel X5550 @ 2.67 GHz
PCIe x16 Gen 2.0	750 GB HD	PCIe x16 Gen 2.0



Rack 1	Rack 2
GigE Switch	
Feynman e740	Feynman e780
HP DL170	HP DL170
HP DL170	HP DL170
Head Node (HP DL160)	KVM
IB Switch	Monitor/Keyboard
HP DL170	HP DL170
HP DL170	HP DL170
Feynman e780	Feynman e780

Feynman e780: 8x e710 – PCIe x4 Gen 2.0, 2 GB/s/dir FD but **1 GB/s/dir FD shared bw**
Feynman e740: 4x e710 – PCIe x8 Gen 2.0, 4 GB/s/dir FD but **2 GB/s/dir FD shared bw**

Packaging: 2x HP DL 170h G6 in a 2U HP Proliant h1000 G6

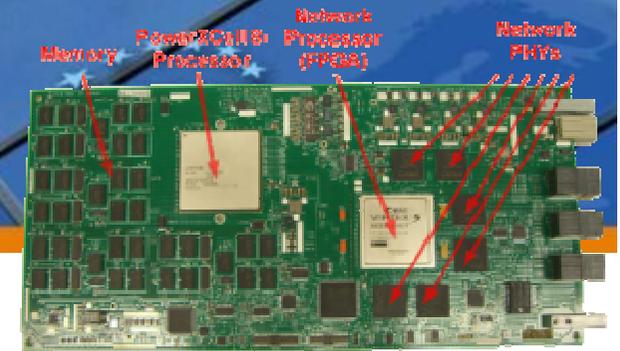


- **12 Feynman e780**: 96 CSX700, 9.216 DP TF/s for production
- **4 Feynman e740**: 16 CSX700, 1.536 DP TF/s for test & development
- 8 HP DL 170h G6 host nodes: 16 Intel Nehalem-EP, 2.67GHz
- 2 PCIe x16 Gen 2.0 per host node, one per Feynman
- 1 HP T2 head node, 2 TB file system + 2 additional development nodes
- DDR x4 IB, 2 GB/s/dir FD, between host nodes (Voltaire)



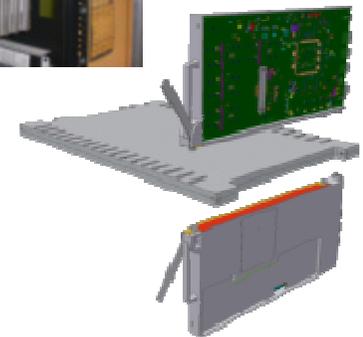
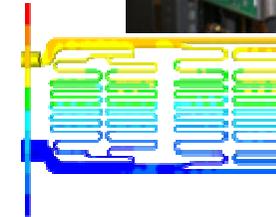
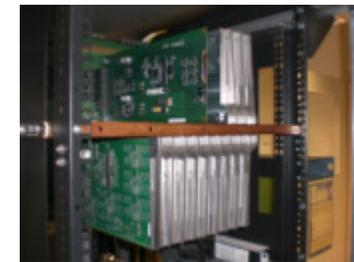
ClearSpeed-Petapath - NCF

- Specific feature
 - **Accelerators doing almost all computation**
- Assessments
 - **4 Euroben kernels + Porting large-scale real applis:**
 - Astronomy, Geophysics, numerical mathematics (very large sparse linear systems), medical tomography
- Some results
 - C^n : particular prog mod , important lib (lib routine in CS SDK for mod2am, mod2f, mod2h)
 - **mod2am, 1 MTA > Nehalem-EP for $N > 5000$, mod2as**, very bad - **mod2f**, mem access not favorable & limited mem/MTAP limits the size, perf quite good - **mod3h**, 1 MTAP ~ Nehalem-EP
 - Algorithm must map well (regular to benefit from the SIMD-type, eg DFT /FFT)
 - Issue:2-stage data transfer to parallel PE. **Must be done asynchronously to hide data transfer**
- Future of ClearSpeed & Petapath rather dim
 - **ClearSpeed in embedded computing, but in HPC? No sucesor of CSX700 chip?**
 - **High energy efficiency ~ 4 GF/s / W**



e(extended)QPACE – FZJ (Jülich)

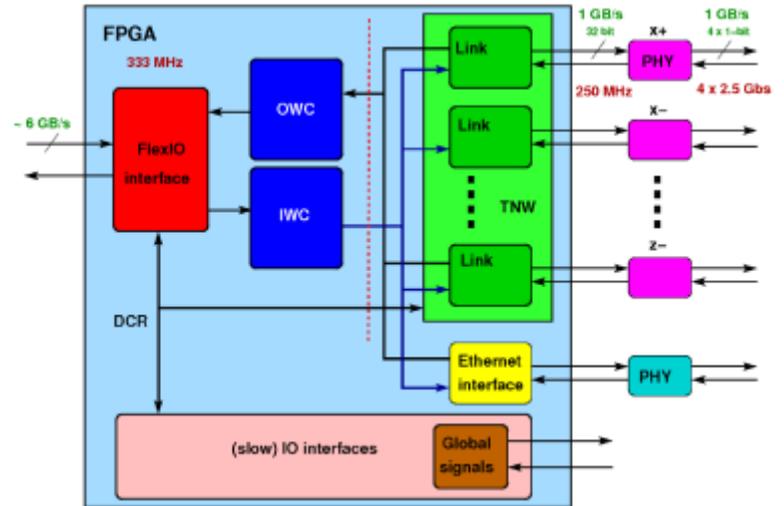
- QPACE: HPC-Project SFB/TR « Hadron Physics » Lattice QCD
 - led by **U. Regensburg**, cooperation with IBM-Böblingen, FZJ Jülich, DESY Zeuthen and U. Wuppertal, also U. Ferrara and Milano
 - **Two 4-rack QPACE systems installed (Jülich, Wuppertal)**
- Specific features
 - Innovative scalable architecture with **PowerXCell 8i**
 - **3D-torus based on FPGA Virtex-5 – nearest neighbor**
 - Direct SPE-to-SPE comm.: $\sim 1 \mu\text{s}$ – 1 GB/s/link/dir (FD)
 - **innovative (liquid) cooling concept**
- Objective: **make system suitable for applications beyond QCD**
 - Extensions to network communication + Extensions to software stack
 - **Top500 / Green500 publish for SC09 0.723 DP GF/s / W** **Mainly marketing: We need a new metric taking care of Energy & compute efficiency**
 - HPL communication requirements differ from QCD
 - **Transfert of large message, Collective operations**



eQPACE

- QCD optimisation
 - QCD kernels mem bw limited (reduce mem access & optimise re-used data in LS). Optimise & hide network lat $\sim 10^4$ cy. Efficient use of SP FP pipeline
- 1D Wave eq. As test application (Using torus network API via SPE-core)
- **Modif of FPGA network proc (Xilinx FPGA Virtex-5 LX110T)**
 - Reduce FPGA resource consumption & DMA ctrl (in place of IWC) for Main Mem-to-MM transfers via PPE
 - Torus Micro-Benchmark MM-to-MM, 4.7 μ s, 850 MB/s
- **SW extension**
 - **HPL for QPACE**: Support a performance critical subset of MPI. Open MPI BTL and COLL components. MPI uses nn com. Whenever possible, Ethernet otherwise
 - Issue: limited set of tools for debug FPGA errors

