E-Infrastructures
H2020-EINFRA-2016-2017

EINFRA-11-2016: Support to the next implementation phase of Pan-European High Performance Computing Infrastructure and Services (PRACE)

PRACE-5IP

PRACE Fifth Implementation Phase Project

Grant Agreement Number: EINFRA-730913

D5.6

Extended best practice guide for prototypes and demonstrators
Final

Version: 1.0
Author(s): Hayk Shoukourian (BADW-LRZ)
Date: 08.04.2019

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: EINFRA-730913**	
	Project Title: Extended best practice guide for prototypes and demonstrators	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: D5.6	
	Deliverable Nature: Report	
	Dissemination Level: PU*	**Contractual Date of Delivery:** 30 / April / 2019
		Actual Date of Delivery: 30 / April / 2019
	EC Project Officer: Leonardo Flores Añover	

* **PU** – Public

Document Control Sheet

Document	**Title: Extended best practice guide for prototypes and demonstrators**	
	ID: D5.6	
	Version: 1.1	**Status:** *Final*
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2013	
	File(s): PRACE-5IP-D5.6_v1.0-1_JL	
Authorship	**Written by:**	Hayk Shoukourian (BADW-LRZ)
	Contributors:	Carlo Cavazzoni (CINECA) Giannis Koutsou (CaSToRC) Radosław Januszewski (PSNC) Walter Lioen (SURFsara) Philippe Segers (GENCI) Volker Weinberg (BADW-LRZ)
	Reviewed by:	Massimiliano Guarrasi, CINECA Thomas Eickermann, JUELICH
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	14/May/2018	Draft	Skeleton, the very first draft.
0.2	01/August/2018	Draft	Assembly of some information from

			ISC'18. First complete draft for Sections 5 and 6. Skeleton for Section 4 including the data from survey. Finalizing the document for distribution to WP5 T3 partners with an input request.
0.3	07/October/2018	Draft	Additional input for sections 5 and 6 obtained from Giannis Koutsou (CaSToRC).
0.4	03/December/2018	Draft	Input from Walter Lioen (PRACE 5IP WP7 representative) regarding sections 5 and 6.
0.5	16/January/2019	Draft	Input from Carlo Cavazzoni (CINECA) for section 3.
0.6	12/February/2019	Draft	Input from Radosław Januszewski (PSNC) for section 4. First complete draft of introduction and conclusion sections.
0.7	28/February/2019	Draft	Includes comments and fixes from Giannis Koutsou (CaSToRC) and Hayk Shoukourian (LRZ).
0.7.1	11/March/2019	Draft	Additional input on PCP and EuroHPC from Philippe Segers (GENCI).
0.8	12/March/2019	Draft	Version submitted for WP5-internal review.
0.8.1	27/March/2019	Semi-final	Version submitted for PRACE internal review.
1.0	08/April/2019	Final	Addressing comments/suggestions from reviewers

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, User Prototyping Requirements

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

[1] PRACE-5IP Deliverable D5.5 “Requirements of new user communities for the use of next generation computing systems evolving towards Exascale”, 2018
[2] http://www.prace-ri.eu/
[3] PRACE-4IP Deliverable D5.6 “Best Practices for Prototype Planning and Evaluation”, 2017
[4] http://www.prace-ri.eu/prace-4ip/
[5] http://www.prace-ri.eu/prace-pp/
[6] http://www.prace-ri.eu/prace-1ip/
[7] http://www.prace-ri.eu/prace-2ip/

[8] http://www.prace-ri.eu/prace-3ip/
[9] http://montblanc-project.eu/
[10] http://www.deep-project.eu/deep-project/EN/Home/home_node.html
[11] http://hpc.desy.de/qpace/
[12] https://www.cineca.it/
[13] https://www.cyi.ac.cy/index.php/castorc/about-the-center/castorc-center-overview.html
[14] https://www.csc.fi/
[15] https://www.lrz.de/english/
[16] https://grnet.gr/en/
[17] https://www.edu.unideb.hu/
[18] http://www.man.poznan.pl/online/en/
[19] https://pg.edu.pl/welcome?p_l_id=52858455&p_l_id=2601414&p_v_l_s_g_id=0&p_v_l_s_g_id=0&
[20] https://www.hartree.stfc.ac.uk/Pages/home.aspx
[21] PRACE-4IP Deliverable D5.5 “Application and HPC Centre Requirements for Prototyping”; 2016
[22] https://ec.europa.eu/digital-single-market/en/news/european-processor-initiative-consortium-develop-europes-microprocessors-future-supercomputers
[23] https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx
[24] www.ucolick.org/~zingale/flash_benchmark_io/
[25] https://computing.llnl.gov/?set=code&page=sio_downloads
[26] https://software.intel.com/en-us/articles/intel-mpi-benchmarks
[27] http://www.hpcg-benchmark.org/
[28] https://www.top500.org/
[29] http://www.netlib.org/benchmark/hpl/
[30] https://github.com/luizbafilho/hydra
[31] Carrington, Laura C., Michael Laurenzano, Allan Snavely, Roy L. Campbell, and Larry P. Davis. "How well can simple metrics represent the performance of HPC applications?." In Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference, pp. 48-48. IEEE, 2005
[32] https://www.nas.nasa.gov/publications/npb.html
[33] https://www.intel.com.tw/content/www/tw/zh/design/test-and-validate/platform-testing-services/power-supply-testing.html
[34] Stone, H. L. (1968). "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations". SIAM Journal of Numerical Analysis. 5 (3): 530–538. DOI:10.1137/0705044
[35] Hager G., Deserno F., Wellein G.: Pseudo-Vectorization and RISC Optimization Techniques for the Hitachi SR8000 Architecture. In: Wagner S., Bode A., Hanke W., Durst F. (eds) High Performance Computing in Science and Engineering, Munich 2002. Springer, Berlin, Heidelberg. DOI:10.1007/978-3-642-55526-8_34
[36] Que, Xinyu, Lars Schneidenbach, Fabio Checconi, Carlos HÃ Costa, and Daniele Buono. "Performance Analysis of Spark/GraphX on POWER8 Cluster." In International Conference on High Performance Computing, pp. 268-285. Springer, Cham, 2016
[37] https://www.cs.virginia.edu/stream/

