SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

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

PRACE-1IP

PRACE First Implementation Project

Grant Agreement Number: RI-261557

D9.3.4
Final Report on Prototype Evaluation

Final

Version: 1.0
Author(s): Lennart Johnsson, Gilbert Netzer, SNIC/KTH
Date: 19.12.2013

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: RI-261557**	
	Project Title: PRACE First Implementation Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: D9.3.4	
	Deliverable Nature: Report	
	Deliverable Level: PU	**Contractual Date of Delivery:** 31 / December / 2013
		Actual Date of Delivery: 30 / December / 2013
	EC Project Officer: Leonardo Flores Añover	

* - The dissemination level is 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: Final Report on Prototype Evaluation**	
	ID: D9.3.4	
	Version: 1.0	**Status:** Final
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2007	
	File(s): D9.3.4.doc	
Authorship	**Written by:**	Lennart Johnsson, Gilbert Netzer, SNIC/KTH
	Contributors:	Eric Boyer, CINES Paul Carpenter, BSC Radosław Januszewski, PSNC Giannis Koutsou, CaSToRC Ole Widar Saastad, SIGMA/UiO Giannos Stylianou, CaSToRC Torsten Wilde, LRZ
	Reviewed by:	Alan Simpson, EPCC Florian Berberich, JUELICH
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	13/September/2013	Draft	Skeleton
0.2	25/November/2013	Draft	
0.3	06/December/2013	Draft	For review
1.0	19/December/2013	Final version	

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure; Prototype Evaluation

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[D933] L. Johnsson, G. Netzer. D9.3.3 Report on Prototypes Evaluation. PRACE-1IP, March 2013.

[FTI] Fault Tolerance Interface. December 2013. https://gforge.inria.fr/plugins/mediawiki/wiki/fti/images/4/47/FTI-v0.9.1-DevDoc.pdf

[GREEN500-13] The Green500 List – November 2013. http://www.green500.org/lists/green201311

[GRGRID] The Green Grid. Homepage, December 2013. http://www.thegreengrid.org

[GRGRID11] D. Azevedo, J. Cooley, M. Patterson, M. Blackburn. Data Center Efficiency Metrics: mPUE™, Partial PUE, ERE. The Green Grid, 2011. http://www.thegreengrid.org/Global/Content/TechnicalForumPresentation/2011TechForumDataCenterEfficiencyMetrics

[GRGRID12] V. Avelar, D. Azevedo, A. French. PUE™ A Comprehensive Examination of the Metric. White Paper # 49, The Green Grid, 2012. http://www.thegreengrid.org/sitecore/content/Global/Content/white-papers/WP49-PUEAComprehensiveExaminationoftheMetric.aspx

[GOM11] Leonardo Bautista-Gomez, Seiji Tsuboi, Dimitri Komatitsch, Franck Cappello, Naoya Maruyama, and Satoshi Matsuoka. FTI: high performance fault tolerance interface for hybrid systems. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis* (SC '11). ACM, New York, NY, USA, Article 32, 32 pages, 2011. DOI=10.1145/2063384.2063427 http://doi.acm.org/10.1145/2063384.2063427

[GYS5D] Gysela5D: a 5D Gyrokinetic Semi-Lagrangian code. December 2013. http://gyseladoc.gforge.inria.fr/

[NWE] The New World Encyclopaedia – Adsorption. December 2013. http://www.newworldencyclopedia.org/entry/Adsorption

[RAJ13] N. Rajovic, A. Rico, J. Vipond, I. Gelado, N. Puzovic, and A. Ramirez. Experiences With Mobile Processors for Energy Efficient HPC. In Design, Automation & Test in Europe Conference & Exhibition (DATE), pages 464–468, 2013.

[SORTECH] System Chiller Aggregate – Functional Principle. Sortech AG, December 2013. http://www.sortech.de/en/architects-designer/functional-principle/system-chiller-aggregate/

[TACONOVA] TacoSetter Tronic. Taconova, December 2013. http://www.taconova.com/en/products/pv/hydronic-balancing/1/tacosetter-tronic/13/

List of Acronyms and Abbreviations

AAA	Authorization, Authentication, Accounting.
ACF	Advanced Computing Facility
ADP	Average Dissipated Power
AMD	Advanced Micro Devices
AMFT	Advanced Multi-level Fault Tolerant prototype
APGAS	Asynchronous PGAS (language)
API	Application Programming Interface
APML	Advanced Platform Management Link (AMD)
APU	Accelerated Processing Unit
ARM	Advanced RISC Machines
ASIC	Application-Specific Integrated Circuit
ATI	Array Technologies Incorporated (AMD)
ATLAS	Automatically Tuned Linear Algebra Software
BAdW	Bayerischen Akademie der Wissenschaften (Germany)
BCO	Benchmark Code Owner
BCS	Bull Coherent Switch
BLAS	Basic Linear Algebra Subprograms
BSC	Barcelona Supercomputing Centre (Spain)
b/s	bits per second
BTL	Byte Transfer Layer (OpenMPI)
C3	Cluster Command & Control tools
CAF	Co-Array Fortran
CAL	Compute Abstraction Layer
CCE	Cray Compiler Environment
ccNUMA	cache coherent NUMA
CEA	Commissariat à l'Energie Atomique (represented in PRACE by GENCI, France)
CGS	Classical Gram-Schmidt
CGSr	Classical Gram-Schmidt with re-orthogonalisation
CIC	Cache-coherent InterConnect
CINECA	Consorzio Interuniversitario, the largest Italian computing centre (Italy)
CINES	Centre Informatique National de l'Enseignement Supérieur (represented in PRACE by GENCI, France)
CLE	Cray Linux Environment
COP	Coefficient Of Performance
CPU	Central Processing Unit
CRAC	Computer Room Air-Conditioning
CSC	Finnish IT Centre for Science (Finland)
CSCS	The Swiss National Supercomputing Centre (represented in PRACE by ETHZ, Switzerland)
CSR	Compressed Sparse Row (for a sparse matrix)
cuBLAS	CUDA BLAS library by NVidia
CUDA	Compute Unified Device Architecture (NVIDIA)
DAQ	Data AcQuisition system
DARPA	Defense Advanced Research Projects Agency
DAS	Direct Attached Storage
DDN	DataDirect Networks
DDR	Double Data Rate

DDR3	DDR version 3
DDR3L	DDR3 Low voltage
DEISA	Distributed European Infrastructure for Supercomputing Applications. EU project by leading national HPC centres.
DGEMM	Double precision General Matrix Multiply
DHCP	Dynamic Host Configuration Protocol
DIMM	Dual Inline Memory Module
DMA	Direct Memory Access
DMF	Data Migration Facility (SGI)
DNA	DeoxyriboNucleic Acid
DP	Double Precision, usually 64-bit floating point numbers
DRAM	Dynamic Random Access memory
DSP	Digital Signal Processor
EC	European Community
ECC	Error Correcting Code
EESI	European Exascale Software Initiative
EoI	Expression of Interest
EP	Efficient Performance, e.g., Nehalem-EP (Intel)
EPCC	Edinburg Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
EPSRC	The Engineering and Physical Sciences Research Council (United Kingdom)
eQPACE	extended QPACE, name of the FZJ WP8 prototype
ERE	Energy Reuse Effectiveness
ESFRI	European Strategy Forum on Research Infrastructures; created roadmap for pan-European Research Infrastructure.
ETHZ	Eidgenössische Technische Hochschule Zürich, ETH Zürich (Switzerland)
EVM	EValuation Module (hardware)
EX	Expandable, e.g., Nehalem-EX (Intel)
FC	Fibre Channel
FFT	Fast Fourier Transform
FFTW	Fastest Fourier Transform in the West
FHPCA	FPGA HPC Alliance
FMA	Fused Multiply Add
FP	Floating-Point
FPGA	Field Programmable Gate Array
FPU	Floating-Point Unit
FTI	Fault Tolerant Interface
FZJ	Forschungszentrum Jülich (Germany)
GASNet	Global Address Space Networking
GbE	Giga (= 2^{30} ~ 10^9) bit per second Ethernet
GB	Giga (10^9) Bytes (= 8 bits), also GByte
Gb/s	Giga (= 10^9) bits per second, also Gbit/s
GB/s	Giga (= 10^9) Bytes (= 8 bits) per second, also GByte/s
GCS	Gauss Centre for Supercomputing (Germany)
GDDR	Graphic Double Data Rate memory
GDDR5	GDDR version 5
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.

Gen	Generation
GENCI	Grand Equipement National de Calcul Intensif (France)
GFlop/s	Giga (= 10^9) Floating point operations (usually in 64-bit, i.e. DP) per second, also GF/s
GHz	Giga (= 10^9) Hertz, frequency =10^9 periods or clock cycles per second
GiB	2^{30} bytes
GigE	Gigabit Ethernet, also GbE
GLSL	OpenGL Shading Language
GNU	GNU's not Unix, a free OS
GPFS	General Parallel File System (IBM)
GPGPU	General Purpose GPU
GPIO	General Purpose I/O (hardware)
GPU	Graphic Processing Unit
GRES	Generic RESource
GROMACS	GROenigen MAchine for Chemical Simulations
GS	Gram-Schmidt
GWU	George Washington University, Washington, D.C. (USA)
HBA	Host Bus Adapter
HCA	Host Channel Adapter
HCE	Harwest Compiling Environment (Ylichron)
HDD	Hard Disk Drive
HE	High Efficiency
HET	High Performance Computing in Europe Taskforce. Taskforce by representatives from European HPC community to shape the European HPC Research Infrastructure. Produced the scientific case and valuable groundwork for the PRACE project.
HMM	Hidden Markov Model
HMPP	Hybrid Multi-core Parallel Programming (CAPS enterprise)
HP	Hewlett-Packard
HPC	High Performance Computing; Computing at a high performance level at any given time; often-used synonym with Supercomputing
HPCC	HPC Challenge benchmark, http://icl.cs.utk.edu/hpcc/
HPCS	High Productivity Computing System (a DARPA program)
HPL	High Performance Linpack
HT	HyperTransport channel (AMD)
HTX	HyperTransport Expansion
HWA	HardWare Accelerator
IB	InfiniBand
IBA	IB Architecture
IBM	Formerly known as International Business Machines
ICE	(SGI)
IDRIS	Institut du Développement et des Ressources en Informatique Scientifique (represented in PRACE by GENCI, France)
IEEE	Institute of Electrical and Electronic Engineers
IESP	International Exascale Project
IL	Intermediate Language
IMB	Intel MPI Benchmark
I/O	Input/Output
IOR	Interleaved Or Random
IP	Internet Protocol
IPMI	Intelligent Platform Management Interface

IPoIB	IP over IB
ISC	International Supercomputing Conference; European equivalent to the US based SC0x conference. Held annually in Germany.
iWARP	Internet Wide Area RDMA Protocol
IWC	Inbound Write Controller
JBOD	Just a Bunch of Disks
JKU	Johannes Kepler University
JSC	Jülich Supercomputing Centre (FZJ, Germany)
KB	Kilo (= 2^{10} ~10^3) Bytes (= 8 bits), also KByte
KTH	Kungliga Tekniska Högskolan (represented in PRACE by SNIC, Sweden)
LBE	Lattice Boltzmann Equation
Linpack	Software library for Linear Algebra
LLNL	Laurence Livermore National Laboratory, Livermore, California (USA)
LQCD	Lattice QCD
LRZ	Leibniz Supercomputing Centre (Garching, Germany)
LS	Local Store memory (in a Cell processor)
LU	Lower times Upper triangular (factorization)
MAID	Massive Array of Idle Disks
MB	Mega (= 2^{20} ~ 10^6) Bytes (= 8 bits), also MByte
MB/s	Mega (= 10^6) Bytes (= 8 bits) per second, also MByte/s
MDT	MetaData Target
MFC	Memory Flow Controller
MFlop/s	Mega (= 10^6) Floating point operations (usually in 64-bit, i.e. DP) per second, also MF/s
MGS	Modified Gram-Schmidt
MHz	Mega (= 10^6) Hertz, frequency =10^6 periods or clock cycles per second
MiB	2^{30} bytes
MIPS	Originally Microprocessor without Interlocked Pipeline Stages; a RISC processor architecture developed by MIPS Technology
MJ	Mega (= 10^6) Joule
MKL	Math Kernel Library (Intel)
ML	Maximum Likelihood
MLC	Multi-Level Cell
Mop/s	Mega (= 10^6) operations per second (usually integer or logic operations)
MoU	Memorandum of Understanding.
MPI	Message Passing Interface
MPP	Massively Parallel Processing (or Processor)
MPT	Message Passing Toolkit
MRAM	Magnetoresistive RAM
MTAP	Multi-Threaded Array Processor (ClearSpead-Petapath)
MUNGE	MUNGE Uid 'N' Gid Emporium (SLURM plugin)
MySQL	Open source relational database
mxm	DP matrix-by-matrix multiplication mod2am of the EuroBen kernels
NAMD	Not (just) Another Molecular Dynamics program
NAND	Negated AND (gate)
NAS	Network-Attached Storage
NB	Block size (for matrices)
NC-BTL	NumaConnect Byte Transfer Layer (see BTL, OpenMPI)
NCF	Netherlands Computing Facilities (Netherlands)

NDA	Non-Disclosure Agreement. Typically signed between vendors and customers working together on products prior to their general availability or announcement.
nm	nano (= 10^{-9}) meter
NoC	Network-on-a-Chip
NFS	Network File System
NIC	Network Interface Controller
NIS	Network Information Service
NSD	Network Shared Disk (IBM/GPFS)
NUMA	Non-Uniform Memory Access or Architecture
NVRAM	Non-Volatile RAM
OFED	OpenFabrics Enterprise Distribution
OmpSs	OpenMP StarSs programming model
OpenCL	Open Computing Language
OpenGL	Open Graphic Library
Open MP	Open Multi-Processing
OS	Operating System
OSS	Object Storage Server
OST	Object Storage Target
PAM	Pluggable Authentication Module
PB	Peta (=2^{50}) Bytes
PCI	peripheral Component Interconnect
PCIe	Peripheral Component Interconnect express, also PCI-Express
PCI-X	Peripheral Component Interconnect eXtended
PCM	Phase Change Memory
pERE	Partial ERE
PFS	Parallel File System
PGAS	Partitioned Global Address Space
PGI	Portland Group, Inc.
pNFS	Parallel Network File System
POSIX	Portable OS Interface for Unix
PPE	PowerPC Processor Element (in a Cell processor)
pPUE	Partial PUE
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PRACE-1IP	PRACE 1st Implementation Phase
PSCC	Poznanskie Centrum Superkomputerowo - Sieciowe
PSNC	Poznan Supercomputing and Networking Centre (Poland)
PSNC-BR	PSNC AMD Brazos prototype
PSNC-IB	PSNC Intel Ivy Bridge prototype
PSNC-ICE	PSNC Iceotope protoype
PSNC-LL	PSNC AMD Llano prototype
PSNC-SB	PSNC Intel Sandy Bridge prototype
PSNC-TR	PSNC AMD Trinity prototype
PSU	Power Supply Unit
PUE	Power Usage Effectiveness
QCD	Quantum Chromodynamics
QCDOC	Quantum Chromodynamics On a Chip
QDR	Quad Data Rate
QPACE	QCD Parallel Computing on the Cell
QR	QR method or algorithm: a procedure in linear algebra to compute the eigenvalues and eigenvectors of a matrix

RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RCUDA	Remote CUDA
RDMA	Remote Data Memory Access
RISC	Reduce Instruction Set Computer
RNG	Random Number Generator
ROI	Return On Investment
RPM	Revolution per Minute
RS-232	Radio Sector 232 standard by the Electronic Industries Association
RTM	Reverse Time Migration
SAN	Storage Area Network
SARA	Stichting Academisch Rekencentrum Amsterdam (Netherlands)
SAS	Serial Attached SCSI
SATA	Serial Advanced Technology Attachment (bus)
SCS	SuperComputing Solutions (a CINECA company)
SDK	Software Development Kit
SGEMM	Single precision General Matrix Multiply, subroutine in the BLAS
SGI	Silicon Graphics, Inc.
SHMEM	Share Memory access library (Cray)
SHOC	Scalable HeterOgeneous Computing
SIMD	Single Instruction Multiple Data
SLC	Single Level Cell
SLURM	Simple Linux Utility for Resource Management
SM	Streaming Multiprocessor, also Subnet Manager
SM-BTL	Shared Memory Byte Transfer Layer (see BTL, OpenMPI)
SMP	Symmetric MultiProcessing
SMX	Streaming Multiprocessor (see also SM)
SNIC	Swedish National Infrastructure for Computing (Sweden)
SoC	System-on-a-Chip
SP	Single Precision, usually 32-bit floating point numbers
SPE	Synergistic Processing Element (core of Cell processor)
SPH	Smoothed Particle Hydrodynamics
SPU	Synergistic Processor Unit (in each SPE)
SSD	Solid State Disk or Drive
SSU	Scalable Storage Unit (Xyratex)
STFC	Science and Technology Facilities Council (represented in PRACE by EPSRC, United Kingdom)
STRATOS	PRACE advisory group for STRAtegic TechnOlogieS
STREAM	Streaming memory benchmark
STT	Spin-Torque-Transfer
TARA	Traffic Aware Routing Algorithm
TB	Tera (= 240 ~ 1012) Bytes (= 8 bits), also TByte
TCO	Total Cost of Ownership. Includes the costs (personnel, power, cooling, maintenance, ...) in addition to the purchase cost of a system.
TDP	Thermal Design Power
TFlop/s	Tera (= 10^{12}) Floating-point operations (usually in 64-bit, i.e. DP) per second, also TF/s
TGCC	Tres Grand Centre de Calcul
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

TMS	Texas Memory System
U	Rack unit = 1.75 inches = 44.45 mm
UEABS	Unified European Application Benchmark Suite
UFM	Unified Fabric Manager (Voltaire)
UiO	University in Oslo, Norway
UNICORE	Uniform Interface to Computing Resources. Grid software for seamless access to distributed resources.
UPC	Unified Parallel C
UV	Ultra Violet (SGI)
UVA	Unified Virtual Addressing (NVIDIA)
µSD	Micro Secure Digital
VHDL	VHSIC (Very-High Speed Integrated Circuit) Hardware Description Language
VTL	Virtual Tape Library
X86	Instruction set architectures backward compatible with the Intel 8086 CPU
X86-64	x86 64-bit instruction set architectures
xCAT	EXtreme Cloud Administration Toolkit (IBM)

Executive Summary

This report supplements deliverable D9.3.3 [D933] and covers future technology prototype efforts from March 2013. Three prototype efforts not reported on in D9.3.3 are fully reported here: the ARM based NVidia Tegra3 with NVidia Kepler GPU acceleration prototype at BSC; the energy recovery prototype using immersive cooling techniques at PSNC; and the results from the Advanced Multi-level Fault Tolerant prototype. Additional results are provided for the x86+GPU prototype at CaSToRC, the DSP prototype at SNIC/KTH, and the energy recovery prototype at LRZ.

The DSP results show that for the benchmarks that were well optimised the observed energy efficiency for the DSP was in line with the nominal energy efficiency that can be derived from specifications. Thus, the energy efficiency is comparable to that of GPUs and better than that of x86 CPUs. Furthermore, the DSP does not require a host processor. In fact, the 40 nm DSP used is, for the optimised benchmarks, more energy efficient than correspondingly optimised codes for the 22 nm Intel Ivy Bridge. ARM Coretx-A9 CPUs are not energy efficient for HPC workloads, but nodes using powerful and energy efficient acceleration, such as the NVidia Kepler, can form part of a viable HPC node. Due to late delivery of the relevant prototype only preliminary results are available at this time.

The shared memory prototype is based on standard x86 servers and a node add-in card for cache coherency. This prototype clearly demonstrated that, though a very large shared memory address space is supported, it is a NUMA architecture and paying proper attention to the memory architecture is necessary for good performance.

One of the lessons learnt in the project is that delivering energy efficiency advantages for architectures different from x86 architectures requires well-optimised codes also for the non-x86 architectures, i.e. for significant energy efficiency gains at the application level resource use needs to be at a level comparable to that for x86 architectures. High resource utilization requires detailed understanding of the alternative architectures as well as developing or inventing new programming methodologies and techniques in a less-rich software ecosystem than that available for x86 architectures today. It also became apparent that quality energy efficiency assessment adds significant complexity to benchmarking, in that instrumentation and measurement technologies as well a good understanding of applications and system behaviours are all necessary.

For the energy recovery approach assessed by the LRZ prototype it was demonstrated that a 20% energy recovery could be achieved even for the partially hot-water liquid cooled system Valuable insights were also gained into how the approach used can be improved for increased energy recovery. The immersive cooling prototype at PSNC exposed some of the many engineering issues in integrating new cooling technologies in a data centre setting.

The Advanced Multi-level Fault Tolerant approach to checkpoint/restart was demonstrated to have good scalability and sufficiently low overhead to allow for frequent checkpointing even at a large scale.

1 Introduction

This deliverable is a supplement to deliverable D9.3.3 covering the period March – December 2013. Deliverable D9.3.3 provides the background for the Future Technologies prototype efforts within the PRACE-1IP Work-Package 9 (WP9) and is still valid and so is not repeated in this deliverable. [D933] The main focus of the PRACE-1IP WP9 efforts was on technologies for energy efficient HPC systems, especially large-scale systems. High energy-efficiency does require high efficiency in resource utilization. With current technologies for power management and system architectures poor efficiency in utilizing system resources result in unnecessary (high) energy consumption. Therefore the WP9 efforts have had a strong emphasis on assessing the ability and efforts to achieve high efficiency and scalability, in addition to a focus on architectures with low power design objective, such as architectures for mobile and embedded markets. Also, high efficiency and energy efficiency require highly reliable or fault tolerant systems in order for computations not to need to be attempted time and again from the start for a successful outcome. This latter issue is addressed by the Advanced Multi-level Fault Tolerance (AMFT) prototype, a prototype effort that was not reported on in D9.3.3. Another prototype effort that was not covered in D9.3.3 is the assessment of achievable energy efficiency of clusters based on nodes using ARM CPUs with NVidia GPU accelerators and high-performance interconnect. A third prototype, PSNC-ICE, exploring immersive cooling technologies for energy recovery is also reported on here for the first time. For all three prototypes delayed delivery and/or engineering issues associated with new technologies are the reasons for results not being included in D9.3.3. Stability problems with the shared-memory system, based on the use of a cache-coherency add-in card that prevented other than highly preliminary information in D9.3.3, have been resolved; this allowed for STREAM, HPL and MPI benchmark results to be reported here together with a report on experiences in using the large core-count NUMA shared-memory system. This deliverable also reports on additional results and insights gained from the DSP, the x86+GPU and energy recovery prototypes.

Table 1 lists the various prototypes in PRACE-1IP WP9 and indicates which prototype efforts are reported for the first time here (added) and for which this deliverable provides supplementary information (updated).

The organization of this deliverable follows that of D9.3.3 with architectural prototype efforts reported first, followed by a report of new results and insights related to energy recovery from HPC system cooling systems. The report from the AMFT prototype is reported last.

Abbreviation	Prototype Name		Site	Reported in
BSC-1	*An NVidia Tegra 2 mobile SoC based HPC cluster*		*BSC*	*D 9.3.3*
BSC-2	An NVidia Tegra 3 SoC with GPU HPC Node		BSC	D 9.3.4
CaSToRC	GPU-GPU communication over PCIe and IB		CaSToRC	Both
CEA	*Exascale I/O*		*CEA/CINES*	*D 9.3.3*
FZJ	*Exascale integrated I/O subsystem*		*FZJ*	*D 9.3.3*
JKU	*FPGA matrix computation acceleration*		*JKU*	*D 9.3.3*
LRZ	Holistic approach to energy efficiency		LRZ	Both
PSNC-BR	*On die integrated CPU and GPU*	*AMD E-350*	*PSNC*	*D 9.3.3*
PSNC-IB		*Intel i7-3770*		*D 9.3.3*
PSNC-LL		*AMD A8-3870*		*D 9.3.3*
PSNC-SB		*Intel i7-2600*		*D 9.3.3*
PSNC-TR		*AMD A10-5800K*		*D 9.3.3*
PSNC-ICE		Intel Xeon E5-2620		D 9.3.4
SNIC	DSP based node for HPC		SNIC/KTH	Both
UiO	Shared memory through a cache-coherency add-in card (NUMA-CIC)		UiO	Both
AMFT	Advanced Multi-level Fault Tolerance		GENCI/CEA/CINES	D 9.3.4

Table 1 Abbreviations for the prototypes used in the graphs and this report.

2 Compute Node Architecture Prototypes

Results on the ARM based prototype with GPU acceleration are reported first, followed by an update on the GPU to GPU communication prototype, some very preliminary first results from the immersion cooling prototype and updated results from the DSP HPC node prototype. This is the same order as in the original D9.3.3 deliverable.

2.1 An NVidia Tegra3 SoC with GPU HPC Node, BSC

The BSC–2 prototype is a hybrid accelerator prototype based on ARM SoCs, NVidia Tesla K20 GPUs and an QDR InfiniBand (40 Gb/s) interconnect. It has considerably higher compute performance and network bandwidth than the previous BSC–1 prototype. This prototype is currently being used to test the portability and scalability of scientific applications that previously executed on GPU-accelerated x86-64 multicore clusters. Of special interest are the impact of the host's low (CPU) performance, memory capacity and memory bandwidth, and the benefits of its low power consumption.

Results for this prototype are reported for the first time in this deliverable.