256 compute nodes, 26 TF/s DP / rack



Maxwell FPGA - EPSRC-EPCC



- Specific features
 - **FPGA as main compute nodes + FPGAs fast direct interconnect**
- Assessments
 - **Programmability of FPGAs for HPC using VHDL and the C-to-gate compiler HCE (Ylichron)**
 - Porting 4 EuroBen kernels
- Configuration: High Performance Computing Alliance (FHPCA), “Maxwell”
 - **32 accelerator cards Alpha Data Ltd + 32 Nallatech Ltd using Virtex-4 Xilinx FPGAs:**
 - **nearest neighbour 2D 8x8 mesh between FPGA (4 RocketIO , < 3.125 Gb/s p2p in 4 dir)**
 - 2 accelerator cards / node (IBM HS21-based Bladecentre) **connected over PCI-X-bus**



Maxwell FPGA - EPSRC-EPCC

- **Maxwell is now old technology (2007)**
 - Does not perform well compared to modern CPUs and GPGPUs
 - Except for certain specific codes (e.g. mod2h)
- **FPGAs still hard to program**
 - Hardware knowledge required
 - Code is very hardware specific and difficult to modify
 - Tools like HCE make it easier (but still harder than software). Harwest only support SP.
Mitrion C from MITRIONIC should be tested
- Performance of C-to-VHDL compilers now approaching that of hand-coded VHDL (for some codes)
- **Power consumption of FPGAs is excellent**
 - Large Virtex-4 @ 200 MHz ~ 2-3 W, EuroBen ports @ < 150 MHz ~ < 3 W
 - HCE mod2am ~ 3 GF/s, ~ **1 GF/s / W**

Future trends with new large FPGA:
- use of dedicated macro cells, e.g.,
proc., FP op.
- use of std interface, e.g., QPI, HT

Programming Models



CSCS “UPC/CAF”

PGAS language compilers



LRZ “RapidMind”

RapidMind Multi-Core Development Platform



CEA “GPU/CAPS”

Tesla GPU Server (CUDA, HMPP, DDT)



CINES-LRZ “LRB/CS”

Hybrid SGI ICE/UV/Nehalem-EP&EX/ClearSpeed/(Larrabee)



NCF “ClearSpeed”

ClearSpeed & Petapath



FZJ “Cell & FPGA IC”

eQPACE (PowerXCell 8i)



EPCC “FPGA”

Maxwell FPGA

*Accelerators
Interconnect
Compute node
Architecture
(+Programming Models)*

*I/O, Storage
File Systems*



CINECA

I/O Subsystem (SSD, Lustre, pNFS)

*Energy Efficiency
H. density packaging*



SNIC-KTH + PSNC & STFC + CSC

Air cooled blade Supermicro with AMD Istanbul proc. & QDR IB

XC4-IO – CINECA – I/O, storage, File Systems

- Specific features
 - Storage subsystems for up to 100 TB of on line storage
 - **SSD online storage for metadata**: DDN S2A9900 Singlet with 1.6 TB SSD Capacity
- Main contributions to the PRACE project
 - **Address metadata performance with SSD technology as a key issue in future Petascale systems**
 - Investigate and evaluate NFS and pNFS over RDMA (over IB) for a scalable HPC environment
 - **Address I/O performances and scalability for advanced HPC systems**
- Actually quite small size system - **assessment of these important topics should be pursued**

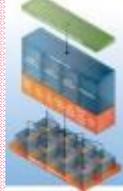


Programming Models



CSCS “UPC/CAF”

PGAS language compilers



LRZ “RapidMind”

RapidMind Multi-Core Development Platform



CEA “GPU/CAPS”

Tesla GPU Server (CUDA, HMPP, DDT)



CINES-LRZ “LRB/CS”

Hybrid SGI ICE/UV/Nehalem-EP&EX/ClearSpeed/(Larrabee)



NCF “ClearSpeed”

ClearSpeed & Petapath



FZJ “Cell & FPGA IC”

eQPACE (PowerXCell 8i)



EPCC “FPGA”

Maxwell FPGA

*Accelerators
Interconnect
Compute node
Architecture
(+Programming Models)*

*I/O, Storage
File Systems*



CINECA

I/O Subsystem (SSD, Lustre, pNFS)

*Energy Efficiency
H. density packaging*



SNIC-KTH + PSNC & STFC + CSC

Air cooled blade Supermicro with AMD Istanbul proc. & QDR IB

Energy Efficiency and high density with standard components SNIC-KTH

- Specific features
 - **Dense blade design** with 240 sockets and 1440 cores in a standard 42U rack
 - 4 sockets, 24 cores and 32 GB per node
 - QDR Infiniband with full bi-section width interconnect
- Assessment
 - **Density and energy efficiency** achievable through
 - careful design of servers **using commodity components**, and their integration into clusters
 - code optimization **without acceleration to preserve programming model**
 - utilization of **server and CPU power management features**
 - control of voltage and frequency settings
 - job and workload scheduling

SNIC-KTH Prototype – Node (Blade)

Proposed Platform: 4 socket Istanbul 2.4GHz, HT3, DDR2-800, PCIegen2, QDR IB

New Blade (Supermicro)