[38] https://repository.prace-ri.eu/git/UEABS/ueabs
[39] http://www.prace-ri.eu/IMG/pdf/D7.7_v1.2_4ip.pdf
[40] S. Matsuoka, "From TSUBAME to ABCI onto Post-K: Convergence of HPC and AI in Exascale", HPC w/ARM workshop, Shanghai, July 2018, https://hpc.sjtu.edu.cn/HPCARMChina20180721matsuoka.pdf
[41] https://abci.ai/jp/about_abci/
[42] http://www.tpc.org
[43] Fox, G., Jha, S., Qiu, J., Ekanazake, S. and Luckow, A., 2015. Towards a comprehensive set of big data benchmarks. Big Data and High Performance Computing, 26, p.47.
[44] https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench
[45] Wang, L., Zhan, J., Luo, C., Zhu, Y., Yang, Q., He, Y., Gao, W., Jia, Z., Shi, Y., Zhang, S. and Zheng, C., 2014, February. Bigdatabench: A big data benchmark suite from internet services. In High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on (pp. 488-499). IEEE.
[46] http://prof.ict.ac.cn/
[47] https://github.com/bigframeteam/BigFrame
[48] https://graph500.org/
[49] https://amplab.cs.berkeley.edu/benchmark/
[50] https://github.com/intel-hadoop/HiBench
[51] http://cloudsuite.ch/
[52] https://github.com/brianfrankcooper/YCSB/wiki
[53] https://www.spec.org/
[54] https://www.spec.org/events/beijing2016/slides/005-SPEC-RG%20Big%20Data%20Overview%20-%20Xiao%20Wei%20Zhang.pdf
[55] https://github.com/google/gemmlowp
[56] https://svail.github.io/DeepBench/
[57] https://mlperf.org/
[58] https://github.com/mlperf/reference
[59] http://deep500.org/
[60] https://www.hpcwire.com/2019/01/07/the-deep500-researchers-tackle-an-hpc-benchmark-for-deep-learning/
[61] https://github.com/deep500/deep500
[62] https://www.hpcwire.com/2019/02/05/deep500-eth-researchers-introduce-new-deep-learning-benchmark-for-hpc/

List of Acronyms and Abbreviations

AI	Artificial Intelligence
CoE	Center of Excellence
CPU	Central Processing Unit
DDR	Double Data Rate
DEISA	Distributed European Infrastructure for Supercomputing Applications
FLOPS	Floating Point Operations Per Second
FPGA	Field-programmable gate array
GB	Giga (= 2^{30} ~ 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
GDDR	Graphics DDR
GEMM	General Matrix Multiplication
GHz	Giga (= 10^9) Hertz, frequency =10^9 periods or clock cycles per second
GPU	Graphic Processing Unit
HBM	High Bandwidth Memory
HDF	Hierarchical Data Format
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
HPCG	High Performance Conjugate Gradients
HPL	High Performance LINPACK
HW	Hardware
IMB	Intel MPI Benchmarks
IOR	Interleaved Or Random
I/O	Input/Output
KPI	Key Performance Indicator
ML	Machine Learning
MPI	Message Passing Interface
NASA	National Aeronautics and Space Administration
NVM	Non Volatile Memory
PCP	Pre-Commercial Procurement
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PTU	Power Thermal Utility
SIP	Strongly-Implicit Procedure
SPEC	Standard Performance Evaluation Corporation
SW	Software
TDP	Thermal Design Power
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
TPC	Transaction Processing Performance Council
TRL	Technology Readiness Levels
UDF	User Defined Function
UEABS	Unified European Applications Benchmark Suite
YCSB	Yahoo! Cloud Serving Benchmark

List of Project Partner Acronyms

BADW-LRZ	Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften, Germany (3rd Party to GCS)
BILKENT	Bilkent University, Turkey (3rd Party to UYBHM)
BSC	Barcelona Supercomputing Center - Centro Nacional de Supercomputacion, Spain
CaSToRC	Computation-based Science and Technology Research Center, Cyprus
CCSAS	Computing Centre of the Slovak Academy of Sciences, Slovakia
CEA	Commissariat à l'Energie Atomique et aux Energies Alternatives, France (3 rd Party to GENCI)
CESGA	Fundacion Publica Gallega Centro Tecnológico de Supercomputación de Galicia, Spain, (3rd Party to BSC)
CINECA	CINECA Consorzio Interuniversitario, Italy
CINES	Centre Informatique National de l'Enseignement Supérieur, France (3 rd Party to GENCI)
CNRS	Centre National de la Recherche Scientifique, France (3 rd Party to GENCI)
CSC	CSC Scientific Computing Ltd., Finland
CSIC	Spanish Council for Scientific Research (3rd Party to BSC)
CYFRONET	Academic Computing Centre CYFRONET AGH, Poland (3rd party to PNSC)
EPCC	EPCC at The University of Edinburgh, UK
ETHZurich (CSCS)	Eidgenössische Technische Hochschule Zürich – CSCS, Switzerland
FIS	FACULTY OF INFORMATION STUDIES, Slovenia (3rd Party to ULFME)
GCS	Gauss Centre for Supercomputing e.V.
GENCI	Grand Equipement National de Calcul Intensiv, France
GRNET	Greek Research and Technology Network, Greece
INRIA	Institut National de Recherche en Informatique et Automatique, France (3rd Party to GENCI)
IST	Instituto Superior Técnico, Portugal (3rd Party to UC-LCA)
IUCC	INTER UNIVERSITY COMPUTATION CENTRE, Israel
JKU	Institut fuer Graphische und Parallele Datenverarbeitung der Johannes Kepler Universitaet Linz, Austria
JUELICH	Forschungszentrum Juelich GmbH, Germany
KTH	Royal Institute of Technology, Sweden (3rd Party to SNIC)
LiU	Linkoping University, Sweden (3rd Party to SNIC)
NCSA	NATIONAL CENTRE FOR SUPERCOMPUTING APPLICATIONS, Bulgaria
NIIF	National Information Infrastructure Development Institute, Hungary
NTNU	The Norwegian University of Science and Technology, Norway (3rd Party to SIGMA)
NUI-Galway	National University of Ireland Galway, Ireland
PRACE	Partnership for Advanced Computing in Europe aisbl, Belgium
PSNC	Poznan Supercomputing and Networking Center, Poland
RISCSW	RISC Software GmbH