2.1.1 *Hardware*

The prototype is composed of 72 compute nodes, each of which contains the components shown in Figure 1(a-d). The system-on-module (a) has one NVidia Tegra 3 SoC with four ARM Cortex-A9 cores clocked at 1.3 GHz, a single 32-bit memory channel and 2 GiB 1500 MHz DDR3L DRAM. The peak floating-point performance is 5.2 GF/s (double precision, 10.4 GF/s single precision) and the peak memory bandwidth 6 GB/s. The prototype nodes support 4 lanes[1] of PCIe Gen 2.0. The NVidia Tesla K20 GPU (b) has one GK110 GPU containing 2,496 cores in 13 Streaming Multiprocessors (SMX) and 5 GiB of GDDR5 memory, giving a compute performance of 1.17 TF/s (double precision, 3.53 TF/s single precision). The Mellanox ConnectX-3 (c) connects the node to the QDR InfiniBand interconnect at 40 Gb/s. A PCIe riser card (not shown) allows the GPU and InfiniBand card to share a single PCIe connector. The carrier board (d) holds the system-on-module daughter board, has one PCI connector, and provides all external interface connectors except InfiniBand. A 256 GB Samsung SSD provides node local storage.

All components for two nodes are contained in a 3U chassis. The prototype hardware also includes a single node in a desktop form factor. The latter configuration does not include the PCIe riser and the InfiniBand card, but is otherwise the same.

The full system is assembled into four Bull bullx 1200 (42U) racks, as shown in Figure 2. In addition to the 72 compute nodes, there are six spare nodes, two login nodes, four 1GbE switches for storage, five 36-port QDR InfiniBand switches for the MPI interconnect, and space reserved for an NFS server. Each login node has two processor sockets, each containing an Intel Xeon E5-2620, as well as 32 GiB 1600 MHz DDR3 DRAM, two 500 GB SATA disks in a RAID 1 configuration, four 1GbE ports and one QDR InfiniBand ConnectX-3 interface.

The complete prototype provides a peak nominal performance of 84.2 TF/s (double precision)

[1] Tegra 3 supports up to six lanes of PCIe, but only with three devices, each with 2 lanes, which is not the configuration used in this prototype.

at 18.7 kW[2] power consumption resulting in a peak energy efficiency of 4.5 GF/J based on the nominal peak performance, measured peak power consumption and maximum specified power for the switches. This nominal energy efficiency shows potential for competitive realised energy efficiency compared to current systems, if a good fraction of the peak performance is achieved in applications. For comparison, the highest HPL (High-Performance Linpack) energy efficiency in the November 2013 Green 500 list is 4.5 GF/J (TSUBAME-KFC, Tokyo Institute of Technology), also using NVidia K20 accelerator technology.

(a) CPU: System-on-module

(b) GPU: NVidia K20

(c) Mellanox ConnectX-3

(d) Carrier board

Figure 1 Hardware in each BSC-2 node.

Figure 2 Layout of Pedraforca (BSC-2) racks, each 3U enclosure contains two nodes.

[2] Total double precision performance: 72 nodes, each at 1.17 TF/s. Total power consumption including InfiniBand and Ethernet switches: 72 nodes, measured at 250 W each, plus five InfiniBand 36-port InfiniScale IV QDR, each 122 W, plus 4 four 26-port SMC8126L2 Ethernet switches, each at 38 W The total power consumption is equivalent to 260 W per node.

The prototype has two independent physical networks. The MPI interconnect, as shown in Figure 3, uses QDR InfiniBand (40 Gb/s) with a two-level tree topology. The storage network connects to the outside world and storage via 1GbE.

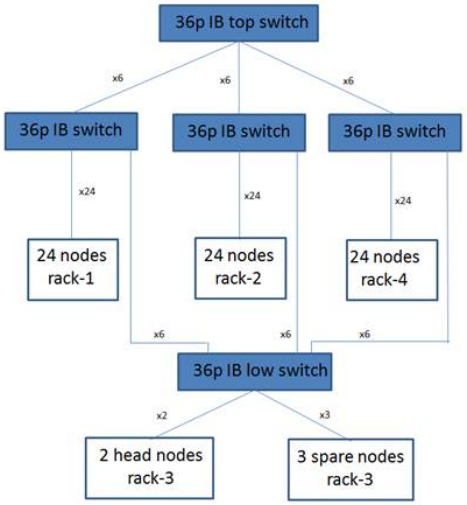


Figure 3 InfiniBand network topology for the BSC-2 cluster.

2.1.2 *Software*

Login node 1 is focused on user activities. It includes the SLURM (Simple Linux Utility for Resource Management) resource and queue manager and the NFS server for the cluster. Login node 2 is focused on administrative tasks. It runs the NIS (Network Information Service) user management, the Ganglia monitor and the DHCP (Dynamic Host Control Protocol) server for the cluster. Both login nodes have the C3 (Cluster Command & Control) parallel shell installed.

The prototype supports native compilation on the nodes, including C, C++ and Fortran, as well as CUDA 5.5 and OpenCL. The Mercurium-based compiler has been installed for OmpSs (multi-core only, excluding GPU). Library support includes OpenMPI, OpenMP and BLAS, as well as other libraries developed for the prototype effort.

Target applications for the prototype, which have already been ported to x86 + GPU, include:

- RTM (Kaleidoscope project with Repsol), ELIAUS (Pegase project with PROMES)
- NAMD, GROMACS (both part of the PRACE benchmark suite)
- HPL
- GPU offloading using RCUDA

2.1.3 *Node architecture prototype measurement setups*

For energy-to-solution assessment, the power measurement setup summarised in Table 2 was used. Voltage and current at the 220 V AC input to the node(s) was measured using a Yokogawa WT230 power meter, as illustrated in Figure 4. The power measurements include losses in the PSU (Power Supply Unit), as well as the power consumption of the forced air-cooling.

No.	Device	Scope/Purpose	Measurand	Error	Sampling
1	Yokogawa WT230	Power measurement of a subset of nodes	Active power (W) Voltage (V) Current (A)	± 0.2%	4/sec

Table 2 BSC-2 prototype measurement equipment characteristics.

Results are reported for one or two nodes. Although two nodes share a 3U chassis, the power consumption of a single node can be isolated, since the two nodes have separate AC inputs. The measurements are scaled with the number of nodes that were included in the benchmark. We verified that the per-node load does not vary significantly among the nodes within a run, so that measuring a single node's power and scaling it to the whole system does not introduce a significant error. The power consumption of the various nodes inside a rack is expected to depend only on the workload, and to be independent of the node's location inside the rack. Since all fans are always running at full speed, we do not expect the nodes in less well-ventilated parts of the rack to incur higher power consumption from the fans. The power consumption of the networking equipment has not yet been measured.

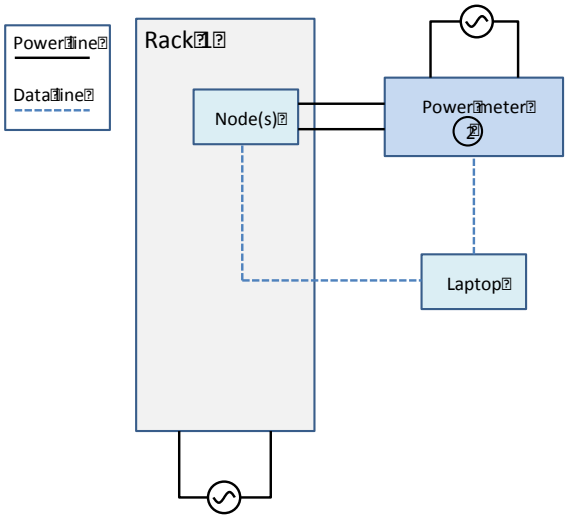


Figure 4 Node power measurements of the BSC-2 cluster.

During power measurements, due to the problems with SLURM mentioned in the conclusions, the job scheduler was bypassed and jobs directly mapped to nodes using the mpiexec command. The power meter was connected to a laptop using a serial cable. A driver script was used to fully automate the process of collecting data; the laptop was instructed to collect power measurements while the benchmark was running.

2.1.4 *Measurement results*

STREAM

Version 5.10 of the STREAM benchmark was used out of the box with no hand optimisation. The only changes made were to allow for separate power measurement of each operation: copy, scale, sum and triad. STREAM was compiled using GCC 4.6.3 and the flags "-Ofast -mcpu=cortex-a9 -mtune=cortex-a9 -mfloat-abi=hard -vfpu=vfpv3-d16". These flags were found to create the best results; since the ARM Cortex-A9 Neon unit has scalar double-precision support, there was no benefit from "-fpu=neon -ftree-vectorise". The throughput results are shown in Table 3 for the cluster node and Table 4 for the desktop node. OpenMP

was used for multi-threaded benchmarks. Figure 5 shows the results for the cluster node.

It is likely that the performance results could be considerably improved by hand-optimising the implementation of the STREAM benchmark, but that has not yet been done. Though the efficiency at best reaches about 27%, it appears that practically the peak efficiency is reached for two threads, and that the energy efficiency is reduced for four threads, i.e. when all cores are used. From an energy-efficiency perspective two threads are optimal with the current STREAM benchmark implementation. It is also interesting to observe that two threads (with increased memory activity and two active cores) increased the power consumption with 0.3 – 0.4 W for copy and scale compared to a single thread. Adding two additional threads/cores increased the power consumption with about 1.1 – 1.2 W (or about 0.6 W/core) for the two added active cores, without a significant increase in performance or memory activity.

Op.	Threads	Perf. [MB/s]	Power W	Eff. %	E.Eff. [MB/J]
Copy	1	1229	68.8	20.5	17.86
	2	1633	69.2	27.2	23.60
	4	1633	70.3	27.2	23.23
Scale	1	1299	69.0	21.7	18.83
	2	1610	69.4	26.8	23.20
	4	1591	70.5	26.5	22.57
Sum	1	750	68.9	12.5	10.89
	2	1247	69.2	20.8	18.02
	4	1281	70.4	21.4	18.20
Triad	1	755	68.9	12.6	10.96
	2	1140	69.3	19.0	16.45
	4	1154	70.4	19.2	16.39

Table 3 STREAM results for the BSC-2 cluster node.

Op.	Threads	Perf. [MB/s]	Power W	Eff. %	E.Eff. [MB/J]
Copy	1	1229	32.3	20.5	38.05
	2	1633	32.7	27.2	49.94
	4	1633	33.9	27.2	48.17
Scale	1	1299	32.3	21.7	40.22
	2	1610	32.9	26.8	48.94
	4	1591	34.1	26.5	46.66
Sum	1	750	32.4	12.5	23.15
	2	1247	32.7	20.8	38.13
	4	1281	33.9	21.4	37.79
Triad	1	755	32.7	12.6	23.09
	2	1140	32.8	19.0	34.76
	4	1154	34.1	19.2	33.84

Table 4 STREAM results for the desktop node (same as cluster node exclusive of fan and IB card).

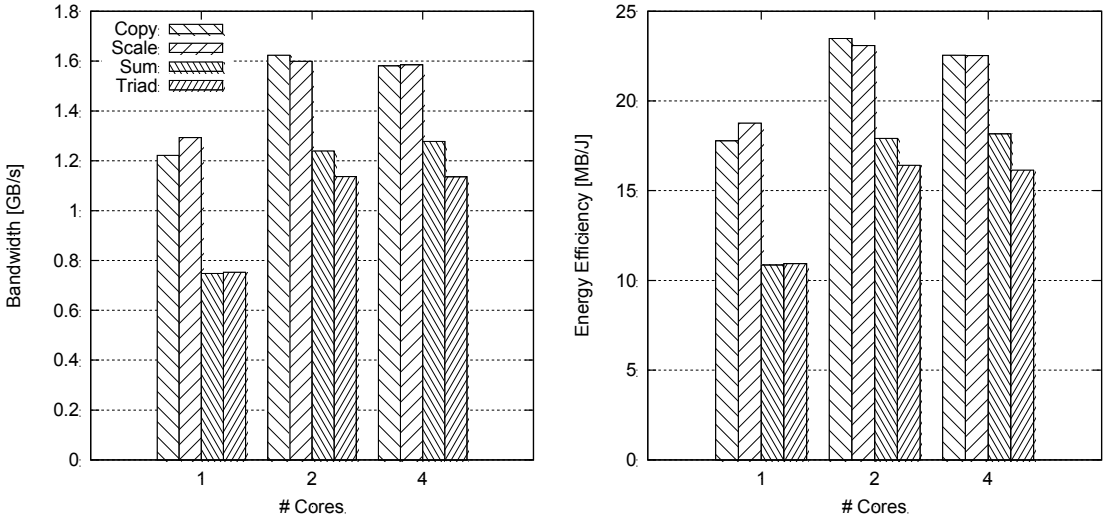


Figure 5 Bandwidth and energy efficiency of the STREAM benchmark on the BSC-2 prototype.

Dense matrix multiply benchmark

The dense matrix multiply code from the Mont-Blanc kernel benchmarks [RAJ13] was used together with the auto-tuned ATLAS 3.11.14 library for the ARM CPUs and with the CUBLAS 5.5.22 library from CUDA-5.5 for the GPUs. The GPU results are 2.5 TF/s single precision (71.0% efficiency) and 1.04 TF/s double precision (88.9% efficiency). The corresponding energy efficiencies are 10.16 GF/J for single precision at a node power consumption of 246 W and 4.19 GF/J for double precision at a node power consumption of 248 W. The former is achieved for matrices of size 3584 and above, which is 51 MB per matrix, and the latter is achieved for matrices of size 2048 or above, which is about 34 MB per matrix. The results are shown in Table 5 and Figure 6.

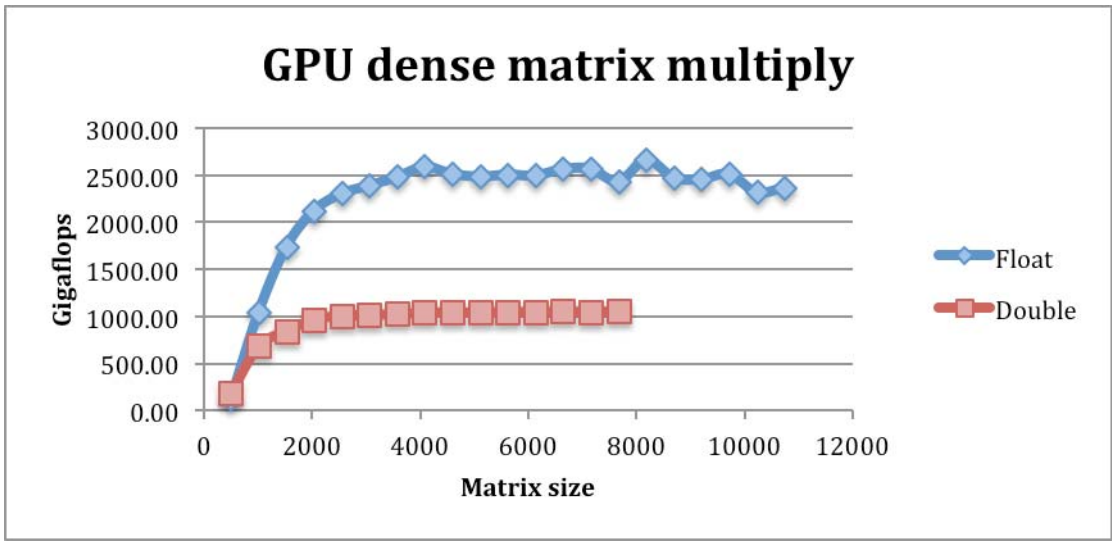
The peak observed dense matrix multiplication performance of a single ARM Cortex-A9 core was 1.6 GF/s in single precision (61.5% efficiency) and 0.9 GF/s in double precision (69.2% efficiency). With OpenMP, the performance with four threads increased to 5.16 GF/s in single precision (49.6% efficiency) and 2.43 GF/s double precision (46.7% efficiency). Unfortunately, large power overheads mean that the CPU-only results achieve poor energy efficiency. As described on page 10, the CPU, memories and other active components account for less than 25% of the power consumption; the rest is consumed by the inactive GPU and fans. However, it is interesting to note that; for single precision, adding three cores increased the power consumption with 2.7 W (or on average 0.9 W/core), and, for double precision, the power increase is 2.1 W, or 0.7 W/core. Though more careful analysis is needed the results indicate that the ARM Cortex-A9 cores may have an energy efficiency of close to 2 GF/J in single precision and about 1 GF/J in double precision.

Op.	Precision	Matrix Size	Perf. [GF/s]	Power W	Eff. %	E.Eff. [GF/J]
GPU	SP	4096	2500	245.7	71.0	10.18
	DP	4096	1050	247.2	89.4	4.25
CPU 1-core	SP	2560	1.60	68.7	61.5	0.023
	DP	512	0.90	68.7	69.2	0.013
CPU 4-cores	SP	5120	5.16	71.4	49.6	0.072
	DP	4608	2.43	70.8	46.7	0.034

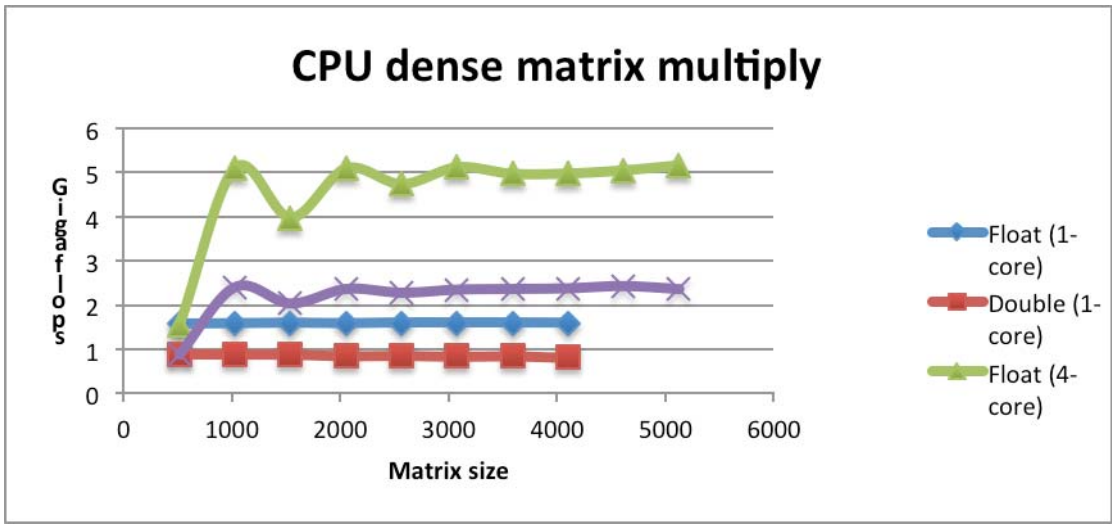
Table 5 Matrix multiplication performance on the CPU and GPU respectively for the cluster node.

Analysis of power consumption

The power measurement results for the BSC-2 cluster are summarised in Figure 7. The blue bars give the power consumption of the single-node Arka desktop unit and the red bars the power consumption of a Pedraforca cluster node. The green bars give the difference between the two. The only differences between the cluster and desktop nodes are the PCIe riser card, the InfiniBand card, and the fans. The total difference was almost constant, at about 36 W, and the largest contribution, at least 25 W, based on the fan's specification and informal measurements at E4 Computer Systems, is certainly from the inefficient fans. Work is ongoing to lower the power consumption and noise level of the fans.



(a) Per-node GPU results.



(b) Per-node CPU results.

Figure 6 Per-node dense matrix-matrix multiplication on the BSC-2 prototype.

The power consumption of the Arka desktop node, with CPU and GPU idle, is 30 W. The power consumption of a single idle Pedraforca node, for the reasons discussed above, is 67 W. In both cases, executing DGEMM on a single CPU core increases power consumption by about 2 W, and executing it on all four cores increases power consumption by 4 – 5 W. In order to determine how much of this is due to the K20 GPU, we removed the GPU from the Arka desktop node, and measured a power consumption of 10 W. Hence, for CPU-only workloads, a power consumption of about 15 W, for all components on the SECO system-on-

module and carrier board, is inflated by 373% to 71 W.

For GPU workloads, the power consumption overheads are less significant. For DGEMM on the GPU, the total power consumption for a Pedraforca node was measured to be 248 W. Of this, the fans, idle CPU and other components outside the GPU account for about 40 W. The overhead of all components except the GPU is therefore low, at about 20%. Finally, we measured the power consumption of the SHOC (Scalable HeterOgeneous Computing) triad benchmark, which is a GPU triad operation, similar to that of STREAM, operating from host memory. This shows the power consumption of the PCIe traffic that was about 350 MB/s.

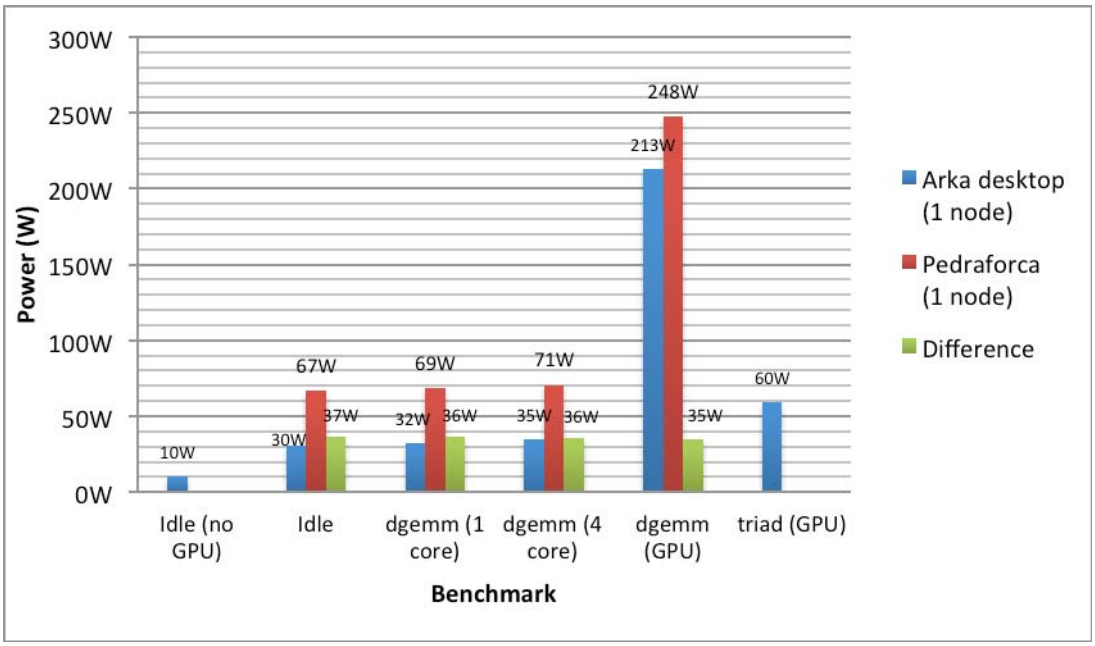


Figure 7 Power measurement results on the BSC-2 prototype.

2.1.5 *Conclusions*

Due to delays in the delivery of the Pedraforca prototype and several technical issues it has only been possible to collect results for a subset of the benchmark suite. Work is ongoing on resolving the problems outlined below in order to be able to operate the system using a wider range of full applications.

The most critical outstanding technical problem is that the InfiniBand network is not working. The Mellanox InfiniBand drivers do currently not support RDMA on our system, due to the lack of coherency between the CPUs and PCIe. This limitation was not known until installation time. In order to use InfiniBand, albeit at lower performance, InfiniBand was configured to use IPoIB (IP over InfiniBand). However, the throughput was less than 25 MB/s, i.e. much lower than that of Ethernet (100 MB/s, observed for benchmarks on Pedraforca). As a result, we can only use the Ethernet network, giving a peak network throughput of 1 Gb/s rather than the expected 40 Gb/s, and there is no support for GPUDirect. Currently, Mellanox has provided no timeframe for the resolution of this problem.

A second problem is that the idle power consumption is much higher than expected. Details are given in the results section. The fans, running constantly at full speed, consume about 25 W per node and they generate a lot of noise. Work to fix this problem is ongoing. In addition, the K20 GPU at idle consumes about 20 W per node. For CPU-only workloads, therefore the total per node power consumption is inflated to almost 70 W, although the SECO system on module and carrier board on their own require only about 15 W.

Various other niggles were encountered. When the prototype was first installed, a bug in the Ethernet driver caused the Ethernet network to be unstable, so that nodes sometimes became inaccessible, frustrating development work. This problem was resolved in mid-November 2013, by changing the PCIe Ethernet NIC. In addition, the SLURM job scheduler could not be used because it expects the core count reported by the Linux kernel to include all cores, whereas the kernel excludes cores in power-saving mode. This problem may soon be fixed.

Despite the above problems, certain aspects of the prototype have been a success. Firstly, the system has excellent energy efficiency for matrix multiply: double precision at 4.2 GF/J (1.04 TF/s in 248 W) and single precision at 10.2 GF/J (2.50 TF/s in 246 W), even including the current inefficient fans. If the network problems can be resolved, this result shows that GPU-dominated workloads can potentially achieve very high energy-efficiency.

The Pedraforca prototype also has high impact, beyond the performance of the actual machine. Firstly, the prototype is contributing to the maturity of the software stack for ARM-based HPC systems. Although the Mellanox InfiniBand drivers do not yet support RDMA, InfiniBand on ARM is only under development through the efforts of the Pedraforca cluster. Once the stability problems are resolved and BSC-internal benchmarking is complete, the 72-node cluster will be made available to the partners of the PRACE and Mont-Blanc projects and other researchers who wish to begin porting large-scale CUDA applications to ARM systems.