6-core CPU, 6 MB cache 55 W ADP

3 HT-3/CPU, up to 19.2 GB/s/link @ 4.8 GT/s

4 DIMM slot/CPU, 8 GB, 4/3 GB/core

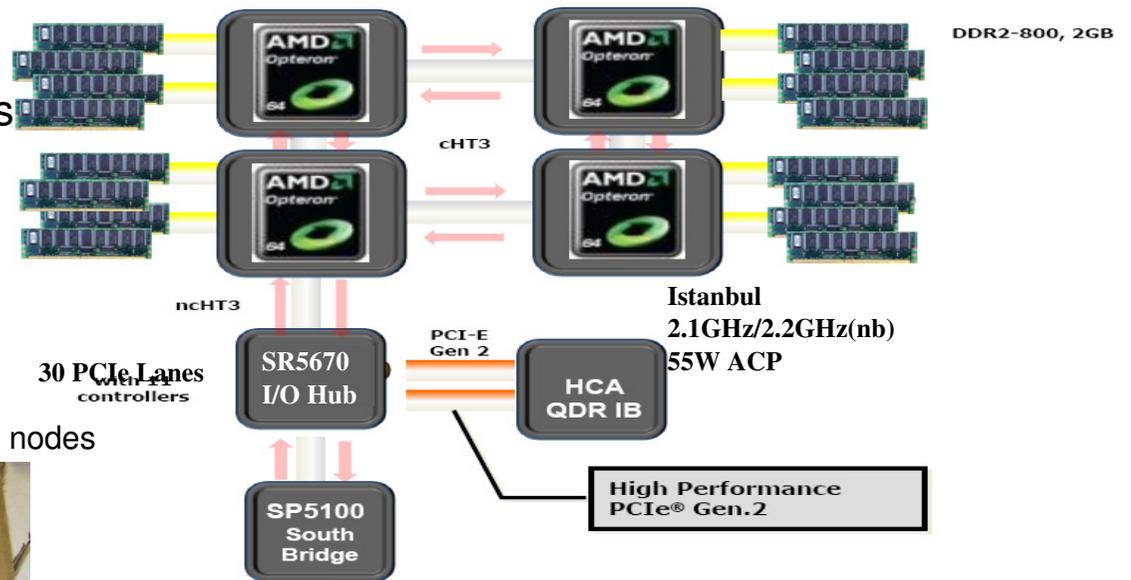
Diskless

New AMD chipset (SR5670 Fiorino)

PCIe Gen 2

QDR IB

AMD's APML



Not in prototype nodes



Various power mgt for AMD CPU

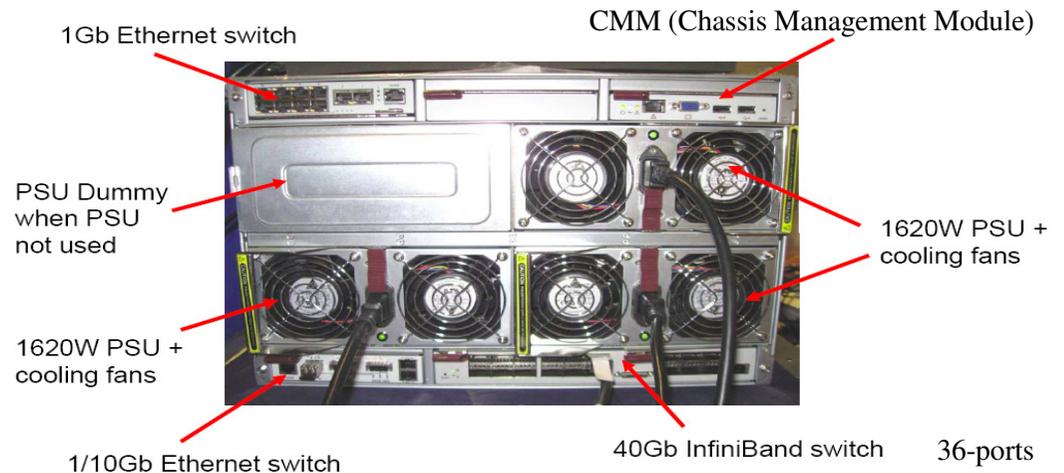
- **Dual Dynamic** Power mgt: core & mem ctrl separately
- **CoolCore techno**: turn off unused parts
- **PowerNow**: ctrl cor f and V
- **Smart Fetch**: mgt cores & caches vs power perspective



SNIC-KTH

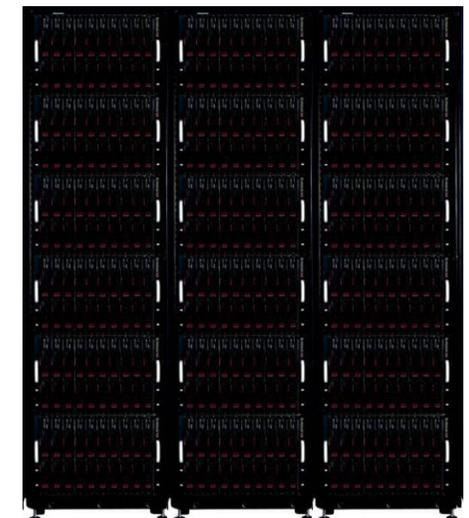
- **New Chassis: 5.1 kW air cooled**

- 10 4-socket blades in 7U (240-core), **5.7 sockets/U**
- New Power Supplies (3x 1620 W)
- New integrated 36-port QDR IB switch 18 ext. ports
- 1/10 GigE switch
- Support for AMD APML



- **System configuration: air cooled**

- 6x Chassis/Rack 46U 19" std 0.6mx1.07m (60 blades, 1440 cores)
- For 1 Rack: 12.1 DP TF/s, 30.6 kW, **395 MF/s /W**, **18.84 TF/m²**
- **3x Racks (180 blades, 4320 cores)**
- **Network IB QDR 2-level Fat-Tree**, Leaf level 36-port switches built into chassis + External switches (5) also 36-port switches



SNIC-KTH Prototype Chassis Design estimates

Component	Power (W)	Percent (%)
CPUs (HE, 55W ADP)	2,880	56.8
Memory	800	15.8
PS	355	7.0
Fans	350	6.9
Motherboards	300	5.9
HT3 Links	120	2.4
IB HCAs	100	2.0
IB Switch	100	2.0
GigE Switch	40	0.8
CMM	20	0.4
Total	5,065	100.0