RZG	Max Planck Gesellschaft zur Förderung der Wissenschaften e.V., Germany (3rd Party to GCS)
SIGMA2	UNINETT Sigma2 AS, Norway
SNIC	Swedish National Infrastructure for Computing (within the Swedish Science Council), Sweden
STFC	Science and Technology Facilities Council, UK (3rd Party to EPSRC)
SURFsara	Dutch national high-performance computing and e-Science support center, part of the SURF cooperative, Netherlands
UC-LCA	Universidade de Coimbra, Labotatório de Computação Avançada, Portugal
UCPH	Københavns Universitet, Denmark
UHEM	Istanbul Technical University, Ayazaga Campus, Turkey
UiO	University of Oslo, Norway (3rd Party to SIGMA)
ULFME	UNIVERZA V LJUBLJANI, Slovenia
UmU	Umea University, Sweden (3rd Party to SNIC)
UnivEvora	Universidade de Évora, Portugal (3rd Party to UC-LCA)
UPC	Universitat Politècnica de Catalunya, Spain (3rd Party to BSC)
UPM/CeSViMa	Madrid Supercomputing and Visualization Center, Spain (3rd Party to BSC)
USTUTT-HLRS	Universitaet Stuttgart – HLRS, Germany (3rd Party to GCS)
VSB-TUO	VYSOKA SKOLA BANSKA - TECHNICKA UNIVERZITA OSTRAVA, Czech Republic
WCNS	Politechnika Wroclawska, Poland (3rd party to PNSC)

Executive Summary

Prototyping is an integral part of the HPC design process that assists in detecting and resolving any issues related to the use of new concepts and technologies. In general, the prototyping of HPC systems has the following objectives:

- *to test and refine the functionality of a given HPC design and enable a better understanding of its purpose*;
- *to save time and costs by empowering the possibility of early influencing design changes*;
- *to assess the usefulness and applicability of new technologies and design approaches*; and last but foremost
- *to address the requirements of current user communities of HPC centres in an efficient way by gaining inputs and insights regarding the usage of the production system*.

This best practice guide aims to deliver information and guidance useful for the evaluation of prototype HPC systems with regard to their usability and fit for purpose. For achieving this goal, this document extends the previously developed (by PRACE-4IP WP5) best practice guide to include guidelines on: (i) a set of benchmarks most suitable for HPC prototyping; and (ii) a set of system and development tools that are currently used by PRACE Tier-0/Tier-1 sites for HPC prototype evaluation, while reflecting on existing co-design opportunities.

The guide's intended audience is prototype and/or demonstrator owners for testing on actual HPC prototypes and demonstrators stemming from EU-funded projects such as MontBlanc, DEEP/DEEP-ER/DEEP-EST, or future Pre-Commercial Procurements (PCP) on HPC. It could also prove useful for FET-HPC technological projects, and also for communities represented within Centers of Excellence (CoEs), providing both groups with some best practice for the assessment of prototypes and demonstrators, useful for the system design as well as for early and customized tests of system usability by thematic communities.

This best practice guide has been designed taking into account the current race to Exascale, with the goal of providing valuable input for the activity of the EuroHPC Joint Undertaking[1].

[1] EuroHPC Joint Undertaking: https://eurohpc-ju.europa.eu/

1 Introduction

The design of a High Performance Computing (HPC) system is a complex task that needs to meet various contrasting demands ranging from: *(i)* the divergent requirements of user communities and system performance targets; over *(ii)* data centre's power delivery, cooling, and floor-space constraints; to *(iii)* capital and operational expenses. Additionally, as these high-end systems scale to support the ever-increasing performance demand, the underlying components become more complex and diverse, making the performance of target HPC systems strongly dependent on the choice of certain hardware (HW) and software (SW) technology and their corresponding configuration. All these demands further increase the importance of HW/SW co-design activities and make the efficient procedures for HPC planning, commissioning, and evaluation even more complex.

This document aims to provide a guide for HPC prototype/demonstrator owners that assists with the evaluation of these systems with regard to their usability and fit for purpose by building on:

a) previously identified requirements of new user communities for the use of next generation HPC systems [1]; and
b) experiences of PRACE [2] Tier-0/Tier-1 sites gathered during various EU-funded projects related to HPC prototyping [3].

More specifically, the document extends the previous PRACE-4IP [4] WP5 efforts that provided an overview on: *(i)* the individual phases of HPC prototyping projects; and *(ii)* prototyping experiences of PRACE Tier-0/Tier-1 sites that stemmed from previous PRACE and other FP7 and H2020 projects [3][2] to include tools encompassing:

a) a checklist for system and development tools that should be installed on a system before it is made available to general HPC users;
b) sets of benchmarks; and
c) Key Performance Indicators (KPIs)

for the productive evaluation of HPC prototyping activities.