The second real impact of the Pedraforca prototype is its commercialization by E4 Computer Engineering. The prototype system architecture corresponds exactly to the Arka Extreme platforms, in the company's Arka series of ARM-based platforms. Arka EK002 is the 3U chassis used in Pedraforca, which features NVidia Tegra 3 and K20 GPU and QDR InfiniBand, and is targeted at applications such as seismic processing, signal and image processing, video analytics and traffic analysis. Arka EK001 is a single-node version, without InfiniBand, in a desktop form factor.

2.2 GPU-GPU communication over PCIe and IB, CaSToRC

Work on the CaSToRC prototype consisted of two tasks. First the SLURM resource manager was installed to gain experience with scheduling accelerator (GPU) resources. Second the scaling behaviour of the prototype was studied using the HPL benchmark.

Earlier results for this prototype are reported in D9.3.3 [D933 p. 39ff].

2.2.1 *Experiences with SLURM on scheduling GPUs*

The main goal was to set up the SLURM utility as the main scheduler on the eight-node "Prometheus" cluster, a phase-1 PRACE-1IP WP9 prototype. Each node has two GPU accelerators and two CPUs. Details of the cluster are presented in deliverable D9.3.3. For the installation it was necessary to configure PAM (Pluggable Authentication Module) to prevent locked memory limit propagation to the compute nodes, and declaring the generic resources (GRES) to be managed. Within this context, a generic resource is any computation unit that co-operates with the CPU, such as GPUs, accelerators, etc. With SLURM, the user allocates a generic resource by using the "gres" parameter during job submission. However, in the default configuration, GPUs are not pinned to the user, meaning a user can use a GPU of an allocated node, if it exists, whether the option is specified or not. There are two situations in which this may be problematic: 1) in a shared system for which GPU resources are accounted for separately from CPU resources, this setup will not allow reliable accumulation of GPU time used; 2) on systems where multiple jobs are allowed to run on the same node, this setup does not exclude undesirable situations where multiple users will compete on the same GPU. The first attempt to solve this issue was setting the default value for the CUDA_ENABLE_DEVICES environment variable to a null value and making it read-only for all users. The SLURM controller, which runs with root privileges, would then set this variable according to the user's "gres" requirements. It was found that placing the configuration files under the same directory tree as the SLURM installation solved the issue.

SLURM authentication requires an external mechanism for which MUNGE is recommended by the SLURM developers. With MUNGE, which interfaces to SLURM as a plugin, authentication services can be configured such that, e.g. users are granted access to compute nodes only through the submission of jobs through the head node. Additionally, for accounting purposes a MySQL database was set up for recording and accounting of resource usage. The HPL power efficiency measurements also served as a use case to test the SLURM installation.

2.2.2 *Power Measurement Characterisation*

Blade level power measurements were taken using the *rvitals* utility included in the IBM Extreme Cloud Administration Toolkit (xCat). Unfortunately two effects distorts the power measurements: the power readings seem to be averaged over a period of about 40 seconds, comparable to the time of some of the benchmarks, and the system temperature needs about 10 minutes to stabilise causing about a 50 W increase in power consumption, probably due to increased fan speed.

Both effects can be seen when using repeated execution of matrix-matrix multiplication (DGEMM) to simulate a step load change. The short-term averaging can best be seen at the tail end of the benchmark shown in Figure 8. The long-term temperature effects are visible in the power profile over the whole benchmark execution shown in Figure 9, which correlates

well with the temperature measurements shown in Figure 10 and Figure 11.

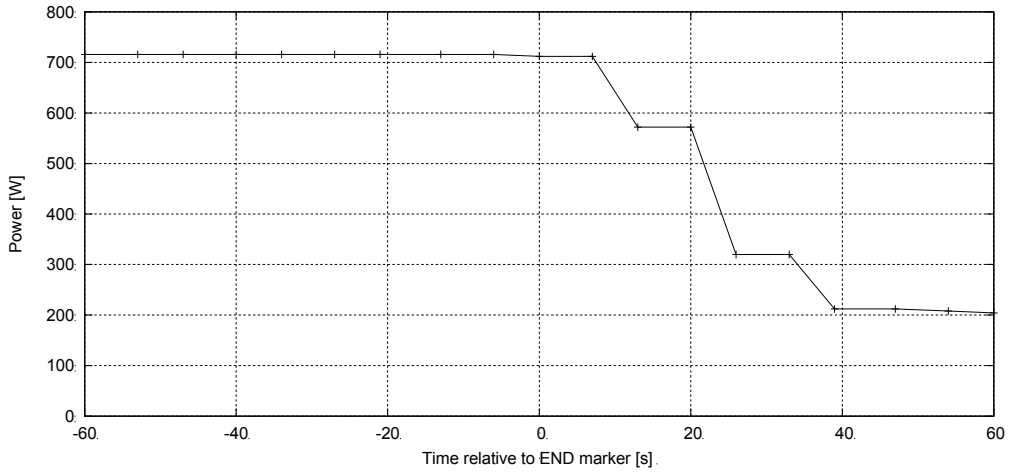


Figure 8 Power measurements around the end of the 2 GPU matrix-multiply benchmark on the CaSToRC prototype.

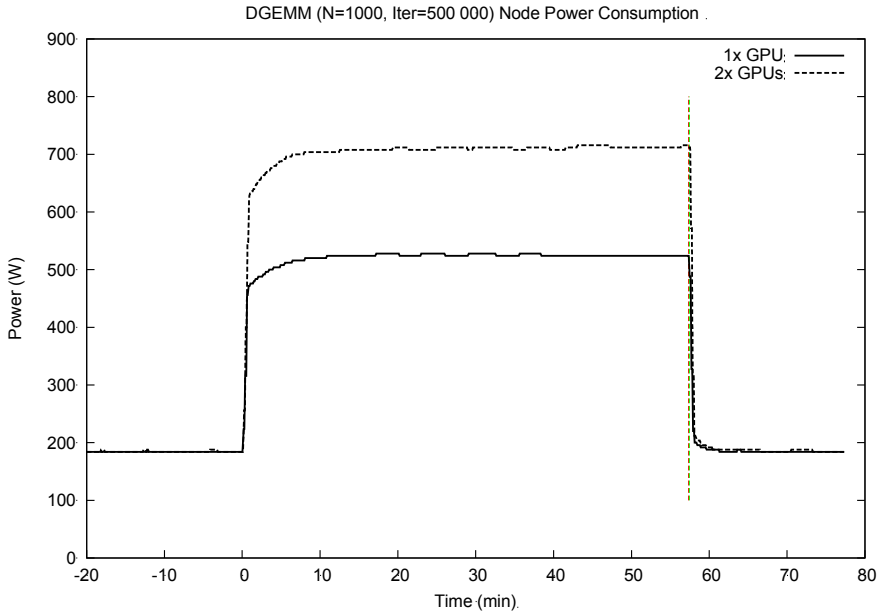


Figure 9 Power profile of repeated matrix-matrix multiplication on the CaSToRC prototype.

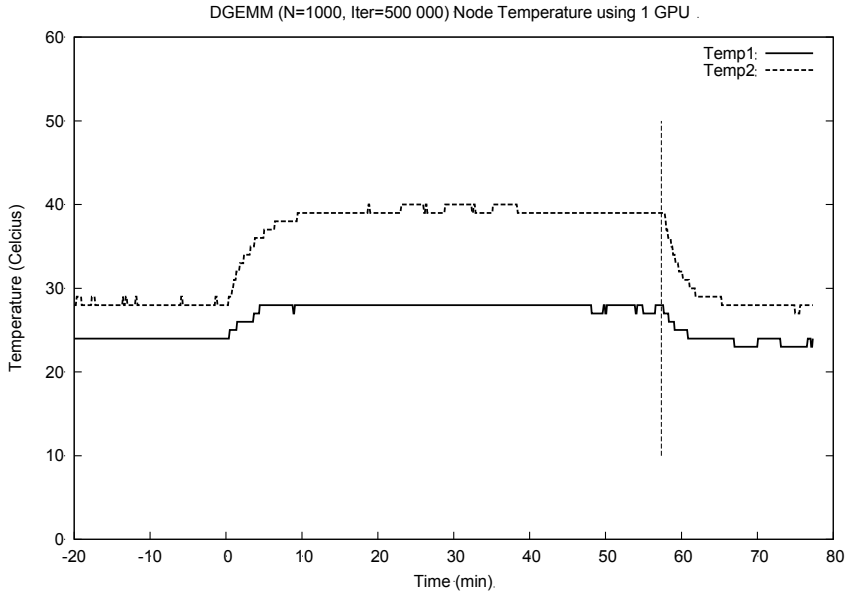


Figure 10 Temperature profile captured during 1 GPU benchmark run shown in Figure 9.

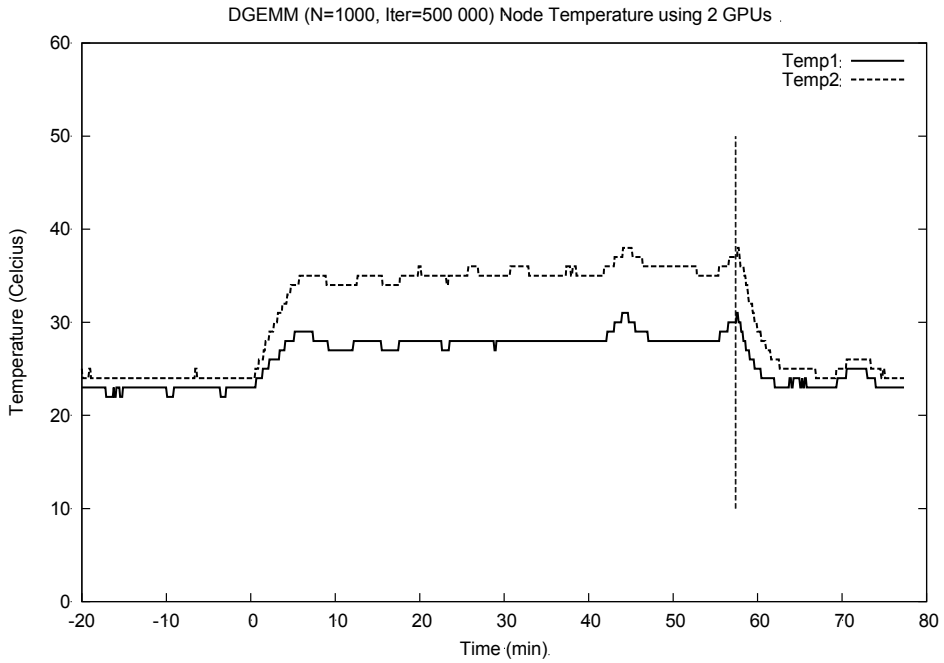


Figure 11 Temperature profiles captured during 2 GPU benchmark run shown in Figure 9.

Figure 12 illustrates the averaging effect on the power data gathered during execution of the HPL benchmark. In this example, the reported run time was 127 seconds and the start and end of the benchmark execution is shown in the graph compared to the power values. Clearly visible is the tail of elevated power readings after end of the benchmark. Furthermore, since the NVidia HPL code could not be instrumented the markers include initialization and verification time, which are much less compute intensive. Based on the reported performance of 544.9 GF/s, the actual LU decomposition took only about 51 s. This calculated run-time roughly corresponds to the high power values between 50 and 100 s in Figure 12. Comparing to the timescales in Figure 9, the longest HPL runs took less than 4 minutes. Thus, the energy measurements for HPL are dominated by measurement errors due to the averaging effect.

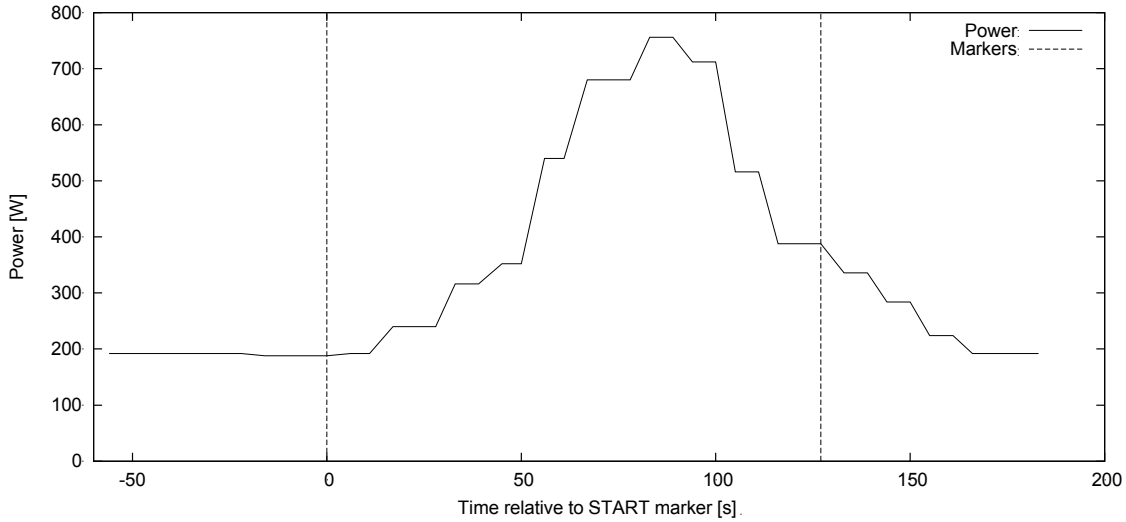


Figure 12 Power measured during a HPL run using N=34 756 equations on 2 GPUs.

2.2.3 *Energy Efficiency Estimation for Matrix Multiplication*

Based on the data gathered from the repeated execution of the matrix-matrix multiplication of two 1000x1000 element matrices, we calculated the energy efficiency for a single node of the cluster. The power values for this calculation were averaged from the steady state, flat-top,

part of Figure 9 using samples, beginning 20 minutes after benchmark start until 10 minutes before benchmark end to eliminate the influence of the averaging. The selected time window contained 246 respective 249 samples with a standard deviation below 1%, better than the estimated accuracy of the power measurement equipment. As a comparison, the leading system equipped with NVidia 2070 accelerators on the November 2013 Green500 list (rank 98, CINECA/SCS) reached 0.892 GF/J, comparable to the two GPU case. The results are shown in Table 6.

GPUs	Perf. [GF/s]	Eff.	Avg. Power [W]	Energy Eff. [GF/J]
1	290.6	0.56	525	0.55
2	580.4	0.56	711	0.82

Table 6 Performance and energy efficiency for matrix-matrix multiplication on the CaSToRC prototype.

2.2.4 *HPL Benchmark Performance*

The NVidia enhanced HPL benchmark written in C/CUDA that runs on both CPUs and GPUs was executed for different configurations to assess the scalability. As a preliminary step the problem size was varied in order to find the optimal value in regards to performance (N, the local problem size) as well as the block size (NB). Power consumption data were collected using xCat's rvitals utility reading blade-level power data every 5 seconds. Idle power consumption was measured during a 60 second time interval preceding every HPL run.

The results are shown in Figure 13. On the left graph the performance of the HPL benchmark running on four GPUs (two nodes) is shown for varying problem sizes. Details, including computational efficiency, are listed in Table 7. The right graph shows the scalability using weak scaling. Details are shown in Table 8. As is typical for this generation of GPUs, efficiency is limited to slightly above 50% due to architecture restrictions. The scaling is almost perfect; the efficiency over all runs varies only by 5%. Interestingly the intuitively optimal square numbers of GPUs cases exhibit slightly worse performance than non-square number of GPUs perhaps due to the fact that two GPUs share a node.

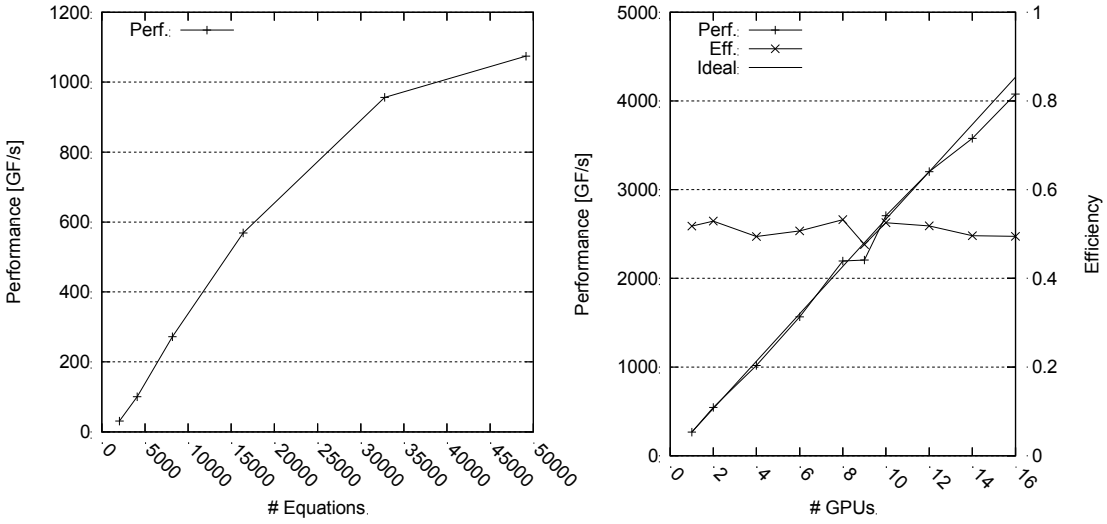


Figure 13 Performance of the HPL benchmark on the CaSToRC prototype.

GPUs	Size	Perf. [GF/s]	Efficiency
4	2 048	31	0.02
4	4 096	101	0.05
4	8 192	272	0.13
4	16 384	569	0.27
4	32 768	957	0.46
4	49 152	1074	0.52

Table 7 Performance of the HPL benchmark for varying sizes on 4 GPUs (2 nodes).

GPUs	Size	Perf. [GF/s]	Efficiency	Speedup
1	24 576	266	0.52	1.0
2	34 756	545	0.53	2.0
4	49 152	1018	0.49	3.8
6	60 199	1566	0.51	5.9
8	69 511	2194	0.53	8.2
9	73 728	2207	0.48	8.3
10	77 716	2707	0.53	10.1
12	85 134	3202	0.52	12.0
14	91 955	3578	0.50	13.4
16	98 304	4077	0.49	15.3

Table 8 Weak scaling performance for the HPL benchmark on the CaSToRC prototype.

2.2.5 *Conclusion*

The low sample rate combined with the suspected averaging of power values reported by the rvitals-based instrumentation caused large errors in the calculated energy-to-solution and energy efficiency values for the relatively short HPL benchmark runs. This could partly be compensated by repeated execution of matrix multiplication, which, in contrast to the HPL benchmark, does not require re-initialisation of the input data to re-execute the benchmark kernel. Through careful analysis of the results, not only was evidence of short term averaging uncovered, but also evidence of about a 10-minute warm-up period that increased the power consumption by about 50 W, or up to 10% was found. Using long-term averages after proper warm-up it was possible to establish a relatively exact energy efficiency value for matrix multiplication.

The results stress the importance of benchmarks which allow adjustable repetition of a kernel to create a constant workload at minutes to hour timescales to be able to reach steady state conditions and gather enough measured data to improve accuracy as well as the importance to find out how reported power readings are determined.

2.3 On die integrated CPU and GPU, PSNC

A new prototype using immersive liquid cooling was installed at PSNC and work was focused on commissioning and initial benchmarking of it.

Results for this prototype are reported for the first time in this deliverable.

2.3.1 *Hardware Description*

Servers

This PRACE-1IP 2nd phase prototype consists of 40 nodes in 46 modules. All nodes are equipped with three Gigabit Ethernet ports, one of which is dedicated for IPMI (Intelligent Platform Management Interface), a single QDR InfiniBand port and a 60 GB SSD drive. Each node has two Xeon E5-2620 (6 cores, 12 threads) processors clocked at 2.0 GHz in normal mode. The processors are able to reach 2.5 GHz in turbo core mode if only few cores are utilised and reach 2.3 GHz with all cores loaded. Of the 40 nodes 34 are considered "normal" and equipped with 32 GiB of ECC DRAM. Six nodes are acting as master nodes and have additional PCI extenders (16x PCI 2.0) that are used to connect GPU modules. These nodes have 64 GiB of memory. All modules are 100% liquid-cooled using an immersion technique, see Figure 14.

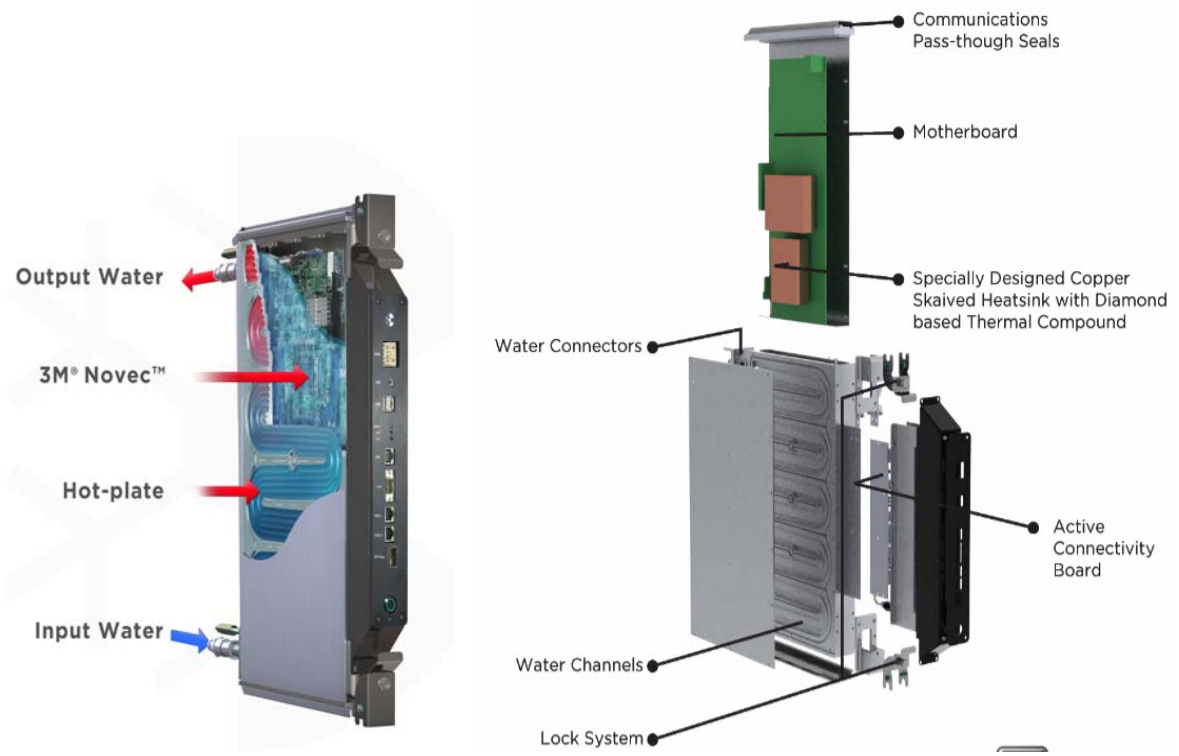


Figure 14 Illustration of the immersion cooled Iceotope modules of the PSNC-ICE prototype.

Rack

The rack is a custom solution designed by Iceotope to host their modules. Inside the rack an internal cooling loop, show in Figure 15, is installed. The loop is fully redundant – there are 2 pumps and 2 paths connecting the upper cold-water tank with each chassis. Eight nodes are installed in a chassis that has its own PSU. It is possible to have 2n redundancy on the PSUs.

The prototype has one water-cooled PSU per chassis.