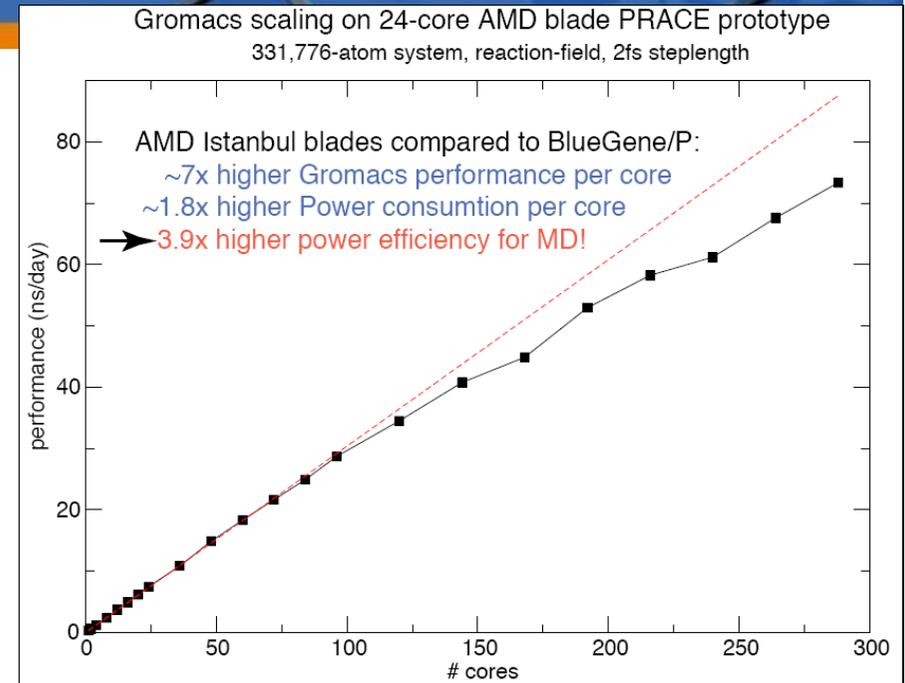


HPL observed: Max 4,647 Avg 4,625 W

Stream observed: 3,620 W

SNIC-KTH – Studies

- Memory study
 - With current chipsets, V & F for DIMMs cannot be ctrl dynamically (in future)
 - After measures of power consumption, Hynix selected / Elpida, Micron, Samsung
- Benchmarks, work in progress
 - **HPL 343.91 MF/s / W** with more mem BG/P Green500 Nov 2008 357.14-371.67 MF/s / W
 - Stream: Add 41.57 GB/s, Copy 41.86 GB/s, Scale 42.32 GB/s, Triad 41.6 GB/s, ~ **42 GB/s**
 - GUPS, GROMACS
 - To be done: EuroBen, MPI, HPL on entire cluster, RINF, Optimization, ctrl of power settings, ...

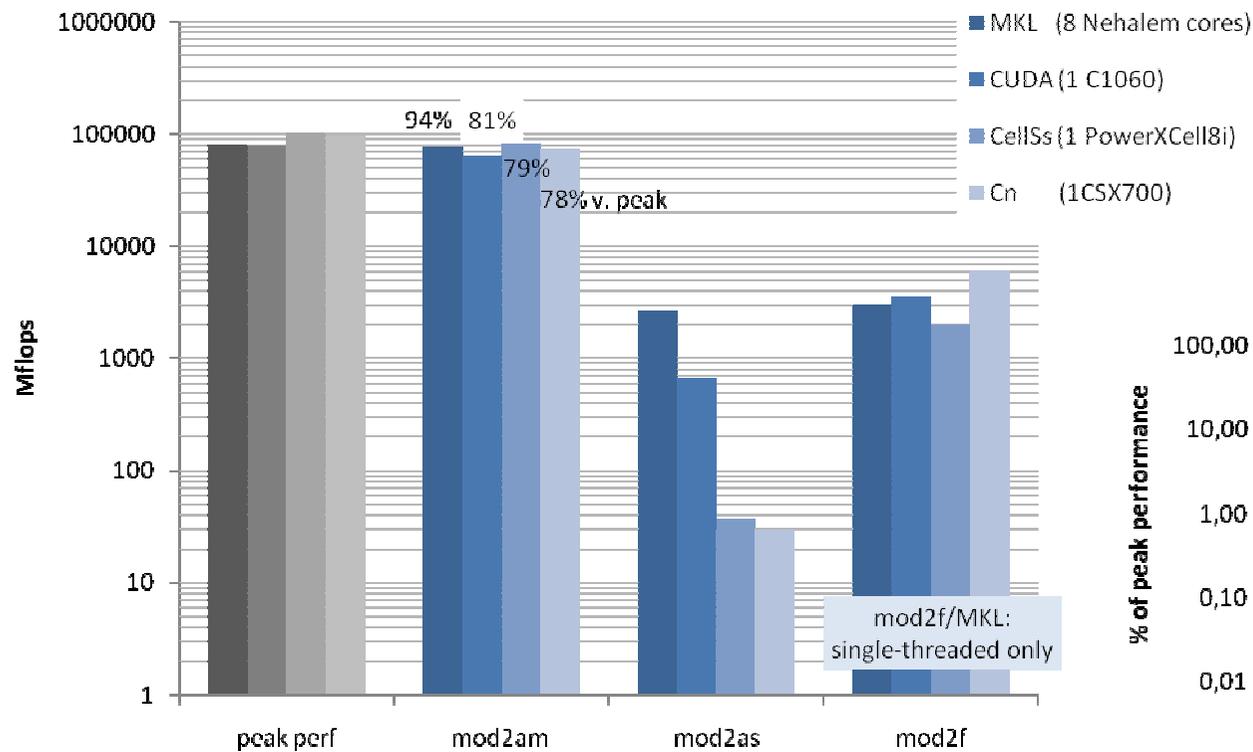


SOME EUROBEN RESULTS (kernels)

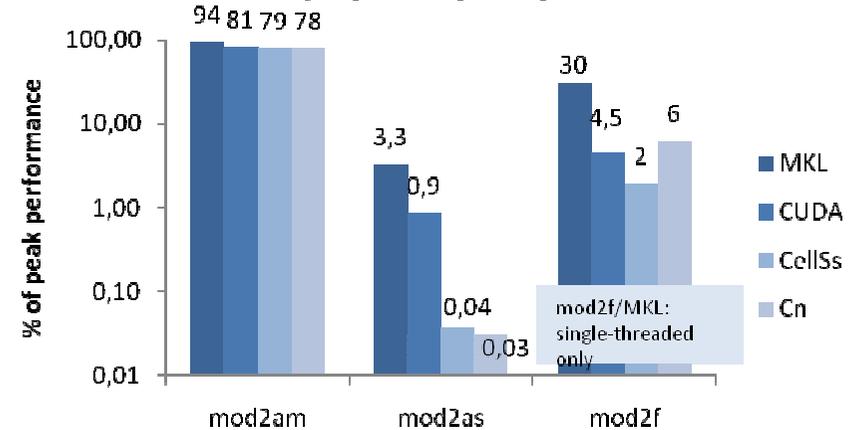
Detailed Results are reported in a public version of Deliverable D8.3.2 available at <http://www.prace-project.eu/documents/d8-3-2.pdf>