The rest of this document is organised as follows. Section 2 lists the PRACE Tier-0 and Tier-1 HPC sites that completed the survey, developed by PRACE-5IP WP5 [1], on which some of the analysis of this deliverable is based on and Section 3 describes the co-design opportunities. Section 4 outlines the minimal requirements (as identified from PRACE Tier-0/Tier-1 sites) in terms of libraries, tools, etc. that should be installed on a system before it is accessed by general HPC users. Section 5 presents a set of benchmarks, including a representative set of synthetic open source kernel benchmarks, with a different set of properties that is used by PRACE Tier-0 and Tier-1 sites to evaluate newly deployed prototype systems. This set of benchmarks was prepared in cooperation with PRACE-5IP WP7, the application-focused work package, which is among other activities in charge of code enabling activities, publication of Best Practice Guides, and the development of the Unified European Applications Benchmark Suite (UEABS). Section 6 describes the main Key Performance Indicators (KPIs) that are most commonly used by PRACE HPC sites for prototype

[2] For example PRACE-PP [5], PRACE-1IP [6], PRACE-2IP [7], PRACE-3IP [8], Mont-Blanc [9], DEEP/DEEP-ER [10], and QPACE [11] projects.

evaluation. Finally, Section 7 provides an outlook, delineates future work, and concludes this report.

The survey as well as the raw data from which the results presented here were obtained have been uploaded to the PRACE repository and can be accessed via https://repository.prace-ri.eu/git/hayk.shoukourian/5IPT3.git (access restricted to PRACE-IP partners).

2 Survey

Some of the analysis presented in this document is based on the results of an online survey that was prepared during PRACE-5IP by WP5 contributors and distributed among PRACE Tier-0 and Tier-1 supercomputing sites [1]. The survey was conducted during the timeframe of October to December 2017.

The following PRACE Tier-0/Tier-1 sites participated in the survey:

PRACE Tier-0/Tier-1 site	Name of the flagship system	Country
CINECA [12]	MARCONI	Italy
Computation-based Science and Technology Research Center (CaSToRC), The Cyprus Institute [13]	Cy-Tera	Cyprus
CSC - IT Center for Science Ltd. [14]	Sisu	Finland
Leibniz Supercomputing Centre of the Bavarian Academy of Sciences (BAdW-LRZ) [15]	SuperMUC Phase 2	Germany
Greek Research and Technology Network (GRNET) [16]	ARIS	Greece
University of Debrecen [17]	VGGD	Hungary
Poznan Supercomputing and Networking Center (PSNC) [18]	Eagle / Hetman	Poland
Gdansk University of Technology [19]	Tryton	Poland
The Hartree Centre [20]	Scafell Pike	UK

Table 1: List of PRACE Tier-0/Tier-1 sites that participated in the survey.

Most of these HPC sites were involved in the previously mentioned PRACE prototyping projects, deploy prototype systems on a regular basis, and thus bring in significant expertise in terms of HPC prototyping [1, 3, 21].

3 Co-design opportunities

With the ever-increasing demand for performance, power consumption continues to remain an important design constraint for future HPC systems. Currently vendors are including all the more vertical or special purpose components (e.g. accelerators, tensor cores, neuromorphic chips, FPGAs, etc.), each designed to maximize the performance of a specific set of workloads. This leads to a richer design space to explore, with multiple opportunities for co-design activities in the near future. A notable addition most relevant to this deliverable is the European Processor Initiative (EPI) [22], with the stated goal to co-design and bring to market a processor to power future

Exascale systems based on European technologies. It is noteworthy that the EPI intends to solicit feedback from EU HPC applications and ecosystems as part of its co-design process.

In what follows, the document briefly reflects on co-design opportunities identified during PRACE-5IP, grouped according to the architectural component the co-design opportunity is most relevant to.

3.1 Co-design at the chip level

Traditionally, there have not been many opportunities for co-design at CPU chip level, with the HPC market and related use cases having limited influence in the design choices of these components. The aforementioned EU initiative for the design and production of European CPUs and Accelerators, suitable for next generation Exascale systems, should provide for more opportunities, as it explicitly states that these technologies will be developed within a co-design process. Examples of features that would be open for co-design within this initiative include: *(i)* the optimal vector unit size, namely in terms of the optimal length, in bits, suitable for the range of targeted EU scientific applications; *(ii)* a heterogeneous design that will include different cores, each dedicated for a given kind of workload (e.g. AI, Big Data, etc.), as well as *(iii)* communication methods and technologies for CPU and accelerator, with options being PCIe, NVlink, OpenCAPI, GenZ, etc. One particular design choice with implications on power consumption is the numerical precision of floating point and integer operations. It will therefore be important to provide feedback on the level of support of numerical precision required on such chips, which depends on whether applications can tolerate accuracies different from double precision floating point, i.e. 64bit IEEE.

3.2 Co-design at the integration level

At the integration level, there are much more opportunities for co-design compared to chip design, and this is exemplified by the eagerness of vendors to solicit feedback which helps them plan future products. European Exascale projects such as DEEP and Mont Blanc are also examples of co-designed HPC systems, with the co-design starting at the integration level. At the time of writing of this report, we can identify the two most characteristic elements of the architecture that are relevant for co-design at the integration level, namely High Bandwidth Memory (HBM) and Non Volatile Memory (NVM). The design choices available are the amount of HBM to be added to the CPU, on one side, and the available system bandwidth for NVM on the other side. These are both design choices that must be defined during the HPC node design stage. Besides HW, together with the node populated with different kinds of memory, an important co-design activity could be the evaluation of different memory models dealing with memory hierarchies (e.g. transparent or explicit memory usage).

3.3 Co-design for networking

Networking of an HPC system has traditionally been a component amenable to optimization and therefore co-design. This is largely thanks to the multiple protocols, vendors, and topologies available to integrators to choose from when designing an HPC system. Beyond these design choices, we have identified two emerging trends that could broaden the design space available for

this component. Namely, there are lately important movements towards better support for low latency network protocols that in turn allow for more efficient implementations of partitioned global address space (PGAS) programming models. Complementary to these developments is the inclusion of active network components, for example FPGA co-processors which implement the network interface, that allow for performing basic computations while the messages are being communicated. In terms of PGAS and future MPI standards, there are opportunities in co-designing functionalities that will facilitate applications to scale on multi-core heterogeneous system, where communication could take place e.g. between two accelerators or other similar devices. For active networks, co-design and feedback from application scientists are necessary for defining which functionalities to implement during message passing.