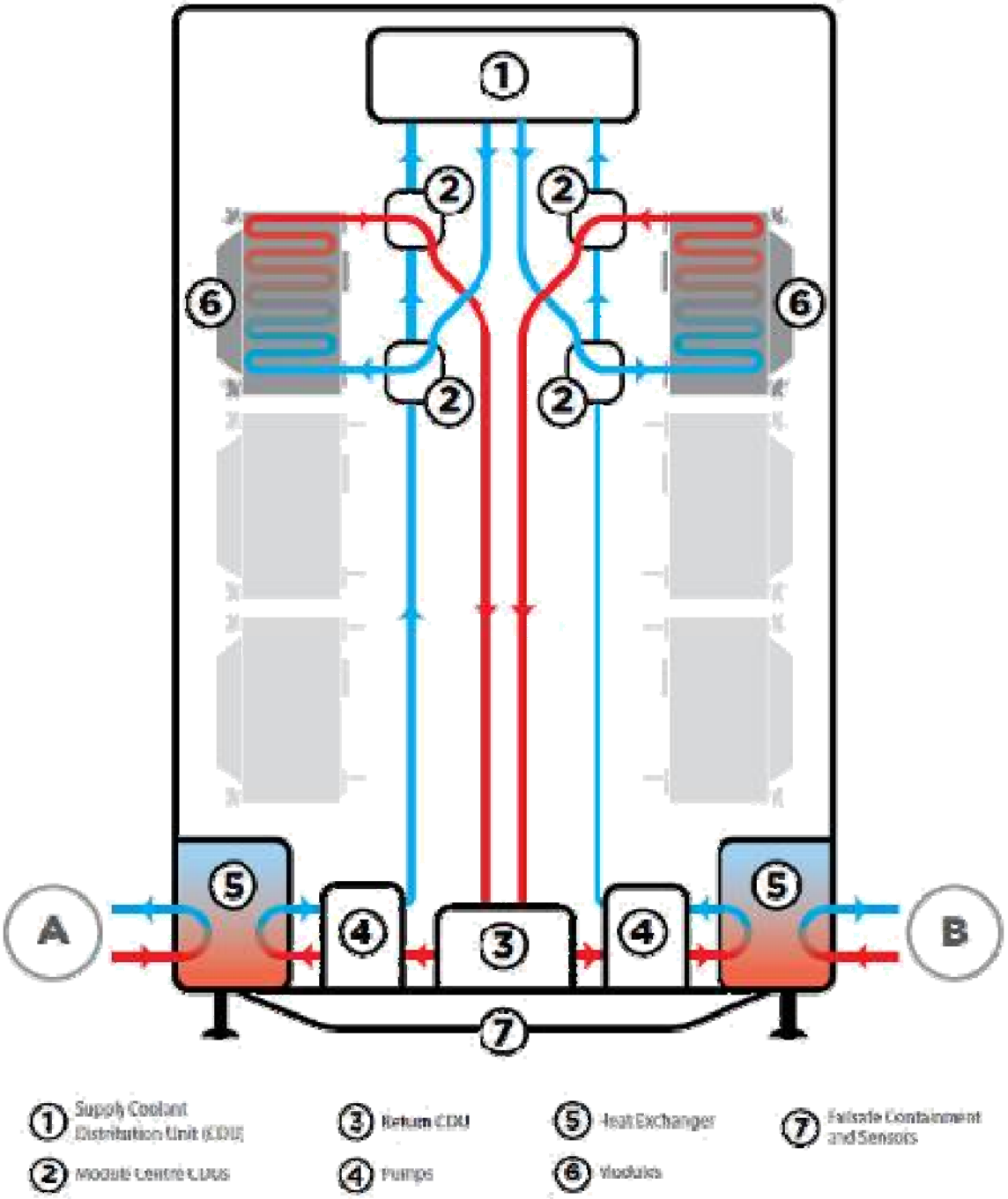


Figure 15 Schematic of the Iceotope rack water-cooling of the PSNC-ICE prototype.

Building Cooling Loop

The system is attached to a separate, custom cooling loop and, as emergency backup, the facility chilled water loop, see Figure 16. The cooling loop is designed for external temperatures in the range -30 – +50°C. Therefore the main heat transmission medium is a 30% glycol-water mixture. The loop control system can run in autonomous mode in which it tries to minimise the power consumption of the cooling loop or, as in our case, be managed manually by external software using the Modbus protocol. The loop is designed to handle up to 30 kW of power.

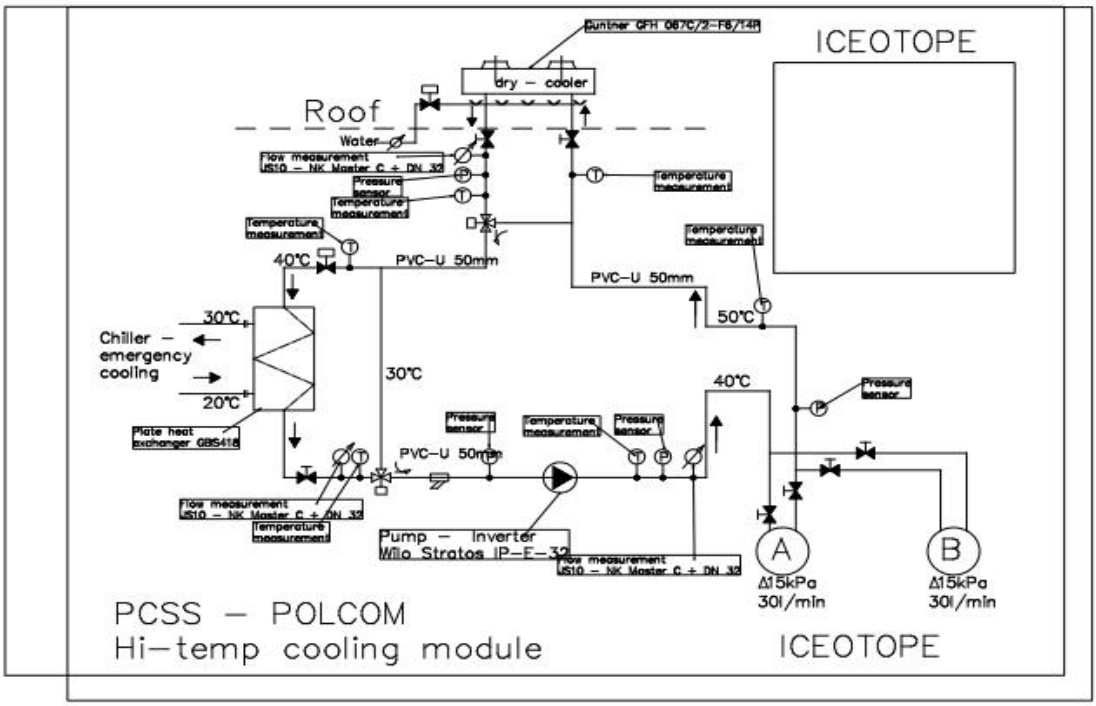


Figure 16 Schematic of the building water-cooling loop for the PSNC-ICE prototype.

2.3.2 *Power Measurement Instrumentation*

The power consumption was collected using two methods: 1) by the Avocent PM3000 PDU units in the rack, 2) by the Lumel P43 3-phase power network parameter transducer.

As the P43 device has the better power-measurement features of the two, all data used for the PRACE deliverables was collected using this transducer. Internally the device is gathering the data with a rate of 8000 samples per second using 24-bit signed converters (sigma-delta ADC converters). For the prototype measurements, however, the data was read from the device every 2 s. The sampling rate could not be increased because the data is obtained using RS-485 (using Modbus protocol) for which the maximum speed is 9600 b/s. In addition, according to the data sheet, the device's response time is 500 ms so a much higher sampling rate is not feasible. The device was set to report active power averaged for a 1 s (8000 samples) window.

The basic error for the measurement device is 0.5%. Since the total power is higher than the device can handle directly, a current transformer was used increasing the measurement error to 1%.

Figure 17 The Lumel P43 3-phase power meter used by the PSNC-ICE prototype.

2.3.3 *HPL Benchmark Results*

Due to stability problems with the InfiniBand interconnect, PSNC could only successfully complete a single large HPL run. The key results are shown in Table 9. Based on the nominal 6 144 GF/s theoretical peak performance for the 32 nodes, the HPL efficiency is 78%, which is good given the network related problems.

Size	Ranks	Time [s]	Power [W]	Energy [MJ]	Perf. [GF/s]	Eff. [GF/J]
314 832	384	4335	7415	32.15	4799	0.647

Table 9 Performance and energy efficiency of the HPL benchmark on the PSNC-ICE prototype.

2.3.4 *Conclusions*

Due to technical issues with the InfiniBand interconnect only very preliminary energy efficiency information could be obtained. Also, due to several issues with the integration of the immersive cooling technology into the PSNC data centre environment, energy recovery data could not be gather in time for this deliverable.

2.4 DSP based node for HPC, SNIC

Further work on the DSP-based node hardware and software provided two important improvements and significant detailed architectural insight, which have been fed back to our Texas Instruments collaborators and their SoC designers. First, the accuracy and robustness of the instrumentation system was improved. Second, a STREAM benchmark implementation utilizing the DMA capabilities of the DSP was developed that reaches a peak efficiency of 95.7% and offers about a factor three performance and energy-efficiency advantage over the cache-only version used in the previous report.

Earlier results for this prototype are reported in D9.3.3 [D933 p. 41ff].

2.4.1 *Node Instrumentation Update*

Although not directly visible in the schematic shown in Figure 18, the instrumentation infrastructure of the DSP based node was improved. In contrast to the earlier used serial RS-232 connection to the acquisition host, timing information from the DSP is now transmitted via a general purpose I/O (GPIO) line directly to the NI compactRIO data acquisition system (DAQ). This eliminates the timing uncertainty associated with the data transmission from the DAQ system to the acquisition host. Furthermore the timing resolution is improved since a single rising or falling edge is used to mark the start or end of the benchmark as opposed to two characters being sent via the RS-232 line.

The update required hardware extensions to connect the GPIO signal to the DAQ as well as software updates both in the DAQ, the DSP runtime system and the acquisition and analysis software to allow appropriate signals to be generated by the benchmark running in the DSP and subsequently to capture, correlate and analyse the resulting data stream offline.

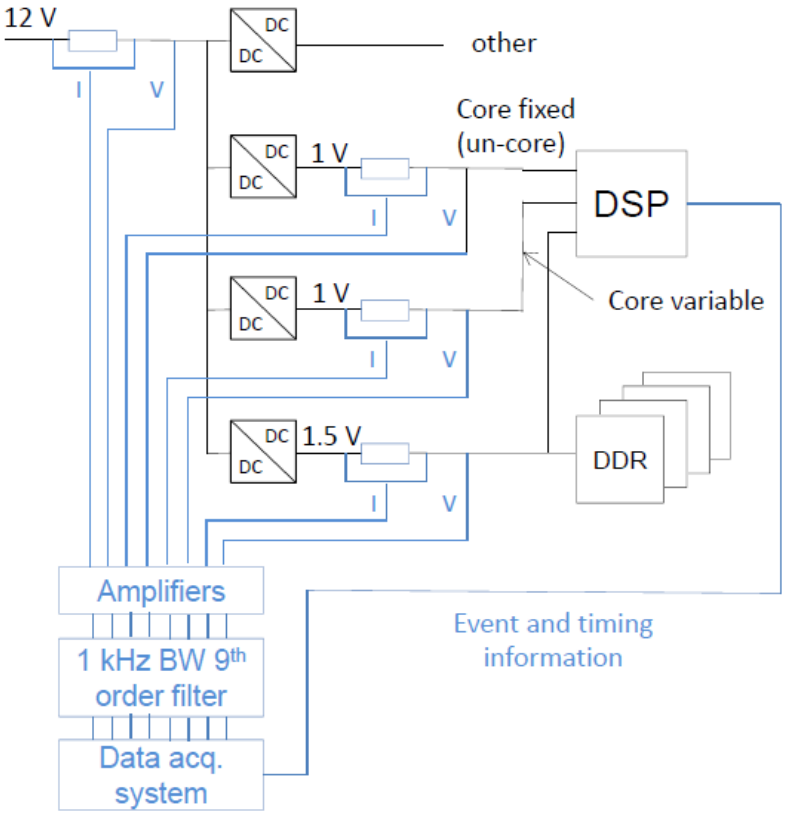


Figure 18 The power measurement instrumentation of the DSP EVM prototype.

2.4.2 *STREAM Benchmark Update*

Traditionally, the STREAM benchmark relies on caches to improve the performance of the memory hierarchy. This approach was also used by the STREAM benchmark used to obtain the results in the previous report (D9.3.3). While the DSP offers two levels of cache, the implementation suffers from two major drawbacks. First, as commonly the case for STREAM, the output vector has to be read into the cache before it is overwritten by new data, thus wasting between 25 – 33% of memory bandwidth. Second, the in-order execution model of the DSP core prevents overlapping of memory accesses with computation.

The DSP contains specialised DMA block copy engines (EDMA3) that allow both limitations to be overcome. That these hardware engines, in combination with double buffering, provide significant advantage was already demonstrated by the DSP HPL benchmark implementation. We adopted the basic technique to allow for a streaming multi-buffer DMA-based copy in and out framework suitable for the memory-bound long vector computations of the STREAM benchmark. The implementation allows the DMA unit to control the execution rate of the DSP cores via in-memory flags. Since the computation can be fully overlapped with memory accesses the DDR memory channel can be fully utilised (almost 96% of theoretical peak).

Another feature used by this benchmark is the ability to switch unused DSP cores into power down mode.

The DSP prototype has a single 64-bit memory channel and is equipped with 1333 MHz DDR3 giving it a peak memory bandwidth of 10.66 GB/s. Figure 19 shows the performance of the STREAM copy benchmark for varying sizes and thread counts. As expected, the odd thread counts exhibit relatively poor performance due to load imbalance between the two DMA engines used by the benchmark. The single thread performance also shows the bandwidth limitation (5.3 GB/s peak) between DMA engine and a single core.

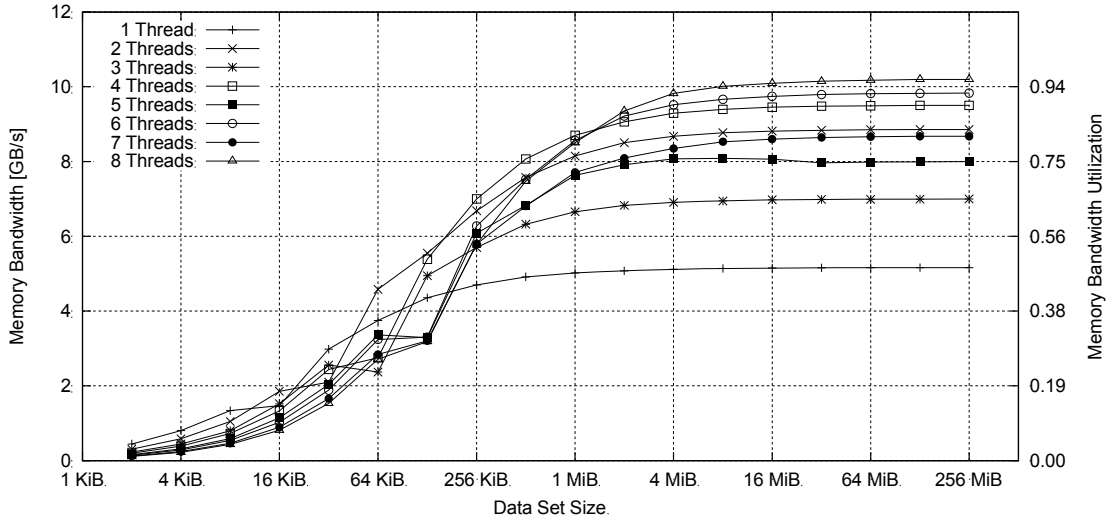


Figure 19 Bandwidth and Efficiency of the STREAM copy benchmark on the DSP node.

Figure 20 shows the energy efficiency obtained by the STREAM copy benchmark. Here the interesting feature is that even though 8 threads could achieve 15% better performance than two threads the latter configuration uses about 17% less energy as a result of switching off the unused cores. The effect reverses if the cores are not powered down but left in idle state.

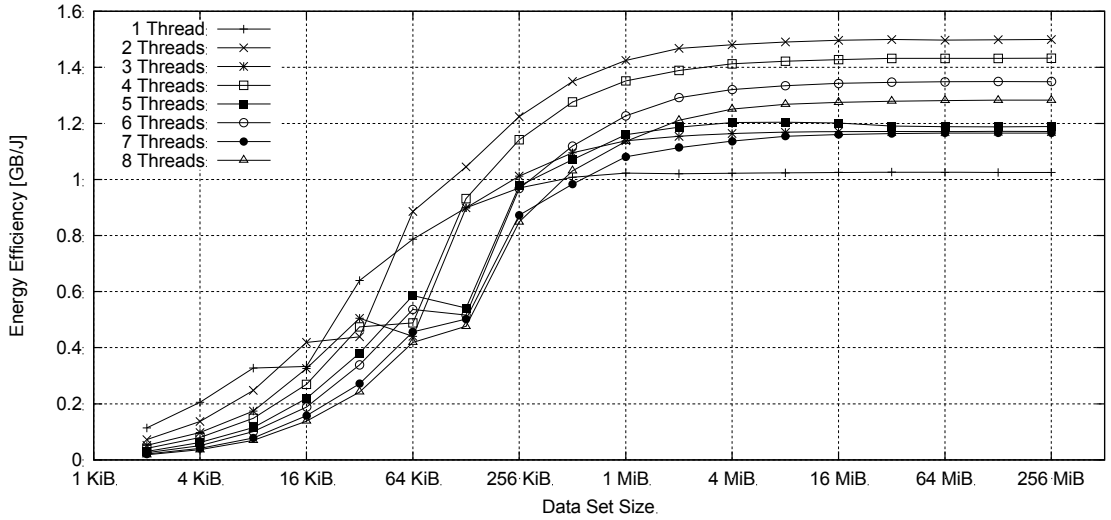


Figure 20 Energy efficiency for the STREAM copy benchmark on the DSP node.

Detailed results are listed in Table 10 for the largest data set sizes (256 MiB for copy and scale, 384 MiB for sum and triad) for all the STREAM operations and thread counts. Compared with previous results performance was increased by a factor of 3.3 for copy, 3.5 for scale, 3.4 for sum and 2.7 for triad. A corresponding increase of 2.6 – 3.3 times in energy efficiency was also achieved.

2.4.3 *Conclusion*

As already demonstrated with the HPL benchmark in D9.3.3 the DSP is able to reach near optimum performance also for the STREAM benchmark. This requires careful optimisation taking advantage of unique features of the platform and adjusting the benchmark implementation accordingly, something often already included in standard implementations for x86 platforms.

The results also show that performance increases were again, like for HPL, matched with corresponding increases in energy efficiency, indicating that proper software optimisation for performance should be a priority even for saving energy in a HPC context.

The most interesting result was that given an optimal implementation energy savings could be made by sacrificing performance by switching off cores.

Op.	Threads	Perf.	E.Eff.	Power [W]			
		[MB/s]	[MB/J]	Core	Uncore	Mem.	Other
Copy	1	5162	1025	2.8	0.4	1.8	11.8
	2	8860	1499	3.2	0.4	2.3	12.3
	3	6996	1171	3.5	0.4	2.1	12.3
	4	9505	1432	3.8	0.4	2.4	12.7
	5	7995	1189	4.1	0.4	2.2	12.8
	6	9830	1349	4.4	0.4	2.4	13.4
	7	8672	1166	4.7	0.5	2.3	13.4
	8	10196	1282	5.0	0.5	2.5	13.9
Scale	1	5161	1027	2.8	0.4	1.8	10.5
	2	8860	1502	3.2	0.4	2.3	10.9
	3	6996	1168	3.5	0.4	2.1	11.2
	4	9506	1428	3.8	0.4	2.4	11.3
	5	7995	1186	4.1	0.4	2.2	11.4
	6	9830	1345	4.4	0.4	2.4	11.8
	7	8676	1164	4.7	0.5	2.3	12.0
	8	10203	1279	5.0	0.5	2.5	12.4
Sum	1	5122	1023	2.8	0.4	1.8	11.3
	2	8420	1461	3.1	0.4	2.2	11.7
	3	6904	1171	3.4	0.4	2.0	11.8
	4	9107	1403	3.8	0.4	2.3	12.2
	5	7862	1183	4.0	0.4	2.2	12.3
	6	9587	1333	4.4	0.4	2.4	12.7
	7	8675	1167	4.7	0.4	2.3	12.9
	8	10046	1268	5.0	0.5	2.4	13.2
Triad	1	5122	1017	2.8	0.4	1.8	10.8
	2	8421	1446	3.2	0.4	2.2	11.4
	3	6904	1162	3.5	0.4	2.0	11.6
	4	9107	1389	3.8	0.4	2.3	12.0
	5	7862	1174	4.1	0.4	2.2	12.1
	6	9587	1324	4.4	0.4	2.4	12.5
	7	8675	1157	4.8	0.5	2.3	12.6
	8	10046	1260	5.1	0.5	2.4	12.6

Table 10 Performance, power and energy efficiency of the STREAM benchmark on the DSP.

3 Shared memory through a cache-coherency add-in card (NUMA-CIC), UiO

This prototype, described in detail in deliverable D9.3.3, uses NUMA-Scale Cache-coherent Inter-Connect (NUMA-CIC) add-in cards to realise a very large scale shared memory multiprocessor. The NUMA-CIC interfaces to a HyperTransport (HT) 3.1 channel and supports three rings with two 3.2 GB/s channels each for a total of 19.2 GB/s bandwidth to other NUMA-CIC cards, with a node by-pass delay of 53 ns. The HT 3.1 port has a maximum bi-directional bandwidth of 6.4 GB/s. The NUMA-CIC card has 8 GiB of cache memory and 4 GiB of tag memory. The prototype nodes have two AMD Magny-Cours CPUs each with four memory channels and 64 GB of 1333 MHz DDR3 memory. Nominally each CPU should have a bandwidth to memory of 42.7 GB/s but, due to Northbridge limitations, the maximum bandwidth per node is 28.8 GB/s.

Earlier results for this prototype are reported in D9.3.3. [D933 p. 44ff]

3.1 Benchmark Results

3.1.1 *STREAM Shared Memory (OpenMP) Benchmarks*

One of the most important benefits of the shared memory architecture is the potentially large memory. The benchmarks attempt to quantify the performance of the prototype using the shared memory and threads paradigms.

The results of using threads for the STREAM benchmarks are shown in Figure 21 for three different configurations of the copy benchmark. Further data is collated in Table 11.

The first case consisted of executing the benchmark using a single thread on one core. The other two cases were executed using 280 threads on the 70-node system. This configuration provided for one thread per memory controller. Two different NUMA data placement strategies, local and interleaved allocation, were used in the 280 threads benchmark runs. The single core runs used local allocation. In this context local allocation tries to find memory as close as possible to the core that first accessed the memory page in question, whereas interleaved allocation tries to spread the data amongst all available NUMA nodes in a round robin fashion.

Figure 21 shows achieved bandwidth (left) and bandwidth utilisation (right) for the STREAM copy benchmark. In the figure the single core bandwidth is scaled exactly 280 times (right axis) representing perfect scaling. This allows a comparison with the 280 cores results. The bandwidth utilisation is based on the aggregate local memory bandwidth to the cores executing the benchmark, which means all memory in the 280 cores cases and two memory channels in the one core case.

As can be seen, the single core benchmark performs quite well for vectors of up to one billion elements, after which performance drops by a factor of 40 – 50. The vectors of 20 billion elements require between 160 – 240 GiB of memory, which is bigger than the 64 GiB memory in a node so coherency traffic becomes a limiting factor. Spreading the calculation to 280 cores while keeping the data allocation local allows the system to keep all accesses restricted to the node local memories with the consequence that the largest vectors also give the highest bandwidth. In this case the about 30% bandwidth utilisation is comparable to typical (un-tuned) STREAM results.

The interleaved 280 cores case, that stresses the coherent memory system, shows about a factor 300 – 400 performance degradation for large vectors compared to the local only case,

showcasing the impact of data placement on performance. It is interesting to note that the aggregate performance (2.5 – 4.6 GB/s) stays below the bandwidth of a single HyperTransport link to a single NUMA-CIC card (6.4 GB/s).

Op.	Size	Bandwidth [GB/s]		
		Local		Interleaved
		1 Core	280 Cores	280 Cores
Copy	10 k	14.6	0.0	0.0
	100 k	5.5	0.1	0.1
	1 M	4.8	1.1	0.8
	10 M	8.9	10.5	5.5
	100 M		24.3	7.1
	1 G	7.8	155.5	5.1
	10 G		744.2	5.1
	20 G	0.2	1195.3	5.1
	60 G		1727.0	4.7
	100 G		1873.8	4.4
	180 G		1880.3	4.6
Scale	10 k	18.1	0.0	0.0
	100 k	6.7	0.1	0.1
	1 M	4.2	1.2	0.9
	10 M	6.2	10.2	7.6
	100 M		21.3	6.7
	1 G	4.4	112.2	2.8
	10 G		446.8	2.8
	20 G	0.1	560.0	2.7
	60 G		719.1	2.6
	100 G		721.8	2.5
	180 G		805.2	2.5
Sum	10 k	18.6	0.0	0.0
	100 k	9.4	0.2	0.2
	1 M	6.7	1.7	1.4
	10 M	6.5	16.4	8.4
	100 M		32.4	9.0
	1 G	5.5	156.8	2.9
	10 G		559.3	2.8
	20 G	0.1	678.5	2.8
	60 G		914.4	2.6
	100 G		935.6	2.6
	180 G		983.7	2.6
Triad	10 k	17.4	0.0	0.0
	100 k	9.5	0.2	0.2
	1 M	9.3	1.7	1.3
	10 M	6.6	15.1	9.7
	100 M		70.5	16.9
	1 G	5.5	234.1	2.9
	10 G		580.3	2.8
	20 G	0.1	738.9	2.8
	60 G		962.1	2.7
	100 G		940.6	2.6
	180 G		959.1	2.6

Table 11 Bandwidth obtained by the STREAM copy benchmark on the UiO prototype.

The other three STREAM kernels exhibit similar behaviour as show in Figure 22. Here the added latency of the more complex operations seem to cause about a factor 2 – 3 performance

degradation over the copy case for the 280 cores cases.