Euroben results - accelerator languages

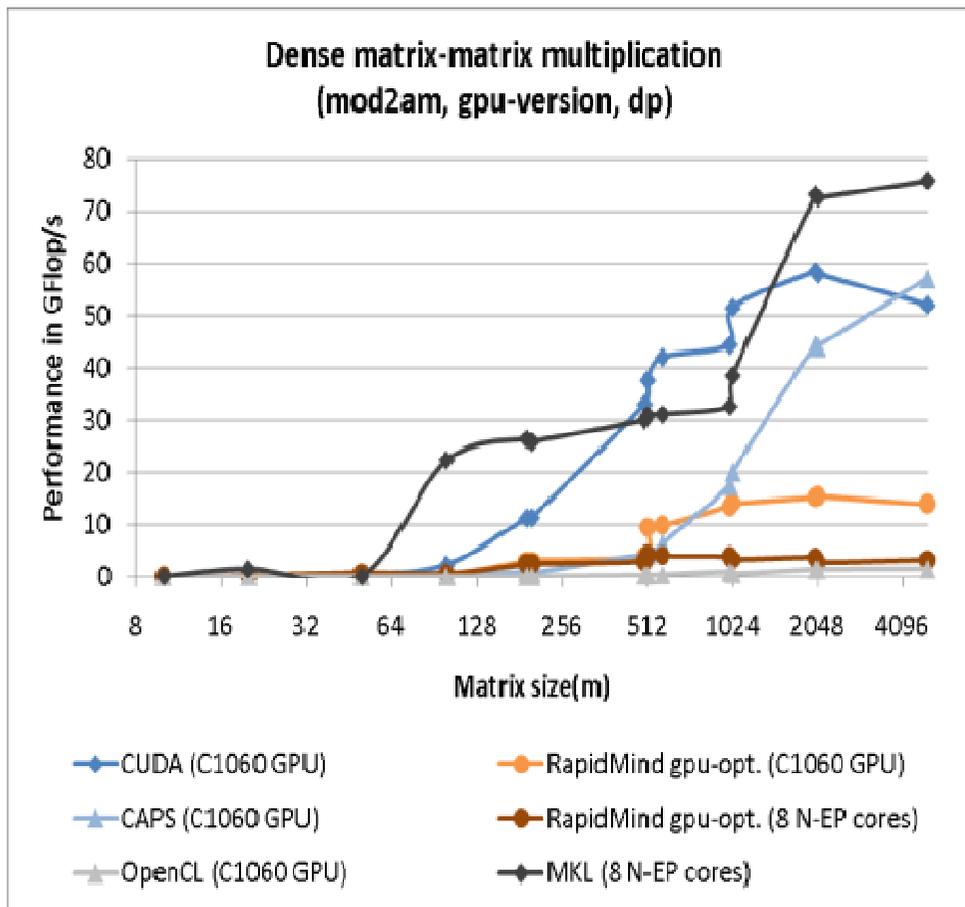
Accelerator Languages (absolute performance)



Accelerator Languages (%peak perf)