3.4 Co-design for energy management

Energy management, combined with energy monitoring and profiling, is becoming all the more an important topic in the design and running of HPC infrastructures. Co-design initiatives during the past decades, mostly through the PRACE prototype projects and PRACE PCP projects, have given Europe a competitive advantage in this regard when compared to international efforts. As new CPU and GPU architectures are being introduced, and with the forthcoming European processor being planned, there is an opportunity to leverage the work that has been done in the past years to co-design architectures compliant with the tools and middleware developed within Europe, such that leadership on energy management tools is maintained. One important issue would be to develop a standard for collecting and storing raw data, so that the access to the data will be kept open irrespective of the specific tools for monitoring and managing energy consumption.

3.5 Co-design for libraries

We see opportunities for co-design of libraries at two levels: *(i)* at the level of the interface of the library with the scientific application code; and *(ii)* at the level of the underlying hardware for which the library is optimized for. The former is important as the performance of application codes becomes more sensitive to the underlying math, I/O, and domain specific libraries. With this respect, engaging the developers of the libraries would be beneficial in designing their API to better support European applications. Actions within PRACE activities and European Centers of Excellence (CoEs) are already along these lines, which can be leveraged to start specific co-design actions, targeting the exploitation of new chips and system architectures. As regards co-designing the libraries for the underlying hardware, it will be important to have libraries able to exploit new instruction sets, such as for reduced precision arithmetic used in AI applications. Furthermore, HDF5 is gaining considerable attention from applications developers due to its performance and reliability characteristics for parallel I/O. It can be an additional opportunity for co-designing new I/O subsystems to better support HDF5, something which is already being included in co-design actions for Exascale in the US.

3.6 Co-design for compiler/programming standards

Among the wide range of programing models that target parallel processing and HPC, OpenMP appears to be the one which on one hand maintains broad support from both users and vendors and on the other hand continues to evolve introducing new features, such as support for accelerators. In particular, it is expected that the version 5.0 of the OpenMP standard and beyond will play an important role in exploiting future exascale architectures, providing many opportunities to engage with the community working in defining the OpenMP standard to better support the needs of the European applications and architectures. This is especially true regarding co-design actions related to new memory management and off-loading features already defined in the standard but which remain poorly supported by systems software, drivers, and compilers.

3.7 Co-design for data management/processing

As the availability of data increases, the need to manage and process it also grows, leading to the so-called convergence of HPC and HPDA (High Performance Data Analysis), both in terms of applications and infrastructure. At the European level, applications dealing with the convergence of large data analysis and HPC are being targeted by Centers of Excellence. At the infrastructure level, there are a large number of co-design initiatives, including the use of accelerators to speed-up data processing, the use of object store technologies as an alternative to POSIX filesystems, and different organization of storage tiers to reduce latency in accessing large datasets (e.g. usage of flash based devices with burst buffer like software technology).

3.8 Co-design for specific HPC workloads

Finally, at the HPC system level, co-design opportunities arise in tuning for a specific scientific workload (e.g. for AI applications), while at the same time similar opportunities arise in refactoring or adapting applications to best match the possibilities offered by new hardware. Examples include new algorithms that exploit mixed precision arithmetic, thus maximizing the performance and energy efficiency on certain types of HPC architectures. Indeed, many iterative algorithms may be refactored to perform most iterations at low precision, with just a few iterations required at full double precision, without an overall loss of accuracy. In this example, a co-designed HPC system would employ components that more optimally perform reduced-precision arithmetic, such as GPUs. Other examples include workloads that may require a preprocessing of data, such as a convolution or Fourier transform of data, before proceeding to the main calculations. Such workloads could benefit from a co-designed system that would include FPGAs to carry out the preprocessing step.

4 Checklist of system and development tools

In this section, we present the results of a survey in which we asked PRACE Tier-0 and Tier-1 sites to identify which system and development tools they consider most crucial to deploy on their prototypes. This provides us with a minimum list of system and development tools, the functionality of which should be somehow enabled on the prototype to allow for easier introduction

in a production environment and maintain aspects of compatibility with more familiar production systems. It should be noted that building a prototype may serve different purposes, therefore the list should be treated as a list of aspects that should be considered rather than a strict list of requirements.

In Figure 1, we show the survey responses for each category of system and development tools. As mentioned earlier in Section 2, the survey results are based on nine responses obtained from PRACE Tier-0 and Tier-1 HPC sites. The general conclusion that can be drawn is that development tools, parallel libraries, and job schedulers are considered by PRACE Tier-0 and Tier-1 sites as the most necessary tools to deploy on a prototype, while production application codes and special-purpose monitoring tools are not considered as crucial.