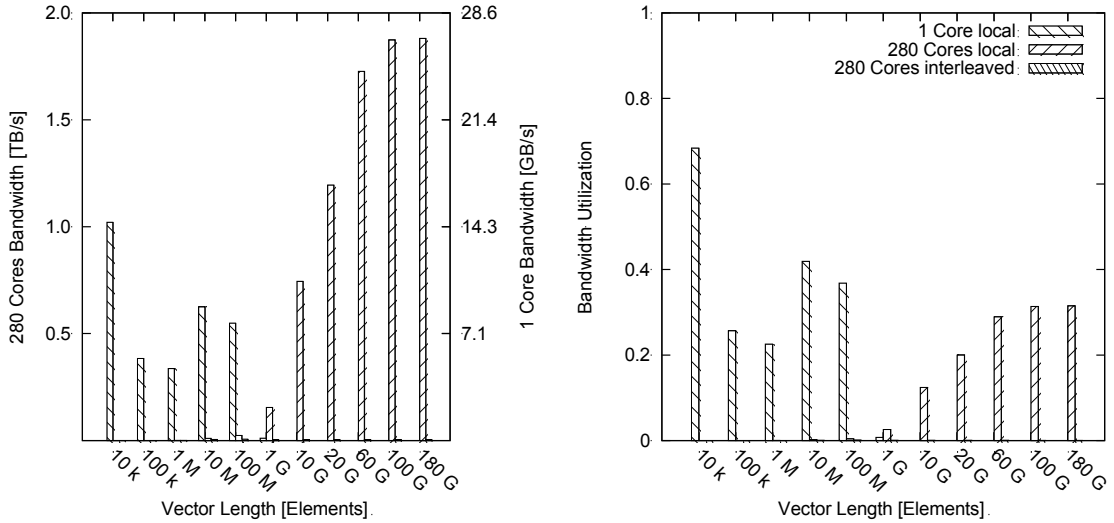


Figure 21 Performance and efficiency of the STREAM copy benchmark on the UiO prototype.

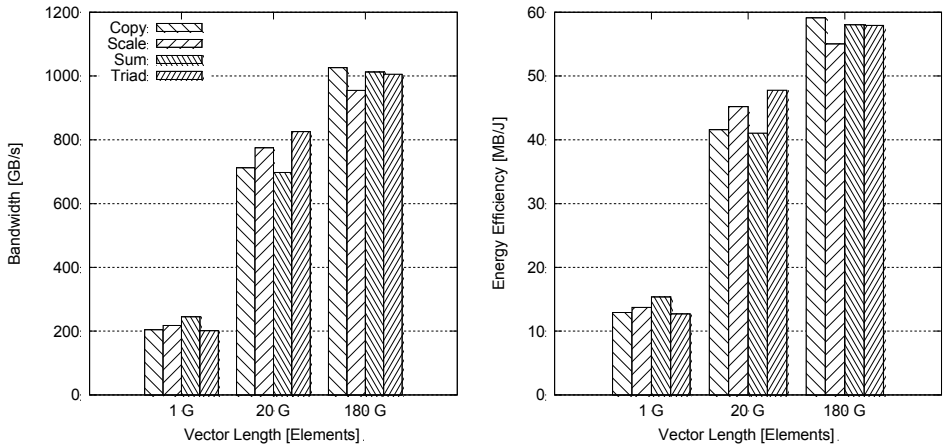


Figure 22 Bandwidth and energy efficiency for the STREAM benchmarks on the UiO prototype using 70 nodes.

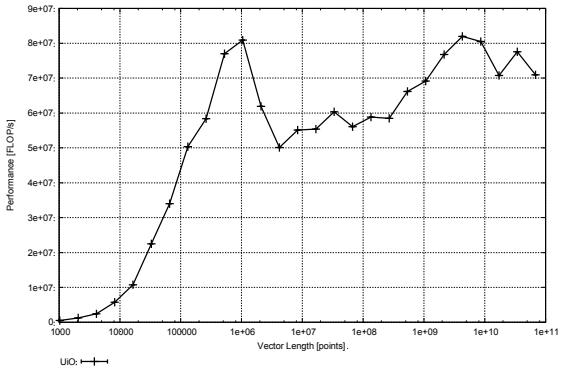


Figure 23 Performance of the Euroben FFT (mod2f) benchmark on the UiO prototype.

The results of the Euroben FFT (mod2f) benchmark showcase the impact of naive shared memory programming on an application level kernel. Performance for the 280-cores case (shown in Figure 23) stays below 90 MF/s, compared with the 11 GF/s measured on a single node (16 cores) of the LRZ system (with comparable node performance/CPUs). In this benchmark, the NUMA policy forced allocation of the data close to the single core that set up the test data, since test data generation is not threaded by default, causing a large amount of remote memory accesses.

3.1.2 *MPI Benchmarks*

NUMA-Scale provided an optimised OpenMPI byte transfer layer (BTL). Figure 24 and Figure 25 compare the performance of two selected MPI operations with the standard OpenMPI shared memory (sm) BTL. As expected, for point-to-point messages the NC-BTL improved both bandwidth and latency when the endpoints are inside a node (cores 180 to 181) compared to the default OpenMPI BTL (local). When going outside a node, bandwidth drops sharply by almost a factor of five while latency more than doubles. Separating nodes even further increases the latency again by 1.7 times, but keeps the bandwidth almost constant. Similar behaviour is reflected in the collective all-to-all operation where bandwidth for both the NC-BTL and the default implementation sharply drops when going outside a single node (> 4 cores). At 280 cores, each core can only transfer about 38 MB/s giving a total aggregate of about 10.6 GB/s, compared with the 19.2 GB/s of network bandwidth available on a single NUMA-CIC card.

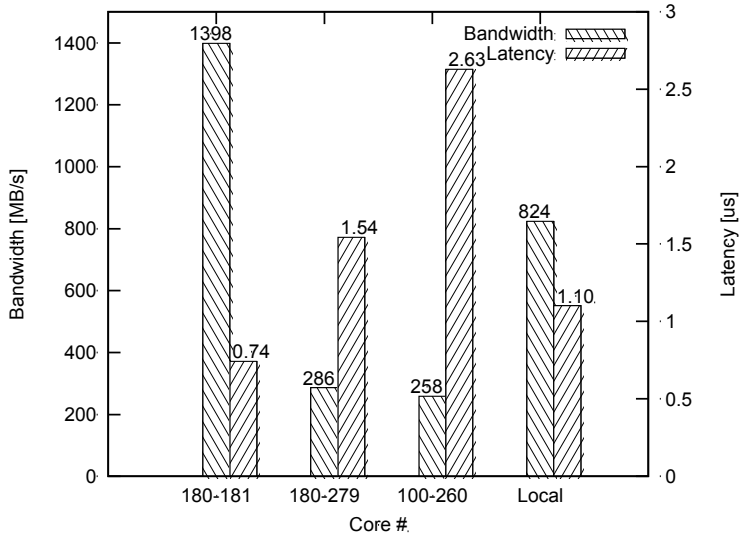


Figure 24 MPI Performance for point-to-point operations.

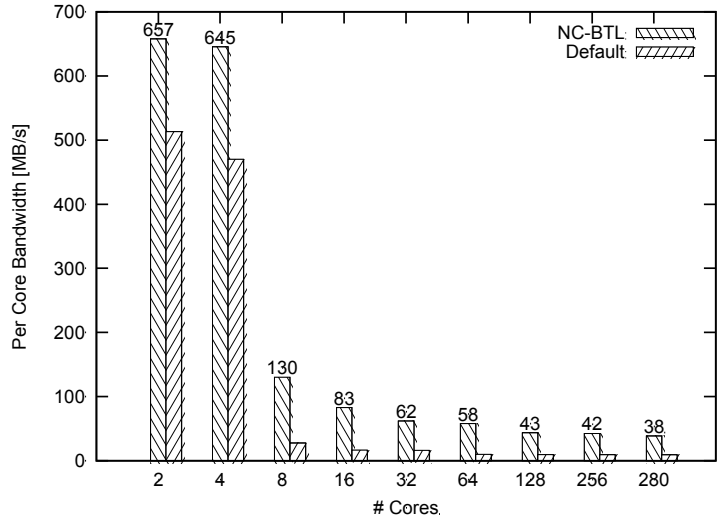


Figure 25 MPI performance for all-to-all collective operations.

Using the NC-BTL optimised OpenMPI library the HPL benchmark was executed on the 70-node system. The performance (left) and computational efficiency (right) are shown in Figure 26. The highest performance was achieved using a 32 by 40 process grid as opposed to using the full 1680 cores of the 70 nodes. The computational efficiency is shown based on the peak performance of the respective number of utilised cores and as such shows the potential for

performance improvement. The efficiency is rather poor compared to the customary >80% for HPL on x86 processors. In this context it is interesting to note that the LRZ prototype also using AMD Magny-Cours processors but a standard QDR InfiniBand interconnect also shows similar efficiency problems for the 16 and 32 node runs, indicating that further tuning of the benchmark may be beneficial.

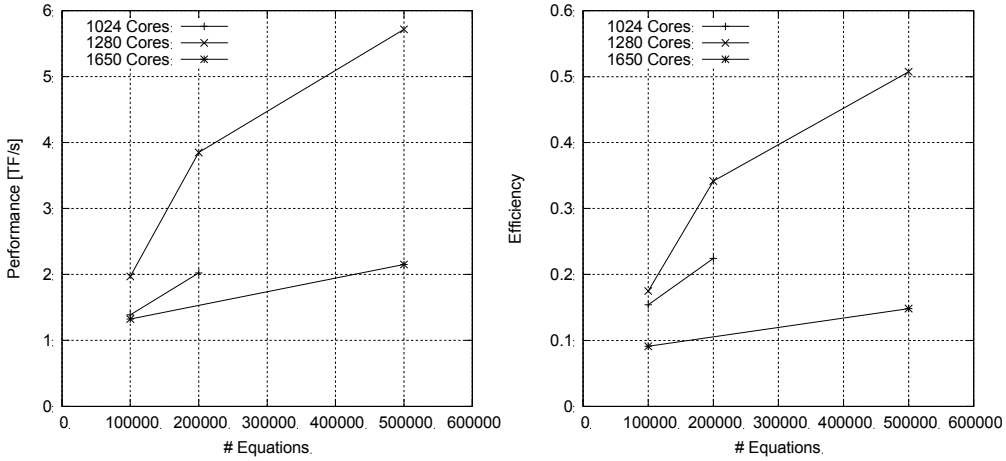


Figure 26 Performance (left) and efficiency (right) of the HPL benchmark on the UiO prototype.

The achieved energy efficiency is shown in Figure 27. Compared to both the LRZ prototype (220 – 330 MF/J) and systems with similar processors on the Green500 List (Rank 205, 400 MF/J, 177 TF/s) the UiO prototype meets expectations with 260 MF/J given the low computational efficiency.

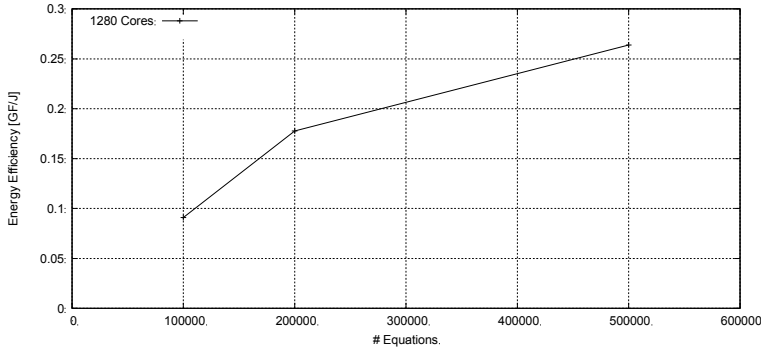


Figure 27 Energy efficiency of the HPL benchmark on the UiO prototype.

3.2 Experiences with the Prototype

During installation and bring up of the prototype system at the University of Oslo, numerous changes to hardware, software and configuration were necessary. This allowed university staff to gain unique experience with the issues and complexity encountered and gave them knowledge of the employed solutions. A partial summary of the valuable insights is presented below.

3.2.1 *Hardware*

The 2U standard servers making up the base platform for the nodes of the system are both easy to install and operate for data centre technicians. The additional NUMA-Scale hardware is more challenging and NUMA-Scale technicians handled two especially complicated tasks:

- Installing the special HyperTransport pickup module into an empty CPU socket on the motherboard and connecting it via a special flat-cable to the NUMA-CIC card.

- Wiring up the multi dimensional (up to 3) torus that connects the nodes.

3.2.2 *Software*

Both the kernel and surrounding software included in standard Linux distributions were not sufficiently tested on systems with more than 1600 cores. Numerous limits and problems with locks, semaphores and other kernel data structures were encountered and largely solved with great effort by NUMA-Scale. Some stability issues still remain but are being addressed by ongoing software releases. For core counts below 1024 the system can be considered as stable as any standard server-class Linux system, allowing operation for applications that have large memory footprints and poor scaling characteristics.

3.2.3 *Performance*

In terms of performance, the Achilles heel of the NUMA-Scale system is the cost of accessing off-node memory. Modern processors barely have enough out-of-order resources to hide the less than 100 ns latency of a local main memory access, let alone the 1 µs latency encountered when accessing off-node memory. Even traditionally compute-bound benchmarks like dense matrix multiplication can suffer severe penalties from processor stalls caused by suboptimal data placement. The presented performance results for the STREAM benchmark provide a vivid illustration of this effect causing up to a 400-fold slowdown for worst case (interleaved) memory placement.

3.2.4 *Ease of Programming NUMA Systems*

A common simplifying assumption made when programming shared memory systems is that main memory accesses have similar costs irrespective of the location of the data. This assumption leads, for instance, to the focus on work-sharing constructs in OpenMP that allow programmers to easily divide the execution of loops amongst several threads. Furthermore, exact work division and thread placement is, in typical scenarios, hidden from the application and delegated to compilers, libraries, run-time and operating systems. This strategy typically scales well within the cores of a single socket, but application performance quite often already degrades when scaling to all the cores of a multi-socket node.

Scaling applications beyond a single node requires the programmer to pay careful attention to the NUMA effects of the system, in particular the placement of data close to the cores that access it. Here the simple automatic approach outlined above can actually be harmful to performance as best illustrated by the FFT benchmark. The code of this benchmark is actually based on a message-passing algorithm that would make it simple to keep most memory accesses within one node. Due to the default automatic data placement based on first touch, it is however almost guaranteed that data used by a single core will be evenly spread across all nodes in the machine.

Fixing the problem would not only require manual memory and thread binding but also bypassing the (dynamic) OpenMP work-sharing mechanisms to guarantee that specific loop iterations are assigned to specific threads bound to cores close to the data. Under these assumptions most OpenMP mechanisms, like scheduling and thread affinity, need to be carefully controlled and become difficult to use in a simple and robust way.

3.2.5 *Developments tools*

All common development tools are able to support large memory applications requiring more

than 2 GiB of data in main memory. Furthermore all tools allow setting the default integer size to 64 bits (ilp64 model), providing a relatively straightforward scale up path for legacy applications that use, for instance, (32-bit) integer array indices and loop counts.

High core counts represent a bigger hurdle with several compilers and OpenMP implementations placing limits at 64 or 255 threads, inadequate for a 1680 core system. In this respect open source implementations offer the advantage of allowing straightforward increases to these limitations. For closed source software, close cooperation with the vendor is necessary.

3.2.6 *Threading libraries, NUMA control and binding*

As mentioned, binding threads and data object to specific cores and NUMA nodes of the system is essential for large-scale high-performance NUMA applications. Fortunately the Linux kernel already offers methods to accomplish these bindings. Unsurprisingly, however, runtime libraries and compilers struggle to use these kernel interfaces efficiently. Moreover most algorithms and strategies implemented in threading libraries have not been extensively used at core counts of about 1600, and are therefore often inefficient.

System provided utilities like numactl allowing the setting of process global policies are a Band-Aid that can sometimes improve the situation but do not offer the fine-grained control required for highly efficient operation.

3.2.7 *Message Passing Interface, MPI*

It is relatively easy to implement message passing on top of a shared memory model and high quality MPI implementations frequently use shared memory for efficient intra-node communication.

Such commonplace implementations however, do not perform well when used at inter-node scales. NUMAScale has therefore implemented an improved shared memory device for use with the OpenMPI library. This implementation makes careful use of non-temporal stores to bypass the cache coherency mechanisms. Interestingly this implementation could also increase performance for the intra-node cases.

4 STREAM Benchmark Results Summary

With the updated and extended data from the STREAM benchmarks on the BSC-2, UiO and SNIC prototypes, it is interesting to relate the new results to the findings presented in D9.3.3 for the same benchmark on the earlier BSC-1, CaSToRC and LRZ prototypes. Since the scales differ widely only normalised quantities like utilisation of available memory bandwidth or energy efficiency can be compared. Even for the normalised results, it is important to appreciate the different scales of the prototypes.

Figure 28 shows how well the STREAM benchmarks could use the available theoretical memory bandwidth. For each prototype, data for the largest available vector length was used, which generally also forces the use of main memory. For the new BSC-2, SNIC and UiO results the number of threads is shown. For the SNIC prototype the best performing and highest energy efficiency cases are shown. All prototypes except the UiO machine used a single node for the benchmarks; in the UiO case 70 nodes in a NUMA configuration were used.

Figure 29 shows the energy efficiency for the same experiments as in Figure 28. Here, it is interesting to see that the SNIC prototype can clearly benefit from the highly efficient benchmark implementation. For the BSC-2 prototype, the burden of the large idle accelerator creates a high overhead and quite unsurprisingly causes very low energy efficiency when using the CPU in this configuration for copying data.

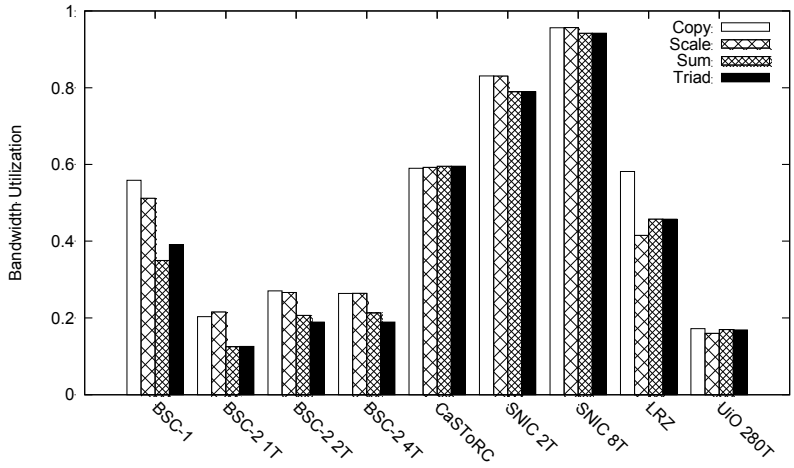


Figure 28 Bandwidth utilisation for the STREAM benchmark across prototypes.

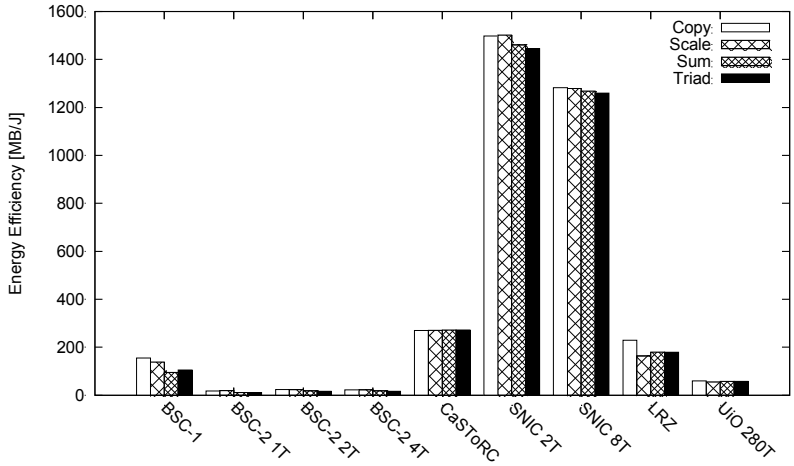


Figure 29 Energy efficiency obtained by the STREAM benchmark across prototypes.

5 Holistic Approach to Energy Efficiency, LRZ

One of the main goals of the LRZ PRACE-1IP WP9 prototype was to evaluate technologies and approaches to achieve high energy-efficiency for multi-petaflops HPC systems through energy recovery. Much of the design, implementation and results were covered in D9.3.3. Here we report the work conducted and results obtained since then.

Earlier results from this prototype were reported in D9.3.3. [D933 p. 48ff]

5.1 Background

With direct hot-water cooling using inlet temperatures of about 40°C, outlet temperatures reach a level for which it is becoming more practical to re-use the heat generated by HPC systems, as well as using return water from energy re-use as inlet for HPC system cooling. This re-use will not make data centres more energy efficient but it can help to reduce overall energy consumption and data centre costs.

Currently air is the most commonly used cooling medium and the outlet air in most data centres is not sufficiently hot for re-use and so the energy is typically dissipated into the environment. One possible exception is during winter in cold climates, where an air-to-air heat exchanger could be used to warm up the air in a forced-air heating system for offices or laboratories. Whether the corresponding savings can offset the necessary infrastructure expenses is unclear and needs to be evaluated on a case-by-case basis.

Liquid-cooled systems are becoming more common due to increased heat densities and the considerably higher heat capacity of liquid coolants. Several technologies for direct liquid cooling have been introduced in recent years. Liquid-cooled systems offer more possibilities for cost effective energy re-use than air-cooled systems. Underfloor heating systems, which typically do not require very high water temperatures, as well as forced-air heating systems, could be driven by the coolant of the computer if the return temperature is on the order of 30-40 °C. Heating systems based on radiators require much higher temperatures. Here the cooling system needs to support return temperatures of at least 65°C. If this can be achieved, there is yet another possibility for energy re-use: the cooling using adsorption chillers (see section 5.4). There are adsorption chillers on the market (e.g., by InvenSor, SorTech and others) that operate efficiently at hot-water inlet temperatures of about 65°C. This is particularly interesting in summer when heating is generally not needed in most locations and demand for chilled water is at its peak.

In assessing the economics of energy re-use, the cost for the additional infrastructure needs to be balanced against the savings that can be obtained from energy re-use to ensure an acceptable return on investment (ROI), as well as the environmental impact of alternate solutions. Hot-water-cooling necessitates insulating the computing equipment and the coolant pipes against heat convection to prevent the heat from escaping into the air of the data centre where it would have to be removed by an air-conditioning system at additional expense. The insulation against heat dissipation adds cost in addition to the cost of the energy re-use infrastructure. It is also necessary to assess the impact of higher operating temperature on power consumption (some results reported in D9.3.3), possibly performance reduction, component failure and associated costs.

5.2 Prototype: Basic Description

The LRZ CooLMUC prototype, see Figure 30, described in D9.3.3 consists of five racks with the 178 compute nodes and their InfiniBand interconnection network contained in three racks (labelled “Compute racks”) and the cooling components contained in the other two (labelled “Cooling racks”). A SorTech ACS-08 adsorption chiller (labelled “Adsorption chiller”) is driven by the hot exhaust water from the cluster and cools the water in a secondary loop. The chilled water is then used to cool the hot exhaust air of a sixth rack (labelled “Extra compute rack”) via a rear door heat exchanger. Figure 31 shows a schematic of the internal CooLMUC cooling infrastructure.

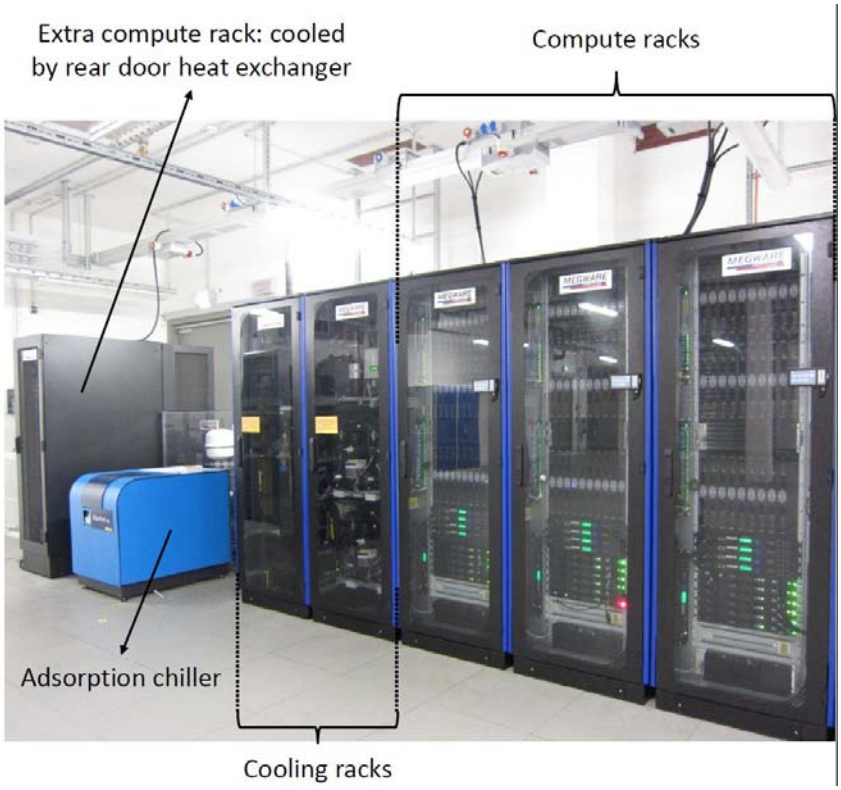


Figure 30 The CooLMUC experimentation cluster at LRZ.