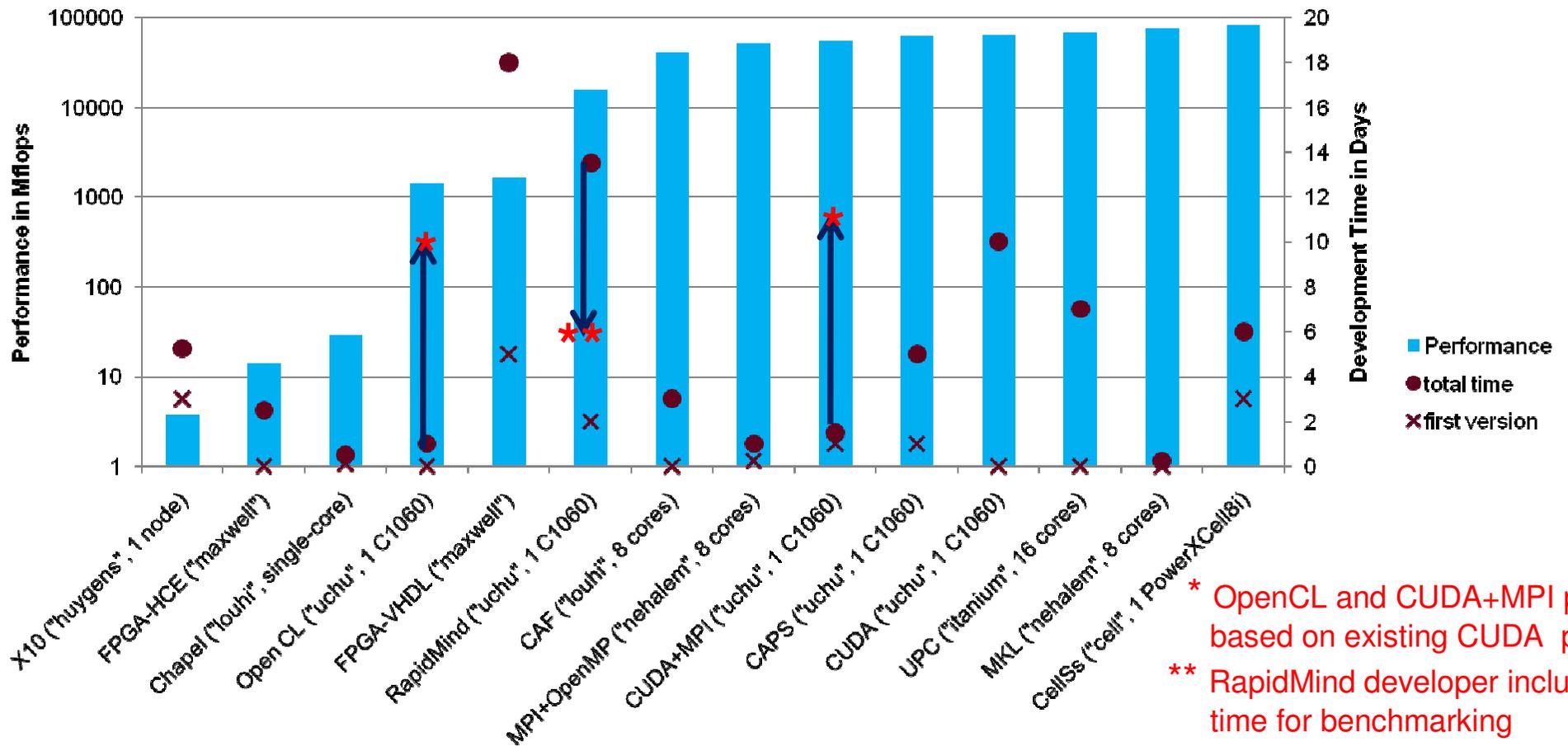
Euroben results - GPGPU languages



Hardware	SP Peak Performance	DP Peak Performance
Nehalem-EP (2.53 GHz, 1 core)	20.24 GFlop/s	10.12 GFlop/s
Nehalem-EP (2.53 GHz, 8 cores)	161.92 GFlop/s	80.96 GFlop/s
1 C1060 GPU	933 GFlop/s	78 GFlop/s
1 PowerXCell8i (8 SPU)	204.8 GFlop/s	102.4 GFlop/s
2 PowerXCell8i (16 SPU)	409.6 GFlop/s	204.8 GFlop/s
e710, 1 CSX700		96 Gflop/s

Euroben results - productivity

Development Time versus Performance (dense matrix-matrix mul.)



About Accelerators

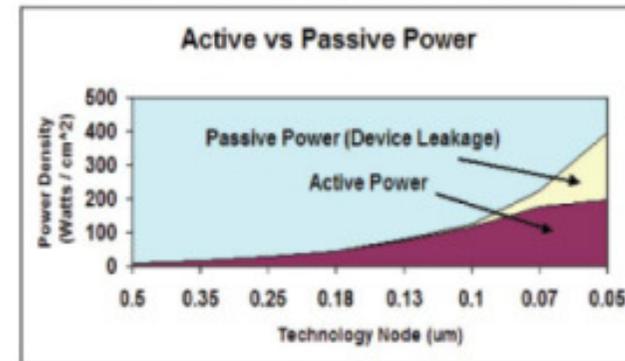
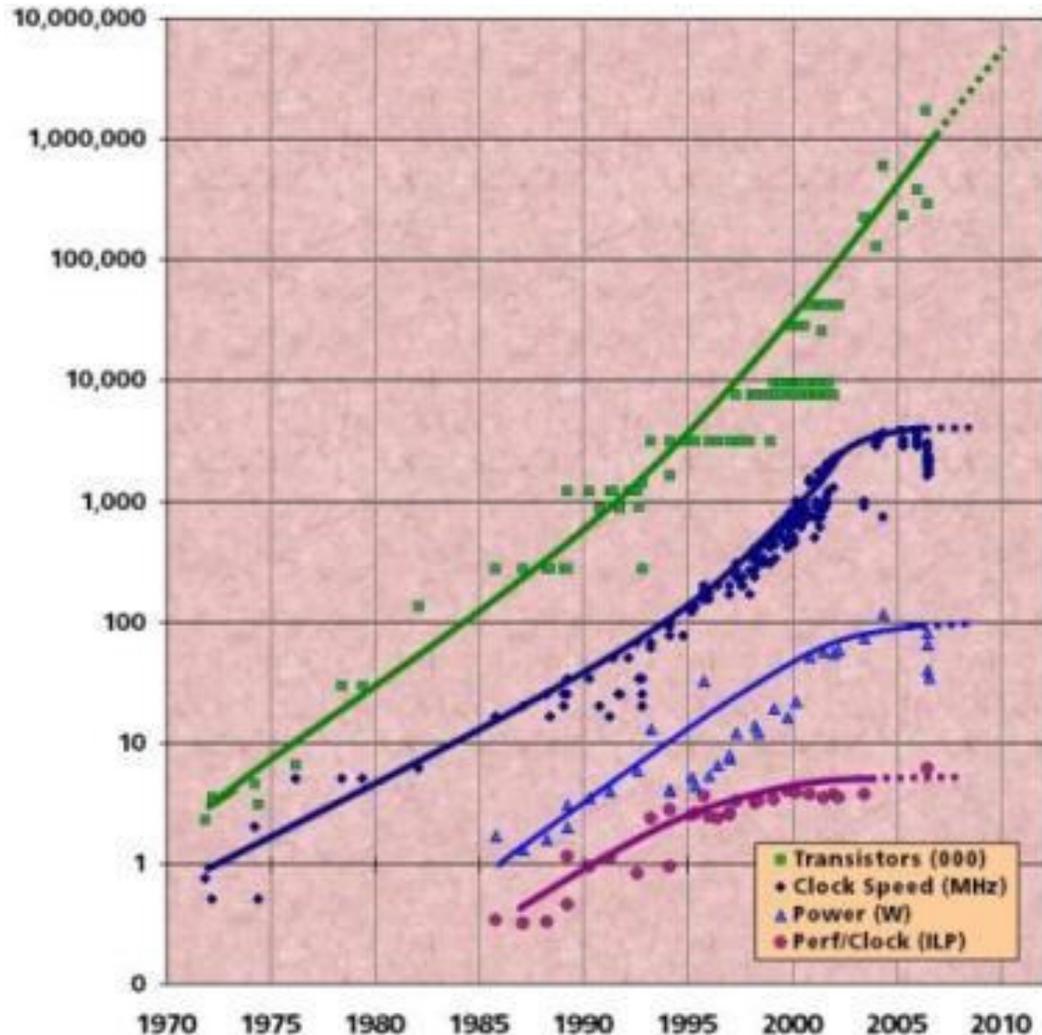
Results show that some hardware accelerators have indeed the potential to substantially increase performance and/or power efficiency of traditional HPC systems. But software environments for hardware accelerators are not tailored to the demands of the scientific computing community. They need to become more stable, easier to use and better supported by debugging and optimisation tools.

A blue banner with a grid pattern. On the left, a woman's face is partially visible. In the center, the text "PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE" is written in white. On the right, the PRACE logo features the word "PRACE" in blue, surrounded by a circle of white stars.

PARTNERSHIP
FOR ADVANCED COMPUTING
IN EUROPE

SOME FINAL REMARKS

Multicore / hybrid multicore



-until mid 1980:

- CISC, vector, RISC

-mid 80-mid 05

- Moore's law + parallelism
MPI etc.