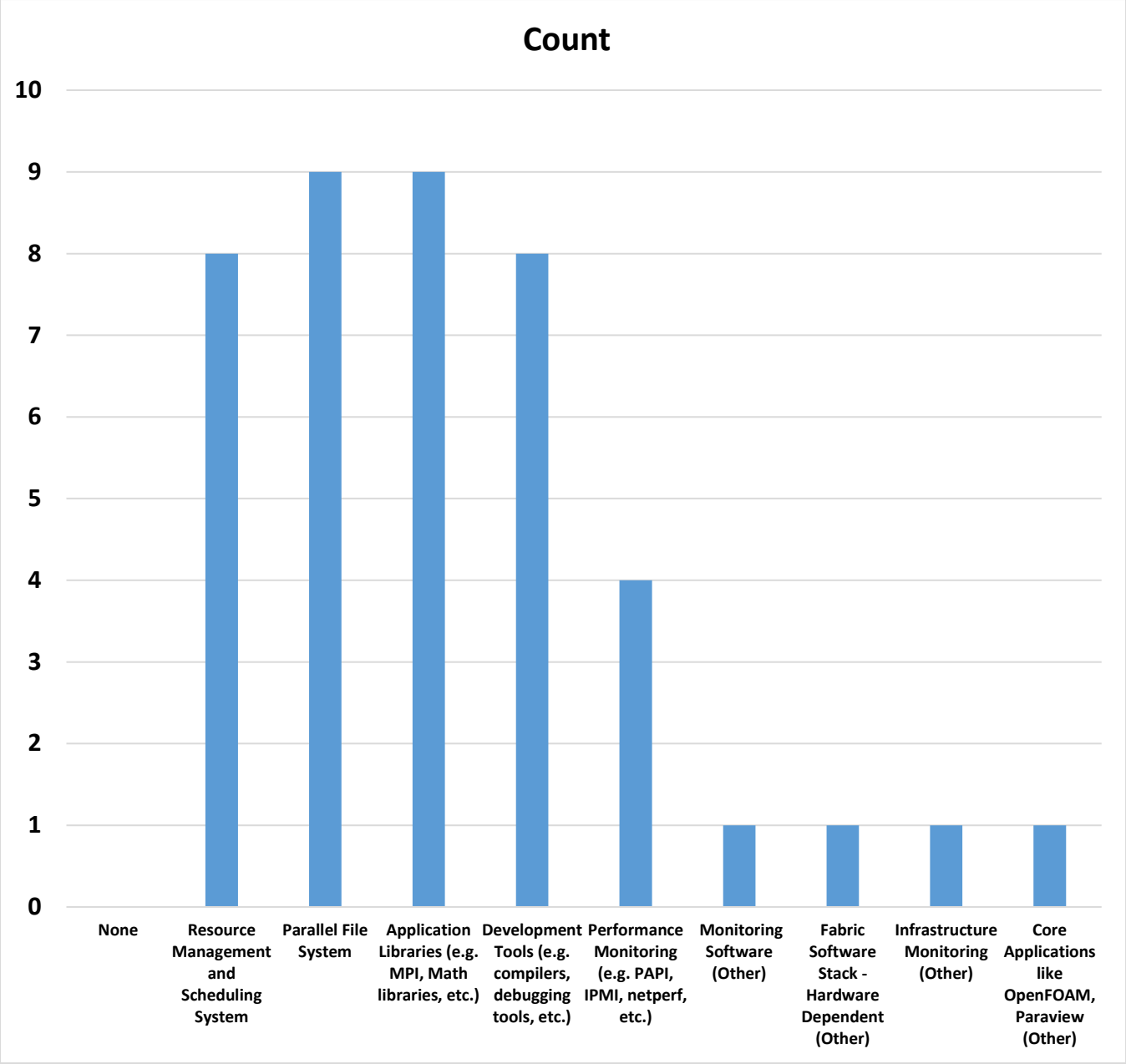

Figure 1 The minimum requirements of PRACE Tier-0/Tier-1 sites in terms of libraries, tools, etc. that should be installed on a system before it gets accessed by general HPC users

4.1 Resource management and scheduling system

The queuing system is a necessary tool whenever several users intent to access the same prototype. Amongst queuing systems, most respondents deploy SLURM on their prototypes, as shown in Figure 2. It is noteworthy that even in prototyping scenarios, co-design opportunities are tested with multiple teams having access to the same experimental environment, which indicates a certain level of maturity in the users as well as the tools that are able to accommodate novel technologies. It should also be noted that the basic functionality of all queuing systems is similar and that the required effort to migrate submission scripts from production systems seems to be less of an issue.

This would suggest that ease of deployment and use should be more significant criteria when deploying a queuing system on a prototype.

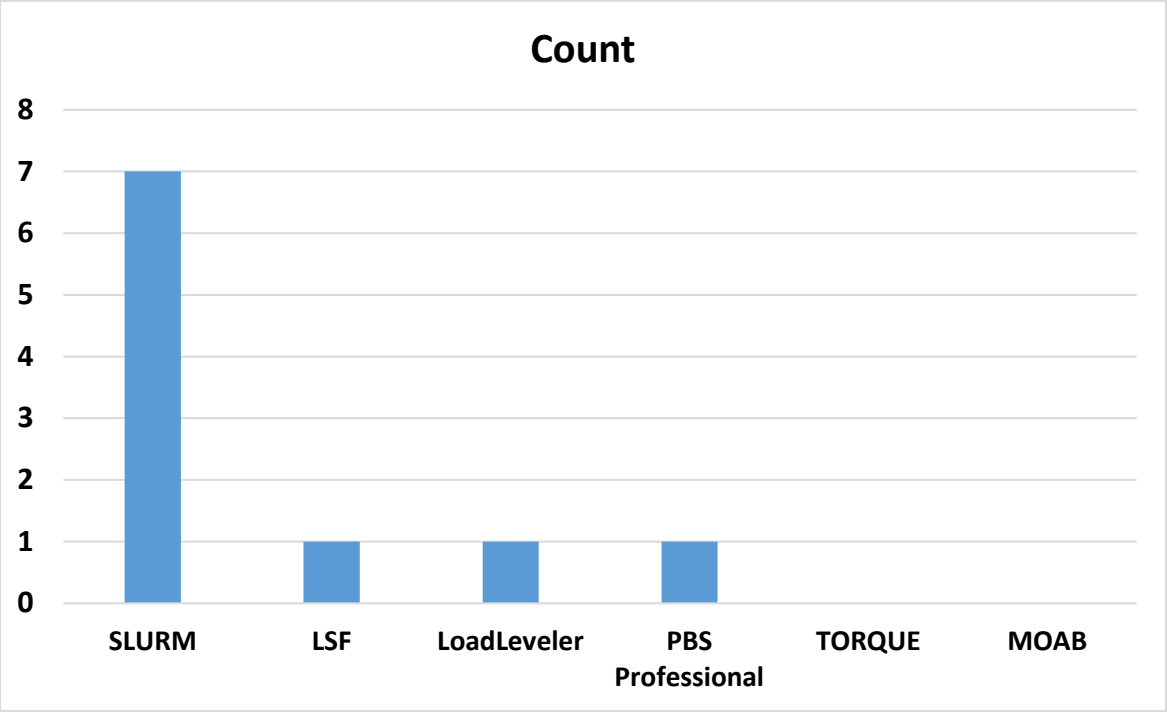

Figure 2 Resource management and scheduling systems currently used at PRACE Tier-0/Tier-1 sites