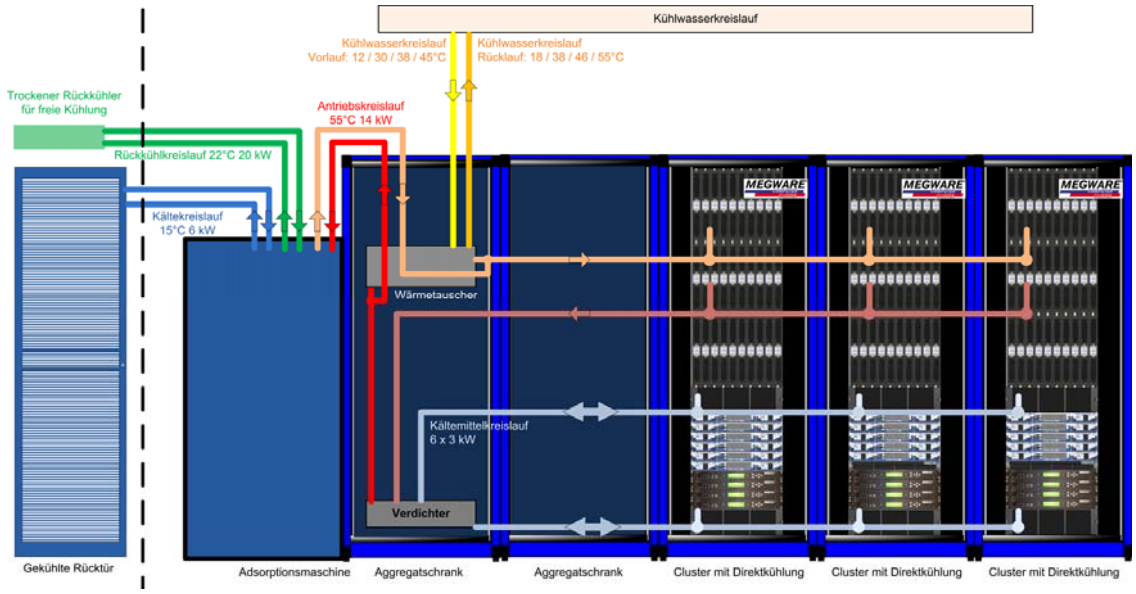


Figure 31 Schematic of the CooLMUC cooling loops.

To recover as much as possible of the generated heat, the CooLMUC system was designed to not require external cooling systems, and thus not depend on LRZ's CRAC infrastructure. Two independent cooling loops are used to cool the compute equipment. One loop provides water at 40°C directly to the nodes, where it flows through copper pipes connecting heat sinks on top of CPUs, chipset, and InfiniBand HCAs. At this time, some components remain air-cooled, for instance the compute node power supply units, the InfiniBand switches, and the rack power distribution units. The second cooling loop for the air cooled components is based on standard compressor-based cooling technology with special 19" in-rack evaporators that push air from the rear part of the racks to the front while cooling it to the set temperature of 30°C. In order to use the heat collected from both cooling circuits to drive the adsorption chiller, the condenser of the second cooling circuit is cooled with water originating from the first cooling loop's outlet. This way, water inlet temperatures to the server of 40°C generate outlet temperatures of around 60°C. This outlet temperature is sufficient for powering an adsorption chiller (illustrated in Figure 32) that is quite compact in comparison to a traditional chiller. However, it does need to be connected to an external pumping unit.

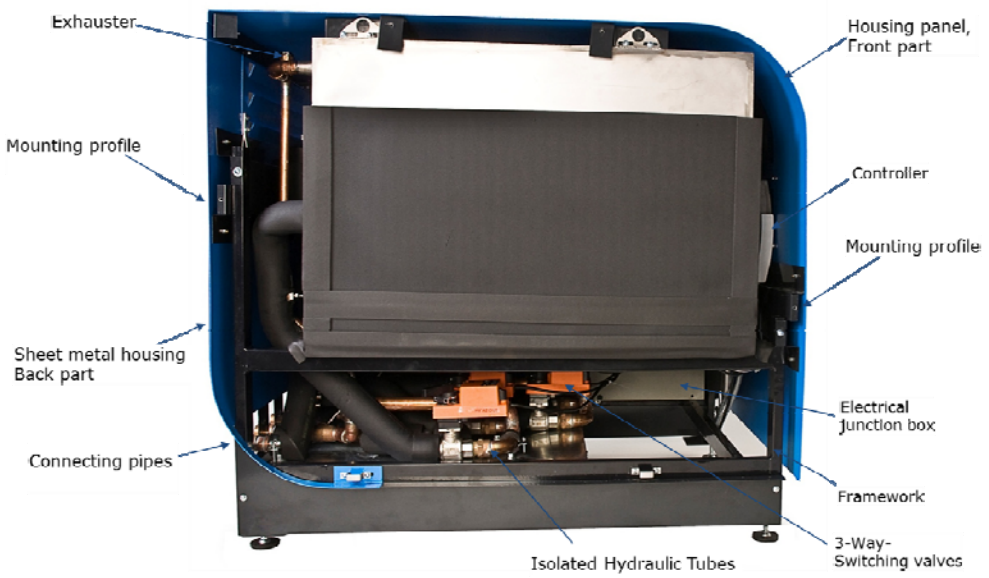


Figure 32 Internal view of the SorTech ACS-08 adsorption chiller.

5.3 Internal Infrastructure changes for enhanced monitoring and assessment

CooLMUC features thorough monitoring for power consumption and temperatures in the entire cooling apparatus, as described in D9.3.3. But, in assessing the efficiency and benefits of CoolMUC, it became clear that the original quite extensive instrumentation was not sufficient. For instance, when increased CPU temperatures were observed, no sensors existed to provide data required for the analysis of the cause. Therefore, a decision was made to augment the system with flow meters. Additionally, to be able to evaluate the efficiency of the adsorption chiller, the supply and cooling loop needed to be instrumented as well.

A total of six Taconova TacoSetter Tronic[TACONOVA] ultrasonic flow meters were added to CooLMUC. In addition to the flow rate, the TacoSetter Tronic devices can also measure water temperature. This, in combination with a single temperature sensor on the other side of the cooling loop, allows for the calculation of the heat quantity being transferred to or from the water in a given water-cooling loop.

The TacoSetter Tronic flow meters output an analogue measurement signal (0.5 – 3.5 V) that cannot be directly processed by the CooLMUC monitoring system. An additional measurement circuit was developed to sample the analogue signals of flow rates and temperatures and make them available to the integrated cluster monitoring solution of CooLMUC using an RS-232 interface.

To calculate the instantaneous heat power transferred via the water (Q), the flow rate and the difference between the input and output temperatures of the cooling loop are required. The following formula is used:

$Q = m\ c_w\ (T_O - T_I)$, where

- m: Flow rate (l/s)
- c_w: Heat capacity of water
- T_I: Inlet (cold-water) temperature (°C)
- T_O: Outlet (hot-water) temperature (°C)

Both the warm-water driving circuit of the adsorption chiller and its cold-water circuit have been augmented with the new sensors allowing for measuring the amount of heat extracted from the cold water loop by the adsorption chiller and the corresponding amount of driving energy taken from the cluster's hot water. These measurements allow for an analysis of the efficiency of the adsorption chiller by calculating the coefficient of performance (COP) discussed in section 5.6.2. Unfortunately, funding for the enhanced instrumentation, procurement, contracting, and integration of the new sensors into the monitoring software took more time than expected with the result that data is only available since the end of October 2013.

5.4 Adsorption

According to the New World Encyclopaedia [NWE]: "Adsorption, not to be confused with absorption, is a process by which a gas, liquid, or solute (substance in solution) binds to the surface of a solid or liquid (called the adsorbent), forming a film of molecules or atoms (called the adsorbate). It differs from absorption, a process by which a substance diffuses into (or permeates) the solid or liquid absorbing medium. The term sorption encompasses both processes, and desorption is the reverse of either of the two processes."

An adsorption chiller uses heat to cool water in a secondary circuit, which in turn can be used for cooling or refrigeration applications. It cools by evaporation of water and binding the vapour on porous solids. The adsorption agent for this technology is mainly silica gel. The process of cooling is based on two constantly recurring steps [SORTECH], illustrated in Figure 33:

Step 1: Adsorption – Accretion of water vapour on the adsorbent's surface

- Warm water located in the condenser is fed into the evaporator. Heat in the water is extracted through the evaporation process cooling the water in the coolant loop. Water vapour formed in the evaporation process is adsorbed by dry silica gel (adsorbent). Once the adsorbent is saturated the regeneration starts.

Step 2: Desorption – Drying the adsorbent (regeneration)

- Warm water in the driving loop is channelled over a heat exchanger through the adsorption chamber in the chiller aggregate. The added heat dries the silica gel, resulting in a discharge of water vapour that flows into the condenser where it condenses again and releases the collected heat from both the coolant and driving circuit to the re-cooling circuit. Once the silica gel is sufficiently dry the addition of heat in the adsorber chamber stops.

Continuous cooling requires operation of two adsorbers counter-cyclically (i.e. during desorption of one adsorber the second one adsorbs and cools and vice versa). Condenser and evaporator form a closed circuit within the adsorption chiller.

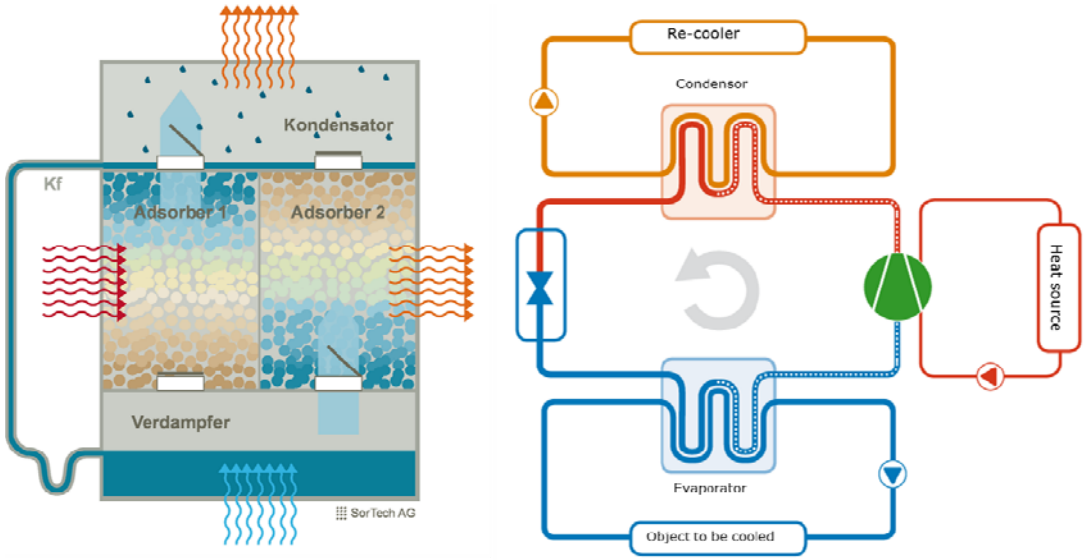


Figure 33 Schematic of the adsorption chiller.

In the CooLMUC case, the adsorption chiller uses the heat collected by the hot water cooling system to chill water that is used to cool the additional rack via a rear door heat exchanger.

5.5 Measurement Setup Adsorption

A schematic overview of the prototype's adsorption measurement points can be seen in Figure 34, details on the accuracy are given in Table 12.

- Sensors labelled 1 measure the flow rate and temperature of water loops.
- Sensors labelled 2 measure the temperature of the water.
- Sensor labelled 3 measures the power consumption of the adsorption chiller.

Number	Device	Scope/Purpose	Measurement	Error	Sampling
1	TacoSetter Tronic	Measurement of flow rate and temperature	1 – 12 l/ min	< 3□%	1/min
			2 – 40 l/min	± 1,5□%	
			Temperature range	0 – 100□°C	
2	Water Temperature Sensor	Measurement of water temperature	Temperature range	[3]	1/min
3	Megware Clustsafe PDU	Monitoring of electrical power consumption of the pumping group of the adsorption chiller	Power	[3]	1/min

Table 12 CooLMUC adsorption chiller sensors details, see also Figure 34.

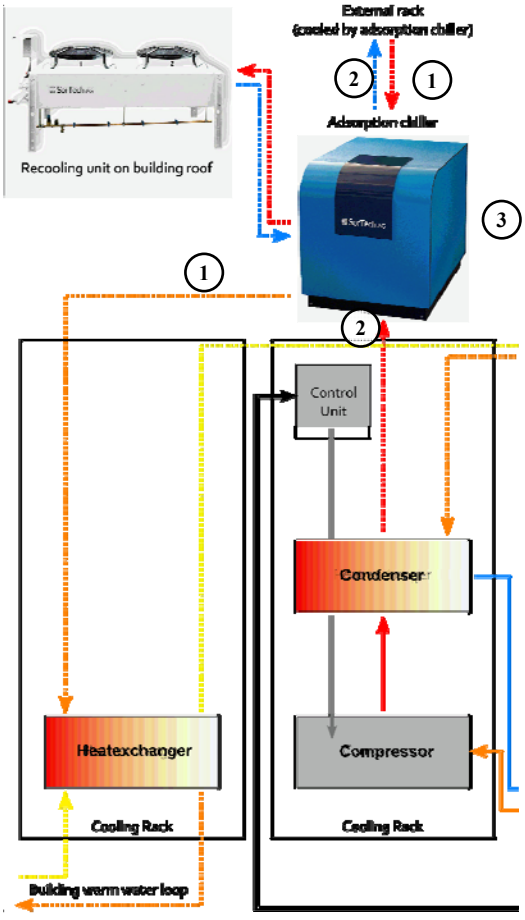


Figure 34 CooLMUC adsorption measurement points, see also Table 12.

[3] Sensor accuracy is still to be checked with device vendor.

5.6 Measurement Results

5.6.1 *Power consumption vs. cooling water inlet temperature*

Semiconductors are known to have increased leakage currents at high operation temperatures, effectively resulting in an increase of the power consumption for operating the logic circuitry. As reported in D9.3.3, the increase of the CoolMUC power consumption is only about 0.133 kW/K, or 0.322 %/K in relation to the power consumption at 27°C. At 50°C the increase is 7.4% (see Figure 35).

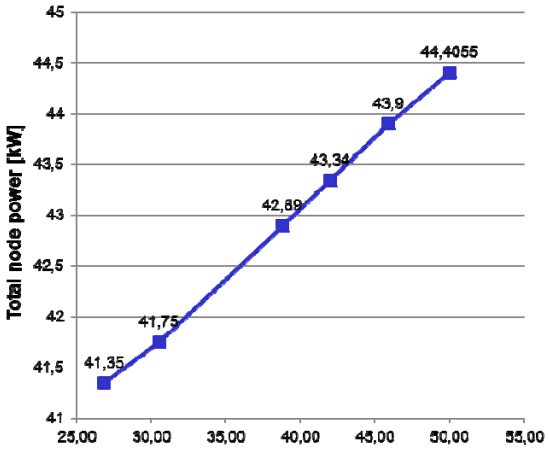


Figure 35 CooLMUC node power consumption under max load in relation to water inlet temperatures.

Unfortunately, because of the late availability and integration of the needed new sensors, there was not enough time to experimentally evaluate the key performance indicators (KPIs), which are Coefficient of Performance (COP), Power Usage Efficiency (PUE), and Energy Reuse Efficiency (ERE), for every possible CooLMUC water inlet temperature. Therefore, all evaluations were done using a CooLMUC water inlet temperature of 40 °C. This water temperature can be generated by the current LRZ facility hot-water-cooling infrastructure all year around without the use of any active chillers.

5.6.2 *Adsorption chiller COP*

The COP of an adsorption chiller is a measure of its efficiency. For example, a COP of 0.5 means that the energy extracted from the cooling circuit is half of the provided energy, mostly from hot water. The COP is calculated as:

$$\text{COP} = \frac{Cold\_Water\_Power}{Hot\_Water\_Power}$$

The Cold_Water_Power for timestamp "i" is calculated using:

$$Cold\_Water\_Power_i = T1_{Flowi}\, c_w\, (T1_{Tempi} - T2_{Tempi})$$

Where:

- c_w: Heat capacity of water.
- $T1_{Flow}$ is the flow rate of the cooling water loop as measured by the TacoSetter Tronic from number 1 adsorption chiller to the "external rack".

- $T1_{Temp}$ is the inlet water temperature of the cooling water loop as measured by the TacoSetter Tronic from number 1 adsorption chiller to the "external rack".
- $T2_{Temp}$ is the outlet temperature of the cold water generated by the adsorption chiller as measured by the water temperature sensor number 2 adsorption chiller to the "external rack".

The Hot_Water_Power for timestamp "i" is calculated as:

$$Hot\_Water\_Power_i = T1_{Flowi}\, c_w\, (T2_{Tempi} - T1_{Tempi}) + AdsorptionElectricPower$$

Where:

- c_w: Heat capacity of water.
- $T1_{Flow}$ is the flow rate of the hot water circuit used to power the adsorption chiller measured by the TacoSetter Tronic number 1 from adsorption chiller to the heat exchanger of the cooling rack.
- $T1_{Temp}$ is the adsorption chiller hot water outlet temperature from adsorption chiller to the heat exchanger of the cooling rack used to power the adsorption chiller measured by the TacoSetter Tronic number 1.
- $T2_{Temp}$ is the adsorption chiller hot water inlet temperature from condenser of the cooling rack to adsorption chiller used to power the adsorption chiller as measured by the water temperature sensor number 2.
- *AdsorptionElectricalPower* is the power measured by the internal power distribution unit (PDU) of CooLMUC to which the adsorption chiller pumping group (number 3) is connected.

The COP depends strongly on outside conditions (air temperature for dry cooling units, and wet bulb temperature for hybrid and evaporative coolers). The COP decreases with higher outside temperatures (and increases with lower outside temperatures). Additionally, the water temperature of the hot-water driving circuit also has a strong influence on the COP for an adsorption chiller. Lower temperatures of the hot-water driving circuit will decrease the COP and higher will improve it. Therefore, the best evaluation method would be to average all COP measurements over one year. For this deliverable all available measurements were used.

Figure 36 shows a plot of the COP from Oct 31st till Nov 14th in combination with water inlet temperatures for the driving hot water circuit of the adsorption chiller, and the outside air temperature. Since the COP constantly changes, depending on where in the adsorption cycle the machine is, the measurement data was averaged. The average COP was 0.52 for the timeframe. The average inlet temperature was 58.8°C and the average outside air temperature was 9.4°C.

The manufacturer data sheet for the adsorption chiller is shown in Figure 37. The adsorption chiller was running in Power Mode. As the data sheet shows, for this mode a maximum COP of 0.6 can be achieved with an inlet temperature of 65°C. Additionally, the COP flattens with re-cooling water temperatures (T_MT_IN) below 25°C. Under the conditions shown in Figure 36 an average COP of 0.52 is very good for our lower inlet temperatures of 58.8°C.

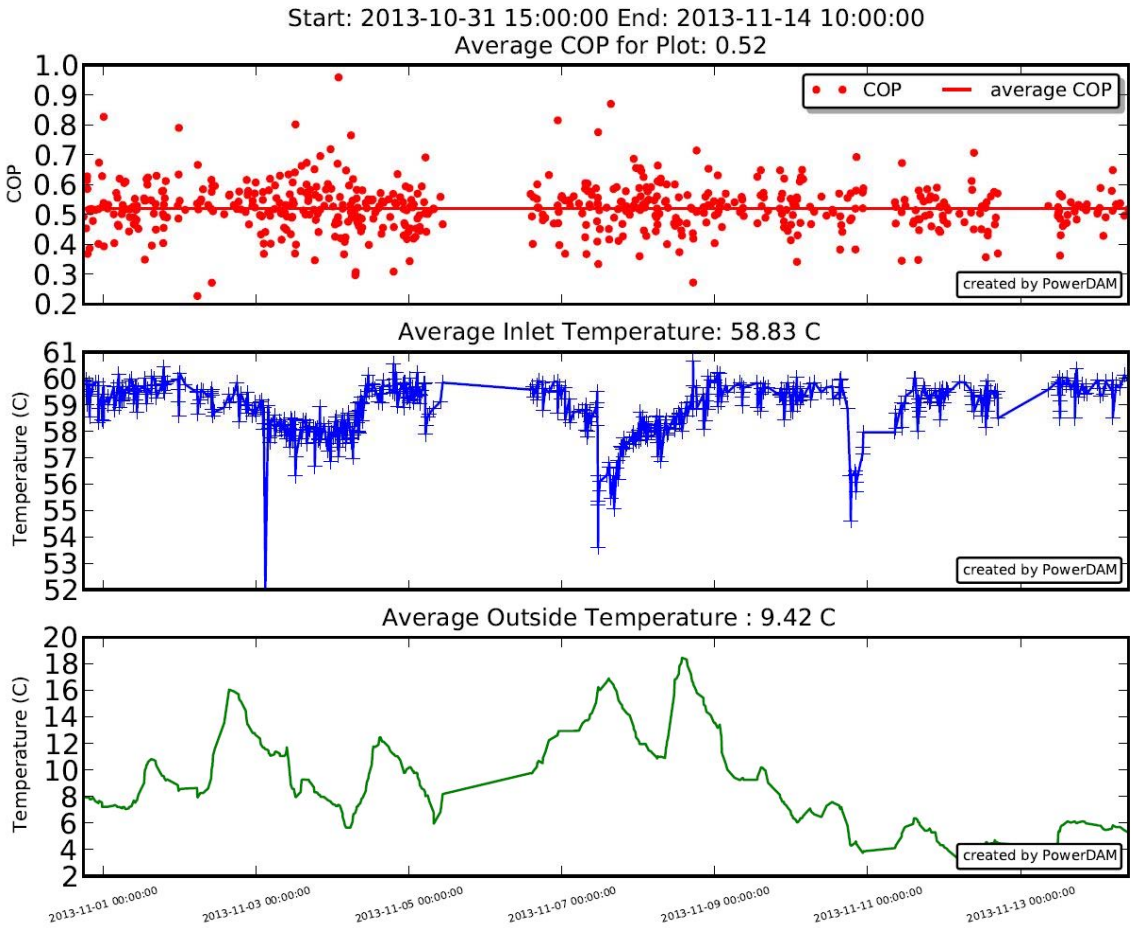


Figure 36 Adsorption chiller Coefficient of Performance (COP), Average inlet temperature, and Average outside air temperature plot.

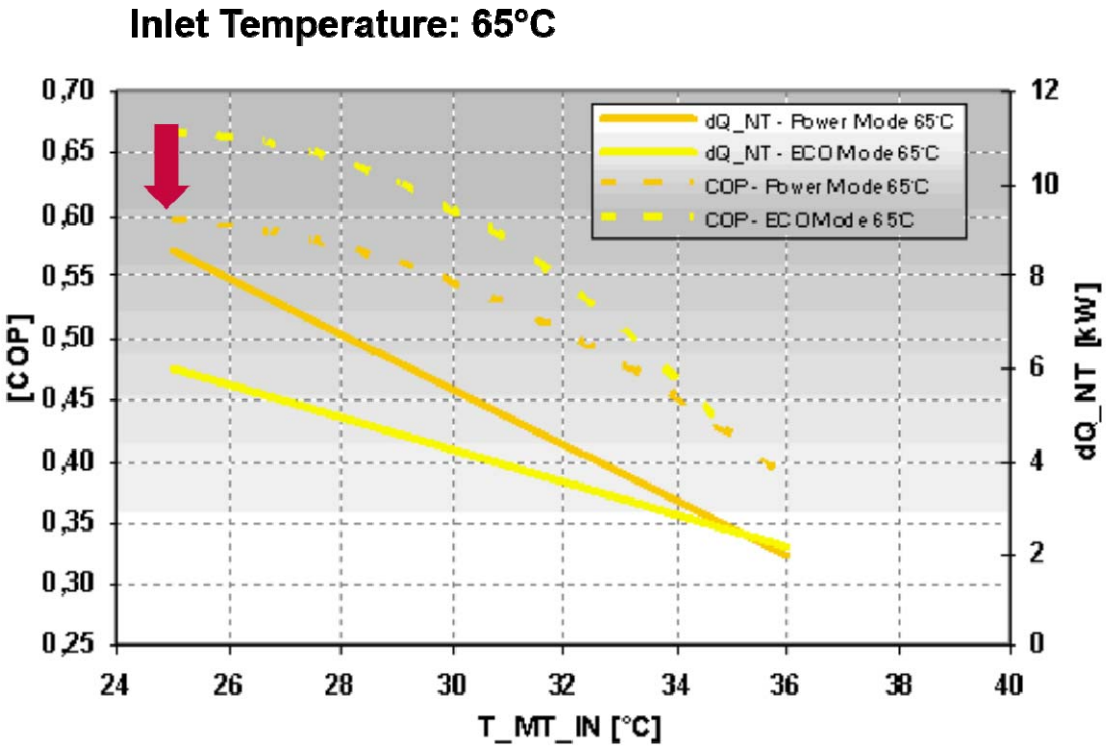


Figure 37 SorTech Adsoprtion Chiller Data Sheet.

The energy used by the additional rack (labelled Rack 5), and the heat removed by the adsorption chiller cold-water loop are shown in Figure 38. The rack produced an average of 6 kW of heat of which around 4.5 kW were removed by the adsorption chiller. The remaining 1.5 kW were released into the room.