- Faster clock , **increase power consumption an dissipation.**

-After 2005 clock stagnation (2-5 GHz)

- **Multicore instead**

The challenges of multicore

- **Functional decomposition (s/w)**
 - Useful for all applications, HPC or not (multimedia, even word processing)
 - Parallelism difficult for EVERYBODY, not « natural »
 - Ubiquity of parallelism may help but huge effort of training required anyway
 - HPC needs it at a large scale with suited programming models and related tools
 - Algorithms themselves may need re-design
- **H/W challenge: memory wall**
 - **bw & size of mem/core lag behind CPU processing speed**
 - Instruction and data caching, complex memory hierarchy (on and off-chip)
 - Multithreading (allow to **hide any kind of latency**)
 - Complex branch prediction mechanism (compiler optim)
 - **Interconnect latency and bw is a concern too**

Relevant issues (1)

Power and energy efficiency

- **Processor**
 - Monitor and control components
 - APIs? Auto, run-time?
- **Memory**
 - Up to 30% power
 - Hope in new non-volatile mem: STT MRAM (2011?), memristor, graphene-based
- **I/O**
 - Already SSD non-volatile flash NAND or NOR techno
 - Then, Phase change RAM (2010?), impact on budget 2014
- **Interconnect Network (IN)**
 - IN, switches & HBA consume power
 - Within 5 years, photonic network, NoC in Si, chip-system without electronic/optic conversion
- **Develop power ctrl capabilities at job scheduler level**

Relevant issues (2)

Programming models and compilers

- **Main issue: low % of Peak perf use by appli**
- **Re-writing codes for multi-PF or beyond desirable or mandatory: which tools?**
 - **PGAS, DARPA/HPCS languages (CAF, UPC, X10, Chapel) ? Still immature**
 - **Accelerators: no complete solution yet for GPUs, FPAGs still worse**
 - **Compilers + TOOLS**
- **Legacy code does exist!**
 - **tools keeping portability: PGI compiler, HMPP**
- **New initiatives: IESP (USA et al.), EESI (Europe),STRATOS**
 - **http://www.exascale.org/iesp/Main_Page**

Relevant issues (3) « Accelerators »

- Very much dependent on algorithm vs. Device matching
- Programmability not easy for HPC and non-trivial codes
- Mainstream CPU tend to catch up quickly with accelerators capacities
 - Speedup must be significant to amortise the porting effort
 - Importance of source code durability
- Nevertheless
 - New NVIDIA (Fermi) and AMD with HPC characteristics: more DP FP, IEEE754-2008, ECC
 - Developed program techniques will be useful for future many-core
 - **Main bottleneck to eliminate: BW limits between host proc and mem, and accelerator**
 - **Need to include power saving mechanisms in accelerators**
 - Future system will be heterogeneous, towards HW and/or SW integrated (technology cycle)

Relevant issues (4) Network (nw) interconnects

- **IB dominant in HPC in upcoming years**
 - QDR, 8 Gb/s/link/dir FD, possibly aggregated in 4x and 12x
- **For cost (~ 20% total) topology will be (3 to 6)–torus, n-dim hypercube, fat tree, other hierarchical nw**
- **Impact of nw bw and latency on perf often not clearly understood**
- **Need of ever growing bw/core or node, but bw/core will decrease. Need to develop algorithms to take care of that**
- **Key importance in future:**
 - Topology aware batch scheduling
 - Intelligent nw routing mechanisms (cf. also resilience)

Relevant issues (5)

Memory bandwidth (bw) and latency: always a main issue

- **Presently latency ~100-1000 cycles**
 - bw increases from DDR2 to DDR3 but latency also
- **Latency: no hope until radically new mem: memristor, graphen-based 2015/2016?**
 - Take care of locality, mem hierarchy
 - Multithreading to hide (all kind of) latency
- **BW: increase with 3D mem stacking (Intel, IBM, TSMC, Samsung...)**
 - also solves shrinking perimeter problem
- **Key research on memory hierachy (size and handling by intelligent runtimes)**
- **Memory per node and core: which relevant ratios?**
 - Too high = expensive and power consuming, not always used
 - Too low = limits codes
 - Decouple computational and mem requirements: virtualization of mem (ScaleMP, 3Leaf)?

Relevant issues (6) Others (not necessarily minor ones!)

- **Resilience at scale**
- **Performance modelling and related tools**
- **Runtime Systems**
- **Libraries**
- ...

Cf Bonus material below...

Foreseeable architectures (3)

- **Until around 2014: ~ 10-30 PF/s (multi-petaflop)**
Either fast proc in 3-4 GHz, eg
 - **IBM Blue Waters (NCSA) ~ in 2011**
<http://www.ncsa.illinois.edu/BlueWaters/system.html>
 - POWER7 8-core @ 4 GHz, 32 GF/s / core, 1 GB/core
 - MCM Quad Chip Module (QCM), 8 QCM/drawer=256 cores,
 - ~ 300 000 cores, 10 PF/s, 20 MW, **0.5 GF/S / W**
 - **Cray Baker et Cascade, ‘post-XT’ (DARPA HPCS)**
 - **RIKEN Keisoku, cluster ~10 PF Fujitsu, SPARC VIIfx ‘venus’**
 - **« GPU » machines : China, Russia, USA, Japan, France(?)**

Foreseeable architectures (4)

- **Until around 2014: ~ 10-30 PF/s (multi-petaflop)**

Or « slow » low power proc

- IBM Blue Gene/Q Sequoia (DOE LLNL) ~ in 2012
 - PowerPC 16 cores/chip, 10.2 GF/s / core, 1 GB/core
 - 1 chip/node, 98304~10⁵ nodes=1.6 x10⁶ cores , 20 PF/s, 6 MW, **3.33 GF/s / W**

Or even slower proc

Toward convergence between HPC and embedded computing?