4.2 Parallel filesystem

As indicated in Figure 1, the vast majority of centres surveyed identified the availability of a parallel filesystem on their prototypes as a necessary component of the prototyping activity. This is despite recent developments and more common usage of object storage. As seen in Figure 3, all solutions used in prototyping by the Tier-0 and Tier-1 PRACE sites surveyed currently deliver POSIX compliant file systems, with the majority split between GPFS and Lustre.

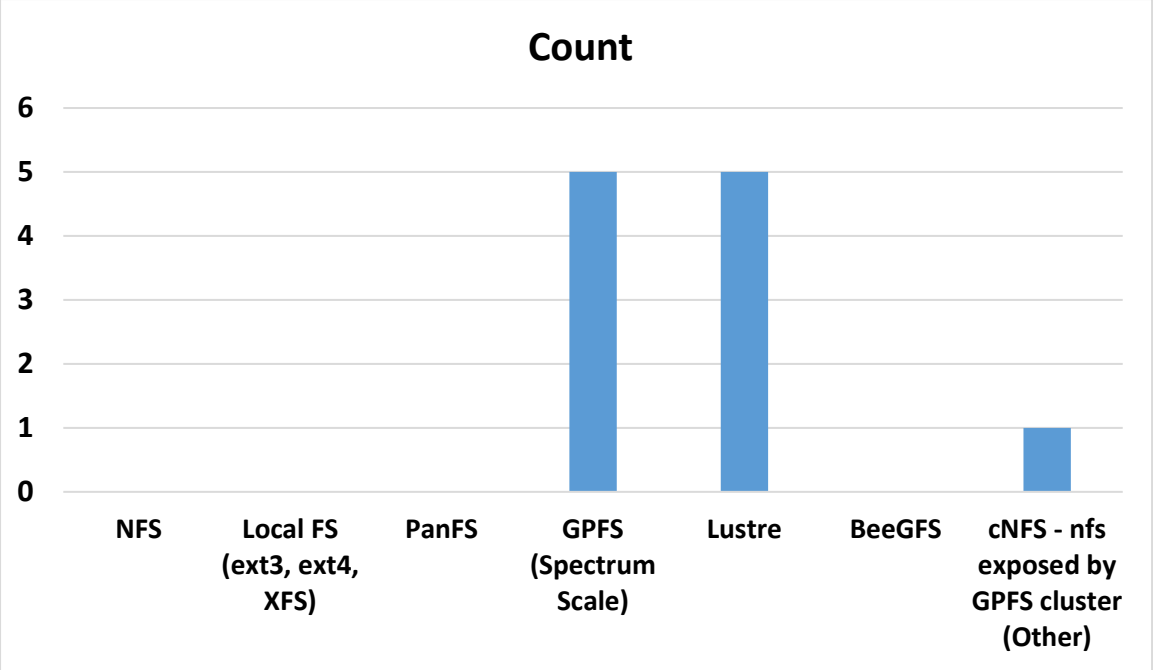

Figure 3 Parallel file system currently used at PRACE Tier-0/Tier-1 sites

4.3 Application libraries

The performance of an HPC system eventually depends on how efficiently it executes applications, and these, in turn, rely heavily on the availability of a certain set of identifiable HPC libraries. All surveyed centres identified the availability of such software as a determining factor in the usability of the prototype (see Figure 1). Before designing a prototype, it is, therefore necessary to check which libraries are needed and which are available for the system being designed. As the survey results shown in Figure 4 suggest, the most important aspect is the availability of an efficient MPI implementation, since this library is employed by all non-trivially parallelized HPC applications. Naturally, some currently identified software may not be directly available on a prototype system; for example Intel MPI or MKL identified in Figure 4 will not be available on an ARM prototype. But the survey result indicates that the availability of an efficient alternative will be crucial in the evaluation of the prototype. An additional, rather non-expected response, was the popularity of the requirement of a Python stack, which exceeds the responses for an OpenMP implementation.