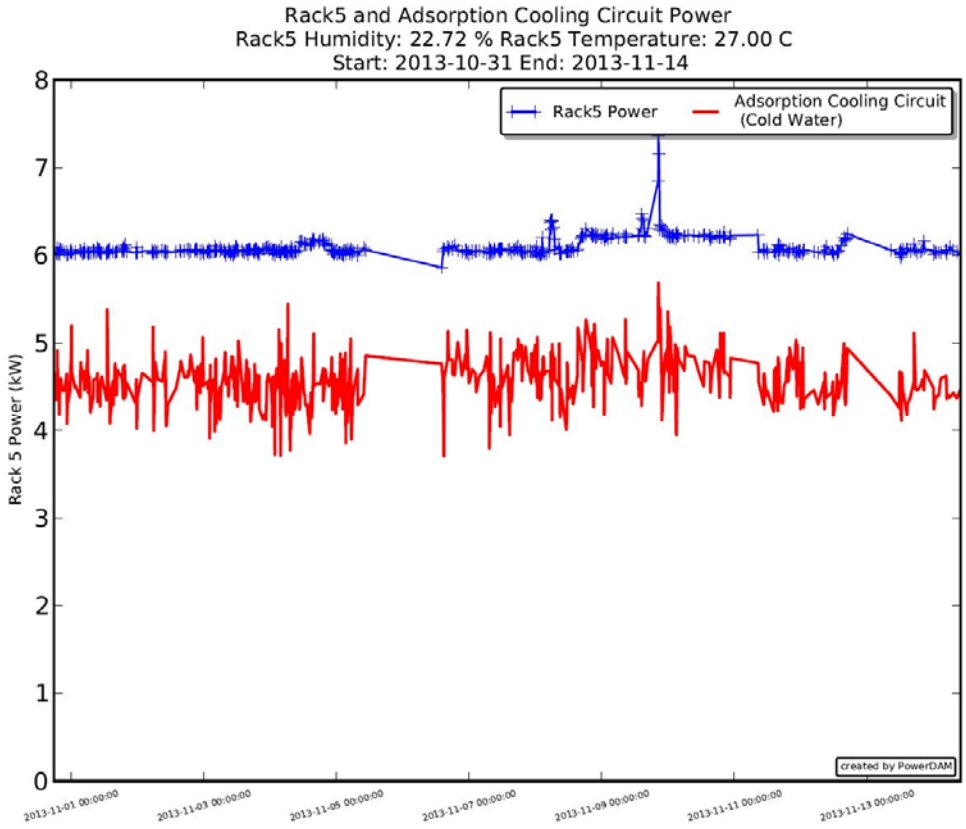


Figure 38 Heat removed by the adsorption chiller and power consumed by the additional rack (Rack 5).

5.6.3 *PUE and pPUE*

PUE stands for the Power Usage Efficiency (PUE) of a data centre. It was introduced by the Green Grid [GRGRID] to help data centre operators to assess and improve the efficiency of their data centre. It is defined as:

$$PUE = \frac{P_{Total}}{P_{IT}} \quad \text{where}$$

- P_{Total} is the complete power consumed by the data centre
- P_{IT} is the power consumed by the IT equipment in the data centre

Because of its definition [GRGRID12] PUE can't be used to determine the efficiency of a specific part (or sub-system) of a data centre. Therefore, pPUE [GRGRID11] was introduced by the Green Grid to report the power usage efficiency of sub-systems of a data centre. The pPUE (Figure 39) of the CooLMUC prototype when compared with pERE (Figure 41) can show possible benefits of an adsorption chiller.

The pPUE is defined similarly to the PUE but is system specific:

$$pPUE = \frac{P_{TotalCoolMUC}}{P_{ITCoolMUC}} \quad \text{where}$$

- $P_{TotaltCooLMUC}$ is the complete energy consumed by the system ($P_{ITCooLMUC}$+ $P_{Cooling}$)

- $P_{ITCooLMUC}$ is the power consumed by the IT equipment of the system (e.g. compute nodes and networking)

The pPUE value can never go below 1 because 1 means that all power going into the system is used for doing IT work. For example a pPUE of 2 would mean that from the ingoing power only half is used for the IT equipment inside the system, or in other words, running the system incurs a 100% overhead.

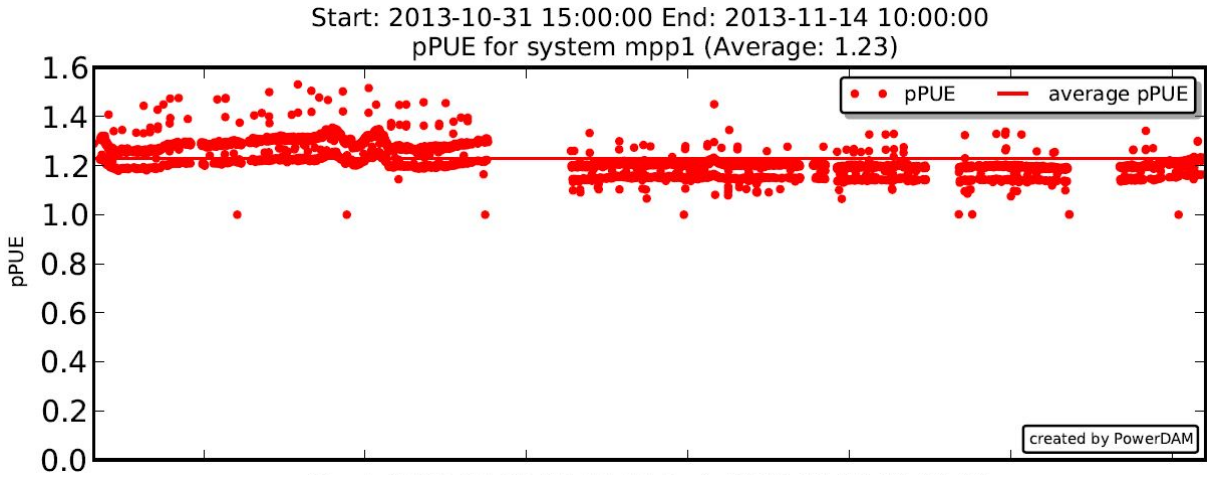


Figure 39 Partial PUE (pPUE) of the CooLMUC system.

Figure 39 shows the partial PUE (pPUE) of the CooLMUC system. As can be seen, the pPUE of the CooLMUC system for the recorded timeframe was 1.23. It is a good PUE value for a closed system that uses a combination of water and air-cooling.

5.6.4 *ERE and pERE*

ERE stands for Energy Reuse Efficiency. It combines the PUE and the energy re-used from the data centre. An ERE of 0 would mean that all energy going into the data centre is reused outside of it. ERE is defined as:

$$ERE = \frac{P_{TotalLRZ} - P_{ReuseLRZ}}{P_{ITLRZ}} \text{ where}$$

- $P_{TotalLRZ}$ is the complete power consumed by the LRZ data centre
- $P_{ReuseLRZ}$ is the power that crosses the data centre boundary and is used somewhere else
- P_{ITLRZ} is the power consumed by the IT equipment in the LRZ data centre

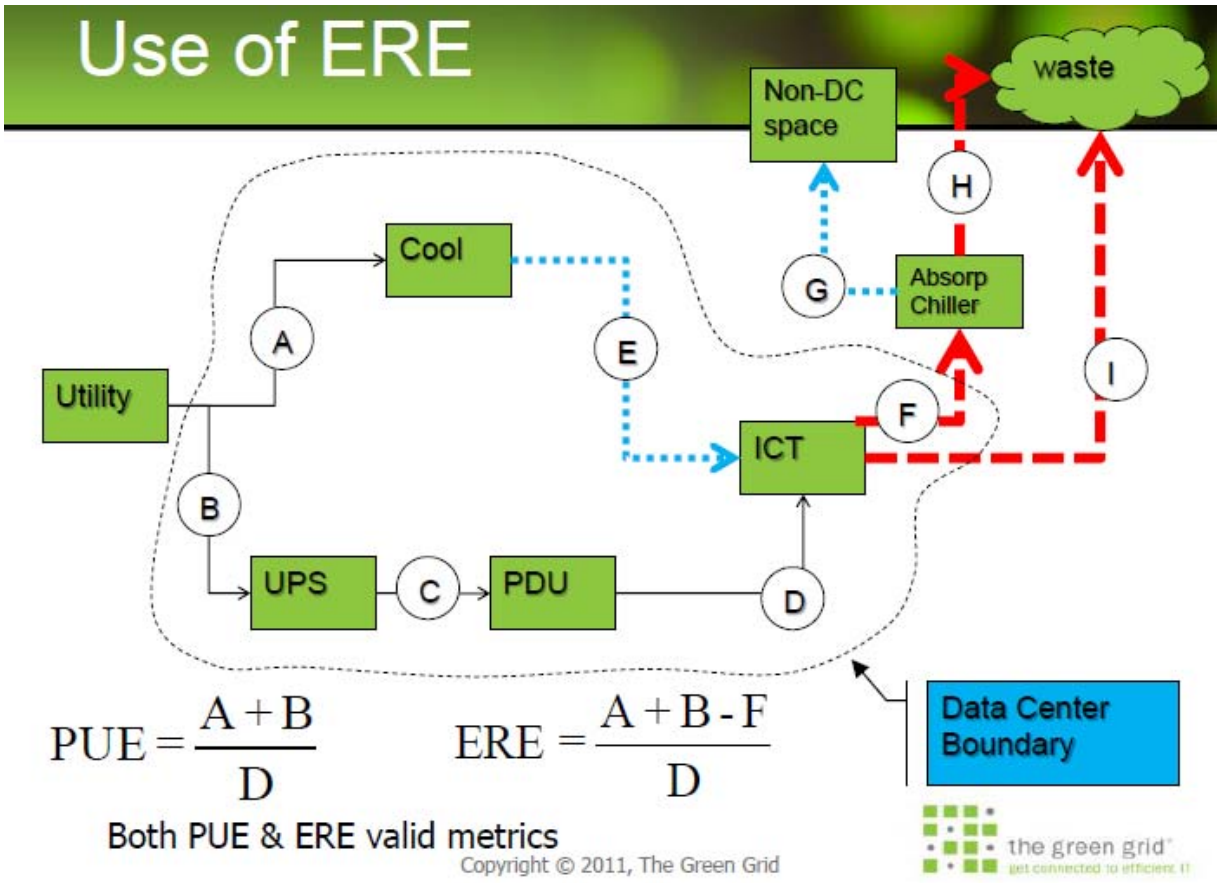


Figure 40 PUE, ERE, and Data Centre Boundary as defined by the Green Grid [GRGRID11].

As defined by the Green Grid, the ERE metric can only be used if the PUE of the facility will not change with/or without heat re-use (e.g. ERE can't be used if the heat is used inside the data centre itself) (Figure 40). Therefore, ERE can't be used to evaluate the impact of the adsorption chiller on the power efficiency of the CooLMUC prototype. This led to the definition of partial ERE (pERE) by LRZ. It is similar to the pPUE. It considers a sub-system inside the data centre and draws boundaries around it. If the heat re-use is not affecting the pPUE of the system, pERE can be used. pERE is defined as:

$$pERE = \frac{P_{TotalSubSystem} - P_{ReuseSubSystem}}{P_{ITSubSystem}} \quad \text{where}$$

- $P_{TotalSubSystem}$ is the complete power consumed by the sub-system
- $P_{ReuseSubSystem}$ is the power that crosses the sub-system boundary and is used somewhere else
- $P_{ITSubSystem}$ is the power consumed by the IT equipment in the sub-system

This means the hot water energy used by the adsorption chiller to cool the additional rack (Rack 5, not part of CooLMUC) can be seen as crossing the system boundary. Therefore, the power taken out of the heat from the hot water cooling circuit ($P_{AdsorptionHotWaterPower}$) can be accounted for in the pERE for CooLMUC:

$$pERE = \frac{P_{TotalCoolMUC} - P_{AdsorptionHotWaterPower}}{P_{ITCoolMUC}} \quad \text{where}$$

$P_{AdsorptionHotWaterPower} = P_{HotWaterPower} + P_{AdsorptionElectricPower}$

Figure 41 shows the plot of the partial Energy Re-use Efficiency (pERE) of the CooLMUC system. The average pERE was 1.03 for the observed period. By comparing the pPUE and the pERE of the CooLMUC prototype it can be seen that the overall energy balance of the system improved from 23% overhead to effectively 3% by re-using 20% of the IT energy elsewhere. This is very good.

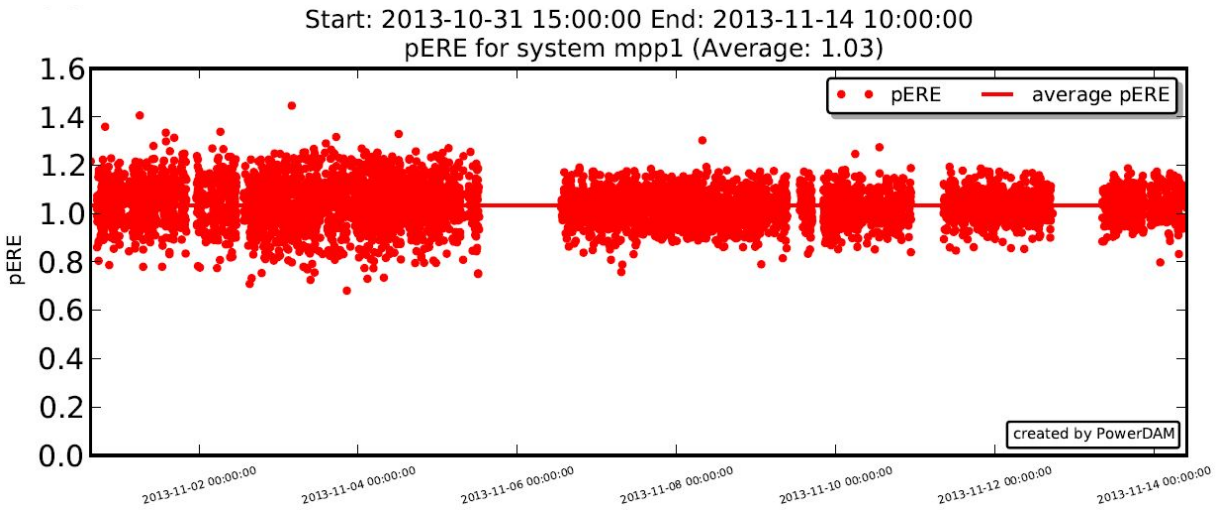


Figure 41 Partial ERE, pERE, for the CooLMUC.

5.7 Conclusion

Even though the collected data does not cover a long period of time it shows that using the heat of a HPC system to power an adsorption chiller to cool water works. For the recorded period it decreased the energy overhead of the system by 20% from 1.23 (pPUE) to 1.03 (pERE). Assuming an average COP of 0.5 and 100% heat capture of the SuperMUC HPC system, adsorption chillers could provide 1.2 MW of cooling (average power consumption of SuperMUC is 2.4 MW). This would be quite substantial. However, there are of course many factors to keep in mind.

Currently, it is not possible to capture 100% of the generated heat in a hot-water-cooling loop. Hot-water-cooling works well for components that have a high temperature threshold, (like CPU, GPU, memory, etc.). But there are components that degrade very quickly with higher temperatures, such as capacitors commonly used on motherboards. Additionally, some components are not designed with direct liquid cooling in mind, such as the power supplies and InfiniBand switches.

Depending on the design of the cooling system, higher inlet temperatures might cause higher power consumption of the system which will reduce any benefit of improved re-use of the generated heat using increased operating temperatures. In the CooLMUC case the need to provide cold air inside the system for the air cooled components limited the ability of the adsorption chiller to produce chilled water. Because it is a closed system, air is cooled using compressors. These use the hot water coming from the CPUs to cool themselves down. So, with increased water temperature, the removal of the compressor heat became more and more inefficient, resulting in increased electrical power consumption of the compressors and overall higher system air temperatures. The higher air temperatures led to increased component failures of heat sensitive parts like the CooLMUC power distribution units (PDUs). Physics

tells us that with increasing temperature difference between the hot water and the circulating air, more heat is transferred from the hot-water-cooling components into the air, which adds load on the air-cooling system.

To increase the efficiency of heat re-use some improvements will be needed. One would be to physically separate the air-cooled components from the direct liquid cooled components (e.g. move networking equipment like the InfiniBand switches into separate racks). Another would be to liquid cool all components on the motherboard using either cold plate or heat spreaders.

Another improvement in efficiency could be achieved if the adsorption chiller could be designed for lower water temperatures. Instead of the currently required 65°C and up, it should work efficiently at lower temperatures such as 50°C. This would allow for a lower hot water temperature, which, in turn, will result in lower system power consumption.

Seeing that heat re-use via adsorption requires hot water, it is a complementary technology to hot-water-cooling (depending on the geographical location and climate). Hot-water-cooling does not require any active chillers and, therefore, saves energy and floor space.

The current generation of adsorption chillers is using a lot of valuable floor space. To create the fictional 1.2 MW of cooling from the SuperMUC power consumption, one would require 334 units of the adsorption chiller used in the CooLMUC prototype. Clearly the units need to be more compact and more powerful. It would be very beneficial for the practical use of adsorption-chiller cooling if the units could either fit inside a standard data centre rack or on top of it. In this way, no special space has to be prepared or provided.

5.8 Lessons learned

Our experience with the CooLMUC prototype shows that anticipating all the sensors required for assessment and control when employing new technologies is difficult even with careful planning. Installing new system sensors after the system is up and running can be quite challenging. Part of this is due to the installation itself but another is the integration of the new sensors into the monitoring software. Even though CooLMUC is a prototype system it could be beneficial to do a detailed risk analysis together with the requirements specification to find areas of the system where sensors are required.

The adsorption chiller used with the CooLMUC prototype was installed with a dry re-cooling unit. This re-cooler can only cool the water down to the outside air temperature. This creates a problem for the adsorption chiller during summer. For an efficient adsorption process, a maximum cooling water temperature of 28°C is recommended. The outside temperature can reach above 30°C for parts of the summer in Munich. Therefore, future adsorption chillers need to use evaporative or hybrid coolers. Here the wet bulb temperature, a combination of air temperature and humidity, determines the cooling temperature. The wet bulb temperature never rises above 28°C in Munich. It would be even better if the adsorption chiller cooling loop were connected to the cooling loop of the data centre. In this way no additional outside water pipe connections need to be made, which reduces costs and removes the possibility of additional rainwater penetration points.

6 Advanced Multilevel Fault Tolerance (AMFT)

Fault tolerance and application resiliency will be a key issue for multi-Peta-scale and Exa-scale systems, as identified by many reports including reports by the IESP and EESI. Checkpointing of Peta-scale systems on a remote file system can take considerable time, up to 30 minutes on current large systems. Checkpointing of Exa-scale systems is expected to take even longer since the network bandwidth is not expected to grow as much as the total memory size. With an expected MTTI of less than 24 h, frequent checkpointing is a necessity for Exa-scale applications.

Results for the prototype are only reported in this deliverable.

6.1 Key Objectives

The AMFT approach to scalable checkpoint/restart is a combination of application-level checkpointing, saving only key variables, and exploiting the different levels of storage available on HPC systems in performing asynchronous high frequency checkpointing. Application-based checkpoint/restart can be realised by adding appropriate function calls for storing pertinent data to application codes, or through directives used by runtime systems for saving pertinent data structures. The AMFT uses the FTI (Fault Tolerance Interface) library co-developed by the INRIA-Illinois joint laboratory on Peta-scale computing and Tokyo Institute of Technology. The FTI library is written in C/MPI and Python. FTI is agnostic to target applications and can be used by simply linking with the FTI library. For applications already featuring application-level checkpointing, existing checkpoint/restart function calls are replaced by corresponding FTI function calls. For other applications, the programmer uses the FTI APIs in the same way one would implement application-level checkpoint/restart explicitly but, in using the FTI library, avoids the complexity of explicitly managing multilevel resiliency, garbage collection and metadata management. In addition, FTI features several configuration parameters that can be easily set up in a configuration file.

The objective of the AMFT prototype was three-fold:

1. evaluate the performance, scalability and overhead of the AMFT approach using different local storage technologies: standard HDD, hybrid HDD/SSD, regular SSD, optimised SSD and possibly new NVRAM technologies like Phase Change Memory;
2. adapt key applications selected from the PRACE benchmarks, the UEABS (Unified European Applications Benchmark Suite) or proposed community codes from PRACE-1IP T7.2 and 2IP-WP8 for AMFT using the Fault Tolerant Interface (FTI) library, and evaluate the effort and complexity of this adaptation and;
3. enhance the programming interface and the performance of the FTI library to advance it towards production level quality.

6.2 Prototype Description

6.2.1 *Hardware Description*

Two hardware resources were used for the assessment:

- Curie, a PRACE Tier-0 production system hosted at TGCC of CEA with about 5200 compute blades each equipped with local SSD devices and
- Ambre, an experimental cluster hosted at CINES used for testing next generation technologies.

The Curie 2 PF/s system is composed of three different and complementary x86 partitions:

- A fat-node partition composed of 360 Bull S6030 nodes, each node having 4 eight-core Intel Nehalem EX CPUs and 128 GB of memory. These nodes are configured as 90 super nodes (128 cores and 512 GB of memory) using a dedicated 4-node interconnect chip called BCS (Bull Coherent Switch).
- A hybrid-node partition composed of 144 Bull B505 blades, each blade having 2 quad-core Intel Westmere EP CPUs, 2 NVIDIA M2090 GPUs and a 128 GB local SSD (Micron C400).
- A thin-node partition composed of 5040 Bull B510 blades, each blade having 2 eight-core Sandy Bridge EP (E5-2680) 2.7 GHz CPUs, 64 GB of memory and a 128 GB local SSD (Micron C400).

The three partitions are interconnected through a full non-blocking fat-tree topology QDR InfiniBand network and share a two level Lustre parallel filesystem providing more than 15 PB of storage and 250 GB/s of aggregate bandwidth.

Both the hybrid-node and thin-node partitions were used for the AMFT assessment. The Micron RealSSD C400 uses NAND Multi-Level Cell (MLC) technology and has a 6 Gbps SATA interface. Early I/O benchmarks showed a bandwidth of about 354 MB/s for read and 228 MB/s for write, validating that the C400 is a good choice for HPC systems.

The SGI Altix XE320 Ambre system at CINES was acquired for the PRACE-PP project. It is composed of 32 nodes interconnected by a QDR IB network and has access to a 750 TB Lustre filesystem having 21 GB/s of bandwidth. Each node has 2 quad-core Intel Nehalem EP CPUs, 32 GB of memory and a local hard drive. The Ambre system was used for assessing the Texas Memory Systems (TMS) 1U 720 product sold by IBM as Flashsystem 720 after IBM's acquisition of TMS. Interesting features of this product that has an IB interface are the integration of a Xilinx FPGA and a Power PC CPU both in charge of offloading control operations (like write setup, garbage collection, error handling, formatting units, backup/restore, statistics collection and similar functions), and the integration of multiple memory controllers enabling multiple concurrent DMA operations on multiple flash units of the 720. The 720 also integrates a proprietary ECC within each 512-byte data set and a patented Variable Stripe RAID (VSR) including RAID 5 across memory modules. The acquired Flashsystem 720 is a 10 TB Single Level Cell (SLC) system. The typical chip endurance of SLC Flash is about 100□000 program/erase cycles, compared to about 30□000 cycles for enterprise Multiple Level Cell (e-MLC) and about 3□000 cycles for standard Multiple Level Cell (MLC) technologies. The specification of the IBM Flashsystem 720 is given in Table 13, the actual hardware is shown in Figure 42. The integration of the Flashsystem 720 into the Ambre cluster is shown in Figure 43.

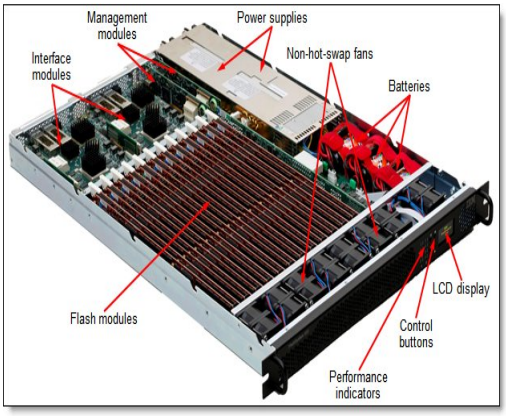


Figure 42 Illustration of the components of the IBM FlashSystem 720.

Specification FlashSystem 720	
Form factor	1U rack-mounted unit
Flash module quantity	12 (10+1+1)
Flash module type	Single level cell (SLC)
Flash module capacity	Double-density (DD): 1 TB
Total capacity	RAID 0 or JBOF 12 TB (DD) RAID 5: 10 TB (DD)
Flash module protection	Overprovisioning, wear levelling, CRC checksum, ECC, no single point of failure.
Host interfaces	4x 40 Gb QDR InfiniBand (or 8 Gb/s FCAL ports)
Read IOPS	525 000
Write IOPS	400 000
Read bandwidth	5 GB/s
Write bandwidth	4 GB/s
Read latency	100 µs
Write latency	25 µs
Input power	350 W

Table 13 Specifications for the IBM FlashSystem 720.

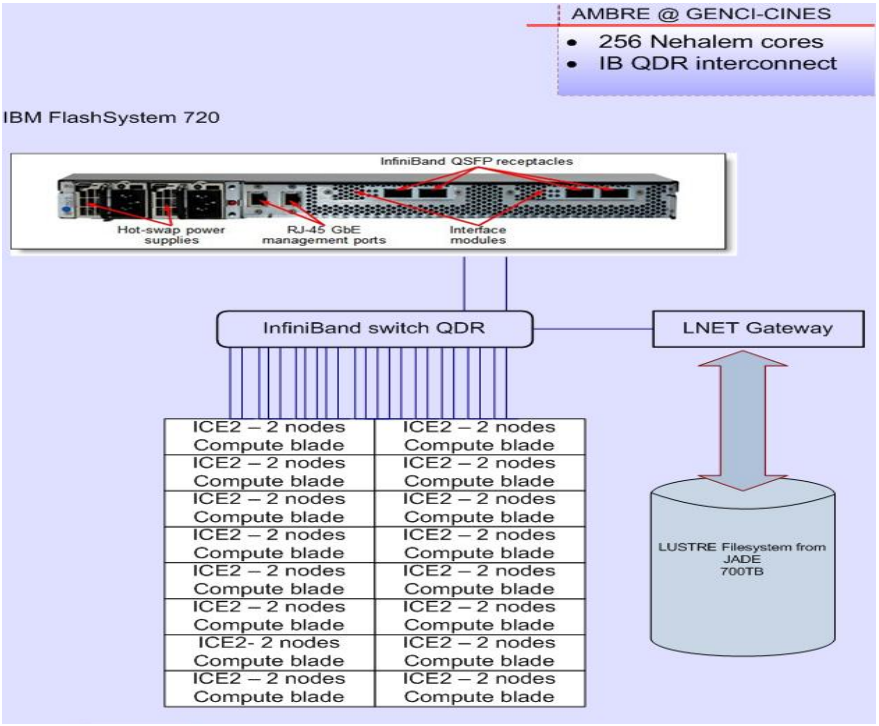


Figure 43 Integration of the IBM FlashSystem 720 into the Ambre cluster IB network.