- 2x10⁷ 8-core Tensilica Xtensa proc, 10 GF/s / proc @ x MHz, 200 mW, **50 GF/s / W !**
- <http://www.lbl.gov/cs/html/greenflash.html>

Foreseeable architectures (5)

- **2015-2018: x 100 PF/s => 1000 PF/s ?**
 - **Cannot just x # cores/nodes at least for power requirements**
 - **speed/core: 3D stacking? Nanotechnos (after usual lithography)?**
 - **change mem techno in speed, bandwidth, power consumption, eg, magnetic RAM, graphene-based (2015-2016?)**
 - **improve interconnect bw and latency: optical**
 - **Might be for 100 PF/s:**
 - **16 GF/S / core, 64 cores/proc, 1 TF/s / socket (or less core/proc+accelerators)**
 - **16 sockets/node**
 - **6400 nodes for 100 PF/s?? Power consumption??**

Foreseeable architectures (6)

- **2019-2020: expected/hoped EF/s (ExaFlop/s)**
 - **Need techno breakthroughs, (need as Bipolar to CMOS jump), e.g. (?),**
 - **supraconducting devices (SQUID)**
 - **pro: no leakage, low energy, THz / con: low T (liquid He)**
 - **Graphene- and carbon nanotube-based techno**
 - **Too far to predict what will be the HW implementation**
 - **Might be: 10 TF/s / socket, 16 sockets/node, 6400 nodes ??**
 - **http://www.darpa.mil/ipto/personnel/docs/ExaScale_Study_Initial.pdf**
- **A lot of HW and SW work to be done until then**
 - **Programming model + training**
 - **Power consumption**
 - **... even to efficiently use PF systems meanwhile!**



PARTNERSHIP
FOR ADVANCED COMPUTING
IN EUROPE



Many thanks to all PRACE members who contribute to all these studies and especially to the authors and contributors of D8.3.2:

***Ramnath Sai Sagar (BSC), Jesus Labarta (BSC), Aad van der Steen (NCF), Iris Christadler (LRZ), Herbert Huber (LRZ)
Eric Boyer (CINES), James Perry (EPCC), Paul Graham (EPCC), Mark Parsons (EPCC), Alan D Simpson (EPCC), Willi Homberg (FZJ), Wolfgang Gürich (FZJ), Thomas Lippert (FZJ), Radoslaw Januszewski (PSNC), Jonathan Follows (STFC), Igor Kozin (STFC), Dave Cable (STFC), Hans Hacker (LRZ), Volker Weinberg (LRZ), Johann Dobler (LRZ), Christoph Biardzki (LRZ), Reinhold Bader (LRZ), Momme Allalen (LRZ), Jose Gracia (LRZ), Vladimir Marjanovic (BSC), Guillaume Colin De Verdière (CEA), Calvin Christophe (CEA), Hervé Lozach (CEA), Jean-Marie Normand (CEA), Sadaf Alam (CSCS), Tim Stitt (CSCS), Neil Stringfellow (CSCS), Giovanni Erbacher (CINECA), Giovanni Foiani (CINECA), Carlo Cavazzoni (CINECA), Filippo Spiga (CINECA), Kimmo Koski (CSC), Jussi Heikonen (CSC), Olli-Pekka Lehto (CSC), Lennart Johnsson (KTH)***

Contact information PRACE / WP8 (future PRACE-1IP / WP9):

Dr. Herbert Huber (WP8 Leader), huber@lrz.de

Iris Christadler (WP8 Co-Leader), christadler@lrz.de

Leibniz Supercomputing Centre (LRZ), Garching, Germany

Some PRACE references:

- PRACE Public Website <http://www.prace-project.eu>
- PRACE Public Deliverables <http://www.prace-project.eu/documents/public-deliverables-1>
and for WP8 prototypes <http://www.prace-project.eu/documents/d8-3-2.pdf>
- PRACE Workshop “New Languages & Future Technology Prototypes” (March 1-2, 2010)
<http://www.prace-project.eu/documents>
- PRACE Benchmark Suite <http://www.prace-project.eu/news/prace-benchmark-suite-finalised>
- PRACE Prototype Access <http://www.prace-project.eu/prototype-access>

THANK YOU FOR YOUR ATTENTION!
COMMENTS? QUESTIONS?

jean-philippe.nomine@cea.fr jean-marie.normand@cea.fr

A blue banner with a grid pattern. On the left, there is a faint image of a woman's face. In the center, the text "PARTNERSHIP FOR ADVANCED COMPUTING IN EUROPE" is written in white. On the right, there is a logo consisting of a circle of white stars with the word "PRACE" in blue in the center. The banner has an orange border at the bottom.

PARTNERSHIP
FOR ADVANCED COMPUTING
IN EUROPE

BONUS MATERIAL

Some Final Remarks: relevant issues

- Performance tools
 - Need **perf predictions & perf modelling** to cope with system and appli size increase
 - Eased by **decomposition of appli into kernels** with known perf on different systems
 - **Allow to answer « what if? »**, e.g., change mem size, bw, latency, ...
- Load balance
 - Often major bottleneck for strong scaling as # proc ~ granularity & heterogeneity of appli
 - Global load imbalance (lib to recompute partitions), Microscopic load imbalance in synchro phases will be more important: **need asynchrony in program. models** (lacking for MPI+OpenMP)
- Runtime Systems
 - SW infrastructure where actual and precise info about system resource availability and perf
 - So should be used to make better informed decisions specific for the appli
 - Going from static to **dynamic intelligent runtime** for max perf and min power consumption
 - Including intelligent management of mem hierarchy

Some Final Remarks: relevant issues

- **Resilience**

- Will be a **huge problem in multi-PF/s and EF/s?**
 - Checkpointing mechanisms need to be improve to scale to M core
 - Repair approach only sufficient until 2014-15?
- **After 2016 fault avoidance needed** (MTF ~ time to write appli checkpoint)
 - Run time environment + HW able to go on even if components fail (both proc & mem chip-kill facility)
- **End-to-end data integrity** mechanisms urgently needed avoiding data lost/corruption

- **Arithmetic**

- **IEEE754-2008 std**, especially DP (64-bit)
- Might be necessary to consider 128-bit for more precision

Some Final Remarks: relevant issues

- Benchmarks (BM)
 - Need of comput., communic., I/O kernels to model and predict perf core/node/system
 - Need of user representative appli BM (PRACE BMS)
 - Always revisited (new algorithm, new user needs, ...)
- Libraries
 - Present: language binding only for C, sometime C++ and Fortran
 - Accelerators have or not lib, **specific lib should be into new std lib interface**
 - Present lib optimized for speed. **Energy efficiency might be another criterion**
- Applications
 - Important classes of problems, characterised by unstructured parallelism, are under-represented
 - E.g., large graph analysis pb, routing pb, automated machine learning algo, pattern recognition in large unstructured data bases
 - Need **massively multithreaded systems with (virtual) shared mem**, like discontinued Cray XMT (actually MTA, Burton Smith), massive core alternative