6.3 The FTI library

For a successful assessment of the AMFT approach and for enhanced applicability of the FTI, the library was extended as follows:

- A Fortran interface was developed by CEA to complement the existing C interface.
- Python was removed to improve portability.
- An option to keep checkpoint files after successful job termination instead of deleting them was added to allow scheduled restarts for multiple stage jobs.
- Duration based checkpointing was implemented. Originally the FTI library only supported checkpointing based on iteration count. This was of some concern since iteration times can be hard to predict and vary greatly.
- The FTI library originally used a dedicated MPI process on each node that may lead to a waste of resources. A thread-based version was implemented for synchronous mode; extension to other levels is under study.
- A synchronous mode was implemented with no requirement for a dedicated process. This mode lacks advanced features but it allows improves portability to machines with restrictions on the number of processes per node.
- The API was redesigned to pass more information about the checkpointed variables. This will allow future enhancements for corruption detection and data compression.
- A study of the possible use of FTI without local (SSD) storage using a "memory map" (mmap) mode was initiated.

A complete description of the FTI library is available online [FTI]. The FTI library v0.8 released in June 2013 offers four different levels of fault protection:

- L1: checkpointing to local storage without any concern for failure of the hardware. This is the fastest of the four levels. This level offers protection against software bugs causing crashes and soft-errors, which are expected to be very common for Exa-scale machines. This level incurs the lowest overhead, but offers the lowest resilience.
- L2: in addition to local checkpointing as in L1, checkpoint data is duplicated to a partner node enabling an application to recover from failure of one node in each partnership.
- L3: checkpoints are allocated to multiple nodes according to the formation of processor groups across nodes as shown in the Figure 44. A Reed-Solomon algorithm is used and the resulting data stored in a way such that failures of up to half of the nodes in a group can be tolerated (for groups of size k, k/2 simultaneous failures).
- L4: checkpoints are written to the parallel file system (PFS). This level offers the best fault tolerance (assuming the PFS is more reliable than compute nodes) but is also the slowest. To reduce the performance impact of this level, the local storage system is used by the FTI library as a buffer before asynchronously writing the data to the PFS while the application continues its computation.

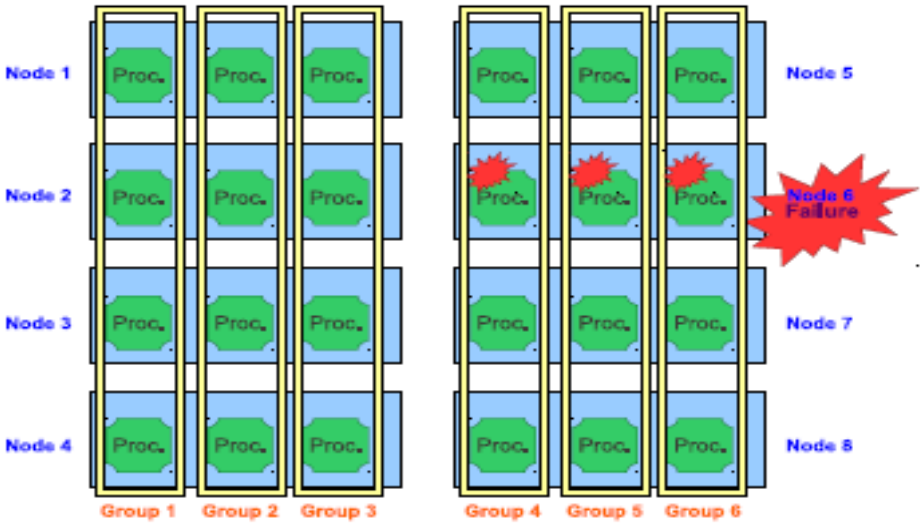


Figure 44 Grouping of processes into redundancy groups to sustain node failure in FTI.

The first three levels of the FTI library enable frequent checkpointing without severe performance impact, yet offer protection against hardware failures of modest extent. Use of any of the first three levels can also justify reducing the frequency of Level 4 checkpoints. It is also worth noting that local checkpointing allows a faster recovery than the PFS-only checkpoints, especially if a large number of nodes are involved.

The FTI has four main functions:

- int FTI_Init (char *configFile, MPI_Comm globalComm)
  This function will initialise FTI.
- int FTI_Protect (int id, void *ptr, long size)
  It stores a pointer to a variable that needs to be protected.
- int FTI_Snapshot ()
  This function takes an FTI snapshot or recovers the data in case of a restart.
- int FTI_Finalize ()
  This function closes FTI properly on the application processes.

In conjunction, a parameter file (config.fti) specifies to the FTI :

- Checkpoint directories (local, global, metadata)
- Checkpoint interval
- Levels (Partner copy, Reed Solomon, asynchronous and related combinations)
- Group organization for levels
- Restart behaviour
- Tuning

6.4 Application impact assessment

In an earthquake simulation of the March 11, 2011 Tohoku event using SPECFEM3D, it was demonstrated that the FTI library adds only 8% to the simulation duration with checkpointing every 6 minutes on the Peta-scale TSUBAME 2.0 system at TITech, Japan, see Figure 45. [GOM11]

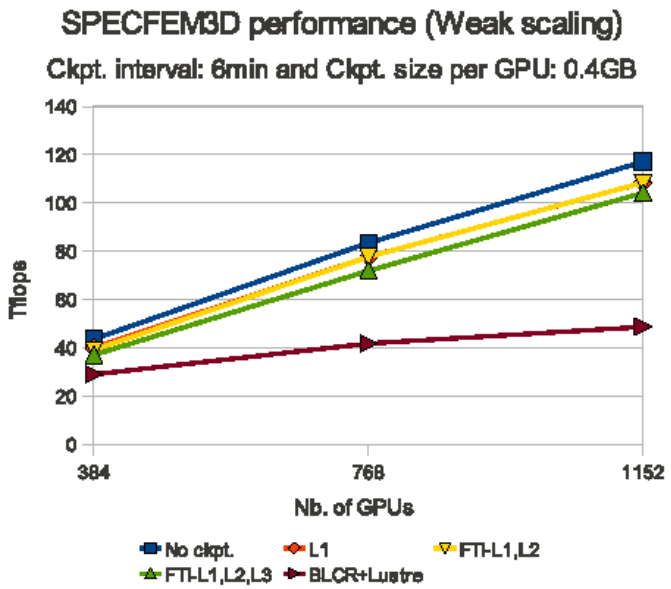


Figure 45 Weak scaling of SPECFEM3D using no checkpoint (in blue), FTI (in yellow and green) and remote checkpoint on Lustre using BLCR.

For our assessment we used the Hydro code used also in several of the other prototype assessments. No version of the code had any fault tolerance system implemented. As a second application Gysela5D [GYS5D], a gyro-kinetic code for the simulation of plasma in a Tokamak, was selected in collaboration with CEA. Gysela5D was shown to efficiently scale to 458 752 cores (1 835 008 parallel threads) on BlueGene/Q (JUQUEEN at JSC) and 65 536 cores on x86 (Curie at TGCC/GENCI and Helios in Japan).

6.4.1 *Hydro results on Curie*

The grid sizes used for Hydro for up to 9600 cores are shown in Table 14 below and the performance impact for different FTI levels for 255 MB/core checkpoints at 6 min intervals shown in Figure 46.

Grid Size	Number of cores used
50□000 x 100□000	600
100□000 x 100□000	1200
100□000 x 200□000	2400
200□000 x 200□000	4800
300□000 x 200□000	7200
400□000 x 200□000	9600

Table 14 Grid sizes and corresponding core counts used for the AMFT assessment for the Hydro benchmark.

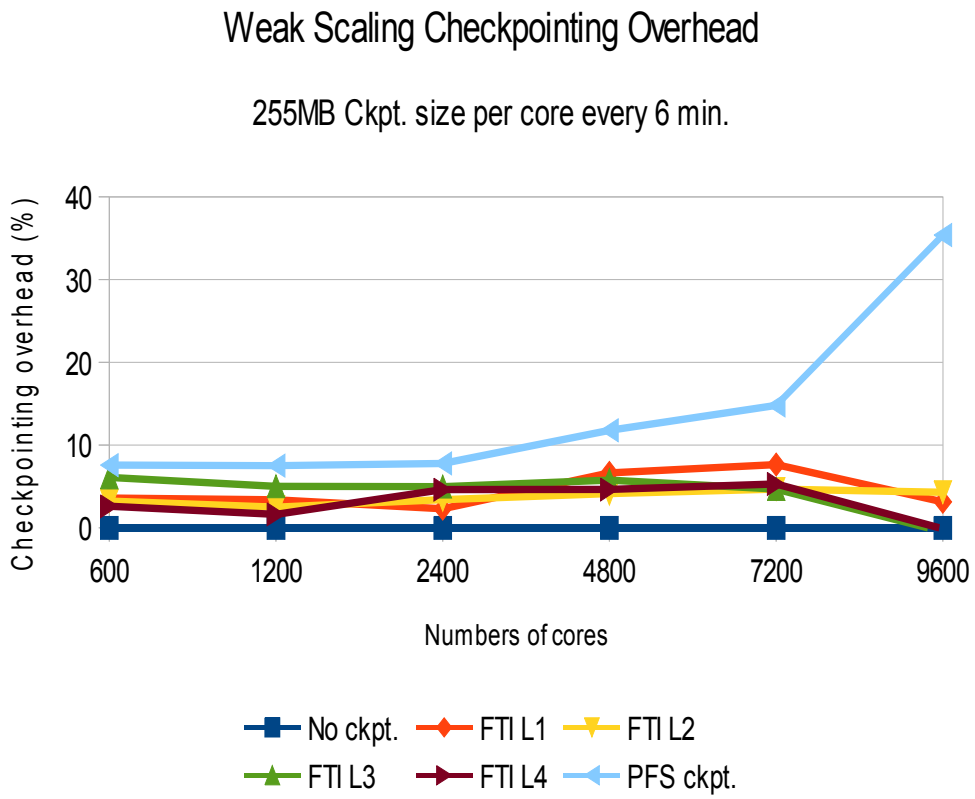


Figure 46 Overhead of the AMFT approach using the FTI library for various checkpointing levels.

The overhead induced by FTI for Hydro was below 6% fairly independent of the grid size and number of nodes, whereas conventional checkpoints to the file system (PFS) increased exponentially (light blue curve).

The results presented in Figure 46 were collected with Curie in normal production, causing some variability in the measurements. A variability of about 5% is commonly observed on this system. We assume this variability is responsible for the fact that in some cases Level 4 exhibits lower overhead than Level 1.

Some screenshots illustrating the use of the FTI in Hydro are shown in Figure 47 and Figure 48. For Hydro, see Figure 47, nine variables are selected to be checkpointed by the FTI library through calls to the "FTI_Protect" function. Note that pre-processor directives (#ifdef FTI==1 for example) are used to allow including the FTI library as an option. The actual saving of the collection of variables selected by "FTI_Protect" is performed by "FTI_Snapshot", see Figure 48.

```
  //pre-allocate memory before entering in loop
  //For godunov scheme
  start = cclock();
  start = cclock();
  allocate_work_space(H.nxyt, H, &Hw_godunov, &Hvw_godunov);
  compute_deltat_init_mem(H, &Hw_deltat, &Hvw_deltat);
  end = cclock();
#ifdef MPI
#if FTI==1
  FTI_Protect(0,functim, TIM_END,FTI_DBLE);
  FTI_Protect(1,&nvtk,1,FTI_INTG);
  FTI_Protect(2,&next_output_time,1,FTI_DBLE);
  FTI_Protect(3,&dt,1,FTI_DBLE);
  FTI_Protect(4,&MflopsSUM,1,FTI_DBLE);
  FTI_Protect(5,&nbFLOPS,1,FTI_LONG);
  FTI_Protect(6,&(H.nstep),1,FTI_INTG);
  FTI_Protect(7,&(H.t),1,FTI_DBLE);
  FTI_Protect(8,Hv.uold,H.nvar * H.nxt * H.nyt,FTI_DBLE);
#endif
#endif
  if (H.mype == 0) fprintf(stdout, "Hydro: init mem %lfs\n", ccelaps(start, end));
  // we start timings here to avoid the cost of initial memory allocation
  start_time = dcclock();

  while ((H.t < H.tend) && (H.nstep < H.nstepmax)) {
    // reset perf counter for this iteration
    flopsAri = flopsSqr = flopsMin = flopsTra = 0;
    start_iter = dcclock();
    outnum[0] = 0;
    if ((H.nstep % 2) == 0) {
      dt = 0;
```

Figure 47 Modifications to Hydro to declare the data to be saved by FTI.

```
    if (H.mype == 0) {
      fprintf(stdout, "--> step=%4d, %12.5e, %10.5e %s\n", H.nstep, H.t, dt, outnum);
      fflush(stdout);
    }
#ifdef MPI
#if FTI==1
    // FTI snapshot
    FTI_Snapshot();
#endif
#endif
  } // while
  end_time = dcclock();

  // Deallocate work spaces
  deallocate_work_space(H.nxyt, H, &Hw_godunov, &Hvw_godunov);
  compute_deltat_clean_mem(H, &Hw_deltat, &Hvw_deltat);

  hydro_finish(H, &Hv);
  elaps = (double) (end_time - start_time);
```

Figure 48 Modification to Hydro to enable FTI based checkpoint/restart.

6.4.2 *Gysela5D*

The Gysela5D application consists to 95% of Fortran90 code and is parallelised with OpenMP and MPI. Two implementations providing fault tolerance exist already in the code, both using the HDF5 I/O library:

- A Fortran implementation, using synchronous I/O.
- A C implementation using a dedicated thread and performing asynchronous I/O.

As part of our assessment, a third version was added using the FTI library. We tried to make the different implementations as similar in functionality as possible. The resulting code was used for large simulations in production mode.

To be able to use the FTI library from Fortran, and therefore in Gysela5D, a suitable interface module was developed to bridge the differences in the C and Fortran calling conventions. Since Fortran prevents generic pointers (void pointers in C), a bash script was written to automate the generation of type specific interface functions.

All the FTI function calls required by Gysela5D were isolated in two application specific functions to minimise modifications to the common code base. Besides simple API calls, management of the MPI Communicators had to be extended to suit the requirements of the

FTI library.

The correctness of the checkpoint/restart mechanism was verified using a special deterministic mode available in the Gysela5D application that guarantees bit-exact results across different simulation runs in conjunction with the ability to generate diagnostic checksums of selected properties of the simulated system during execution.

The stream of checksums created by two different simulations, one using FTI based checkpoint and restart and one running uninterrupted were compared and agreed for more than 97% of the values. The disagreement was attributed to the specific way in which Gysela5D determined the simulation time at which the diagnostic checksums were generated, which was based on counters that were reset in case the application was restarted from a checkpoint. This feature of the application allowed checkpoints to be used to interrupt, control and restart long running simulations. Unfortunately it could also cause a misalignment in the checksums if the checkpoint/restart happened at the wrong time.

6.4.3 *SSD assessment*

An assessment was also made of the performance of the SSD technologies used on Curie, single SATA SSD per node, and Ambre, the QDR IB connected 12 SSD Flashsystem 720. The results using the IOR benchmark with 1 MB blocks and a 40 GB file size are shown in Figure 49. The Curie SSDs have a perfect aggregate scaling and reach the performance of the 12 SSD FlashSystem720 with 16 nodes.

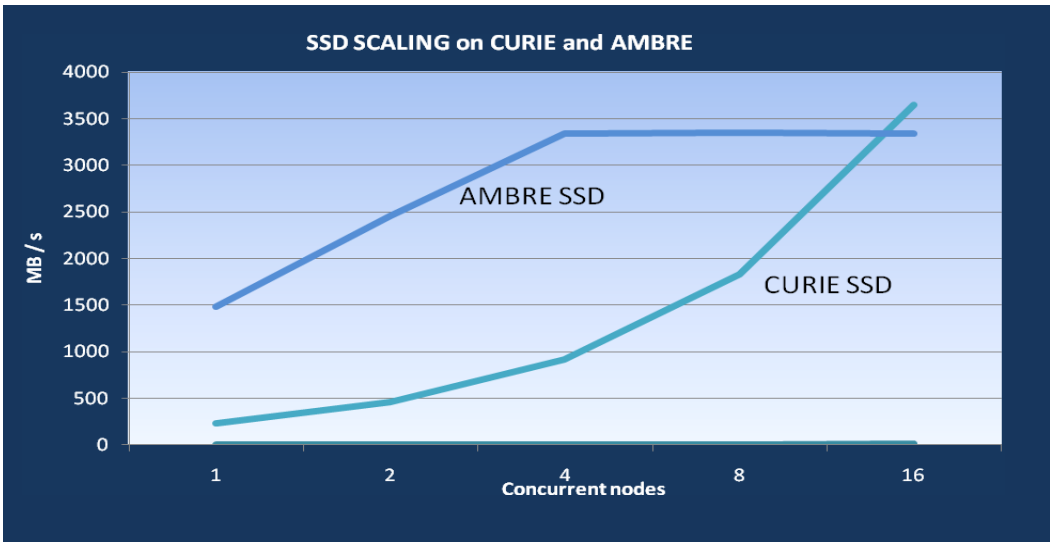


Figure 49 Measured bandwidth for the IOR benchmark on the QDR IB connected Flashsystem 720 and aggregate bandwidth for Curie nodes with single SATA SSDs.

6.4.4 *Conclusion and future directions*

The QDR IB IBM FlashSystem 720 with 12 SSDs had a maximum throughput in our measurements of about 3.4 GB/s, a throughput that was achieved on CURIE with 16 nodes each with one SATA connected SSD.

The use of the FTI library for the Hydro application showed a modest impact on performance with checkpointing at 6 min intervals. For the tests, the FTI library was run on 1 core of each node with the applications using the remaining cores. Hyperthreading was explored, but adverse effects on performance were observed; most likely due to degraded memory affinity.

The effort to incorporate the FTI in the Hydro and Gysela5D codes is estimated to be 1 to 2

months for each code. The Hydro code had no checkpointing prior to this effort whereas Gysela5D had two existing checkpoint implementations using HDF5.

As part of the AMFT prototype effort the FTI library was extended to support Fortran application codes (required for the Gysela5D code) and the usability was enhanced.

Future developments of the FTI library include exploiting the new resilient features of MPI3 and the possibility to use main memory with the mmap feature for checkpointing on local and adjacent nodes memory into account.

Technology improvements are expected with the availability of a new generation of SSD technology based on 28 or 22 nm memory cells, 3D NAND stacking, better flash controllers, better ECC algorithms, increased local storage and PCIe 3.0 attachment to the host system. It is also expected that TLC (3 bits MLC instead of current 2-bits technology) technology will be available in high volumes with a better capacity and reduced price per GB.

The main breakthrough will come with availability of resistive NVRAM, like Phase Change Memory (PCM), where energy (heat) converts material between crystalline (conductive) and amorphous (resistive) phases. This technology promises a 1000x gain in speed compared to current NAND technology and a performance close to DRAM. PCM was not available within the timeframe of the project and could not be assessed. Figure 50 details expected improvements using future NVRAM technologies.

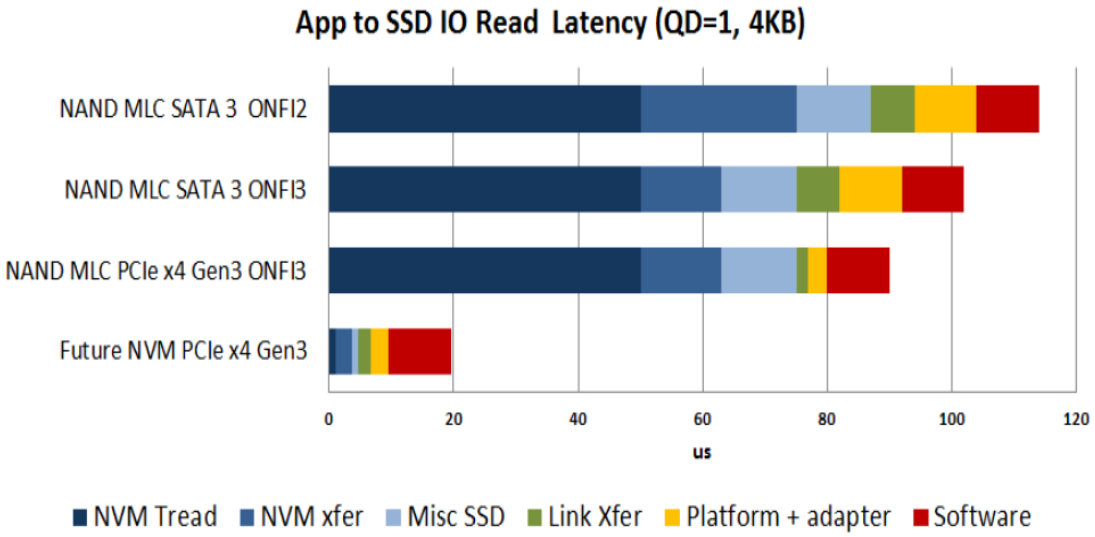


Figure 50 Estimated performances of future non-volatile memory technologies.

7 Conclusion

In regards to energy efficiency of technologies for future HPC systems we have validated that, for the TI TMS320C6678 DSP for DGEMM, HPL and STREAM, efficiencies comparable to those of well-optimised code on x86 architectures were achieved. That means that the nominal energy efficiency advantage this DSP has over x86 based CPUs also holds in practice for the benchmarks we had the opportunity to optimise in evaluating the DSP. In fact, the 40 nm DSP, that by now is more than two years old, is more energy efficient than the recent 22 nm Ivy Bridge x86 CPU. The ARM plus GPU prototype based on ARM Cortex-A9 cores and NVidia Kepler GPUs clearly demonstrated that this ARM generation does not have the capabilities to be a serious contender for HPC systems in regards to energy efficiency. On the other hand, the GPU is highly energy efficient for suitable workloads as is also clear from the Green500 list ranking computer systems on their energy efficiency for the HPL benchmark. Based on our benchmarks, DSPs are an interesting alternative to x86 CPUs in regards to energy efficiency, and so are GPUs as accelerators. DSPs are complete CPUs and do not require a host, unlike the current generation of GPUs.

To achieve the potential energy efficiency advantage of the alternative architectures WP9 studied, high efficiencies in resource utilization are necessary, which requires well-optimised codes. In fact, even for x86 architectures with mature software environments for code development and optimization, using codes “out-of-the-box” does not guarantee good performance. However, not unexpectedly, for non-traditional HPC architectures the optimization efforts were quite time consuming as they included: learning about details of the architectures and available tools; developing methodologies for effective use of the architectures; and translating those into programming strategies and working code. Naturally, available programming tools for debugging and performance monitoring are less evolved, as are compilers and availability of optimised libraries relevant for HPC applications. Though this was expected, the full extent of the necessary effort was not anticipated in the project planning.

Understanding memory systems is critical to achieving good performance. This became very apparent for the SMP prototype, particularly. Though the large shared address space offered by the prototype, built out of standard servers, provides many conveniences in developing a working code, good performance still requires attention to the architecture of the memory system, and allocating data and computations (threads) accordingly. The different versions of the STREAM benchmark vividly demonstrate the NUMA aspects of the shared memory. Tools, such as OpenMP, are focused on thread allocation for load balance and affinity, but do not address data allocation.

Quality benchmarking is a non-trivial undertaking. Including energy efficiency assessment, at a sufficiently detailed level that it can provide guidance for future designs of hardware and software or for dynamic power management, adds significant complexity in that it requires a good understanding of instrumentation and measurement technologies and their pitfalls as well as to system and program behaviours. All prototype efforts “struggled” with this challenge and in several cases instrumentation needed to be revised and complemented as it became clear that information was either missing or not sufficiently detailed or accurate to draw firm conclusions.

Through the additional instrumentation of the energy recovery technology prototype at LRZ, the benefits of the technology could be demonstrated and operational characteristics could be much better understood than at the time for the D9.3.3 deliverable. Through the heat recovery system, about 20% of the energy used for the cluster could be used elsewhere. The prototype

at PSNC, reported on in this deliverable, targeted immersive cooling in which compute blades have their own liquid filled enclosures. This new product by Iceotope did need a few iterations of engineering improvements for reliable operation, which in combination with a need for revised instrumentation lead to few results being reported here.

The Advanced Multi-level Fault Tolerance approach seems to be a very promising approach to low overhead checkpointing. It was demonstrated for the Hydro application benchmark code that frequent checkpointing could be carried out with significantly less than 10% overhead. It was also demonstrated that the approach has very good scalability, unlike traditional checkpointing.