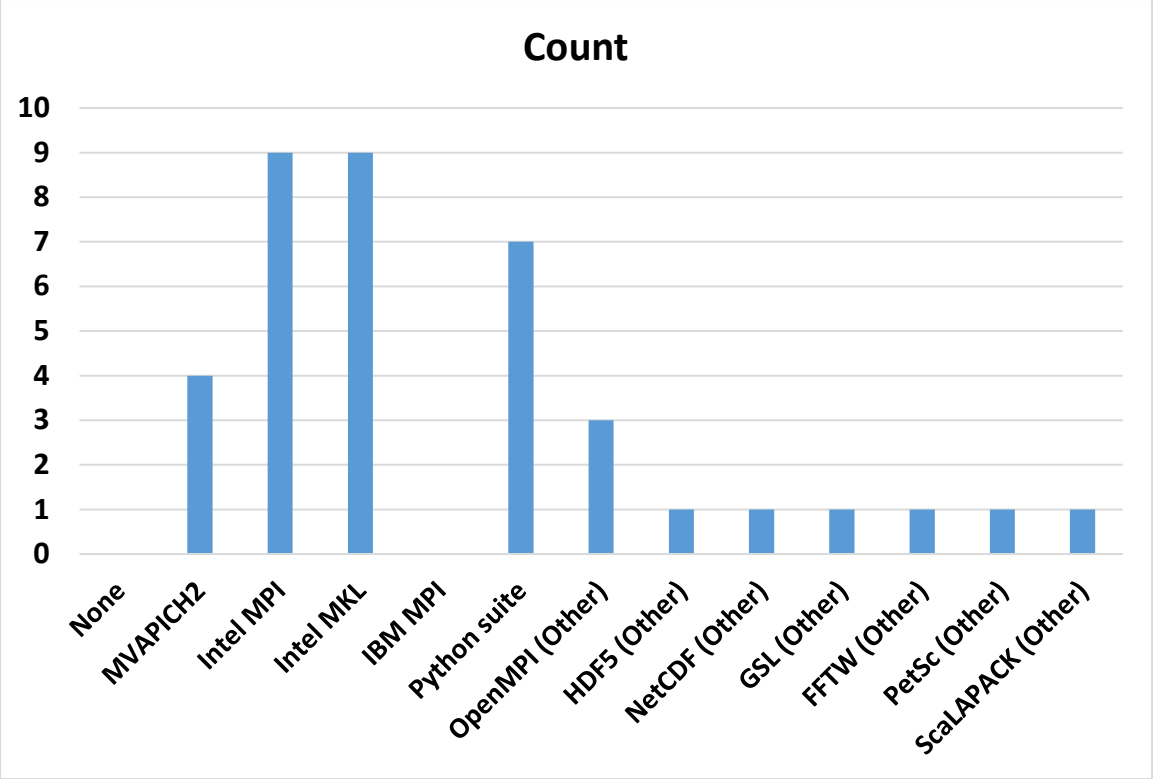

Figure 4 The minimum requirements of PRACE Tier-0/Tier-1 sites in terms of application libraries (e.g. MPI, Math libraries, etc.) that should be installed on a system before it gets accessed by general HPC users

4.4 Development tools

Many HPC software packages come either in the form of source code or are directly developed by local user communities. In either case, the availability of optimized compilers and debugging tools are crucial for a prototype since these determine the possibility of performing comparative checks with existing solutions. The dominant position of x86 platforms in HPC is exemplified by the strong requirement on the availability of Intel Cluster Studio, shown in Figure 5. In many cases, such as other CPU architectures or accelerator solutions, this response should be interpreted as a requirement for an optimized and mature compiler stack for the prototype. This conclusion is supported by the responses to the requirement of a GNU toolchain, also shown in Figure 5.

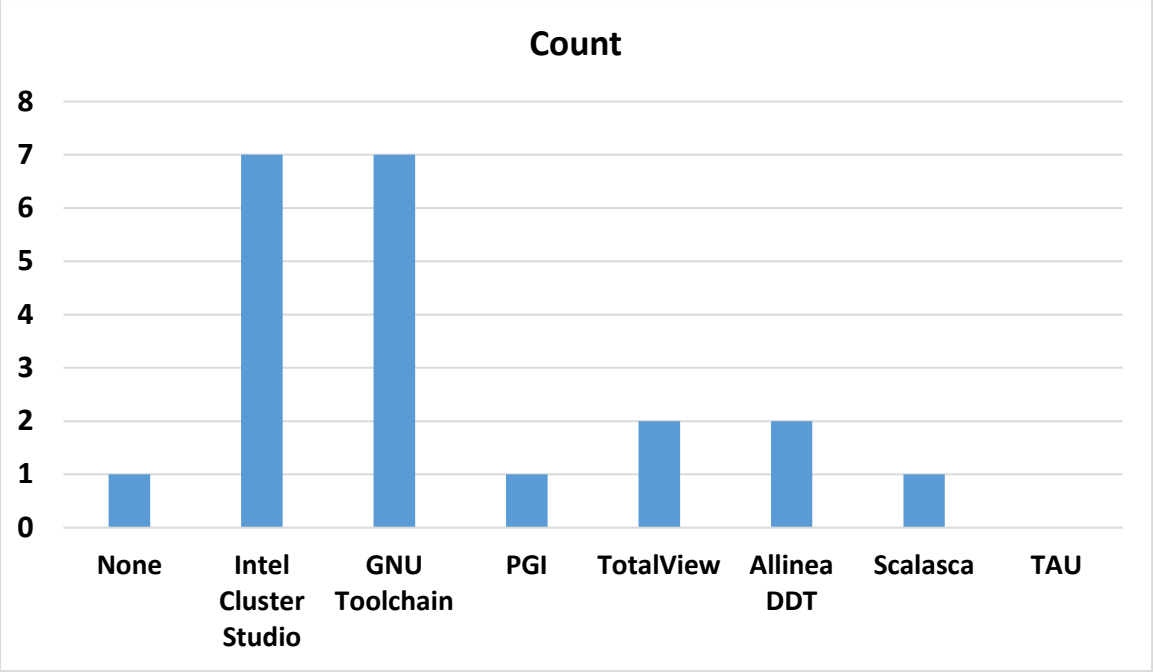

Figure 5 The minimum requirements of PRACE Tier-0/Tier-1 sites in terms of development tools (e.g. compilers, debugging tools, etc.) that should be installed on a system before it gets accessed by general HPC users

4.5 Performance monitoring tools

In many cases, a prototype is specifically deployed in order to check or compare specific performance metrics against currently used technologies. In such cases, having the same software stack available for measuring the performance of the code, as well as the response of the hardware to system utilization makes comparisons with established systems easier or, in some cases, is necessary altogether for any meaningful comparison. When designing a prototype, one should, therefore, try and maintain the same performance analysis and monitoring tools. In this respect, IPMI and PAPI appear to be the most established tools for monitoring and measurement and this is reflected in the survey responses shown in Figure 6.

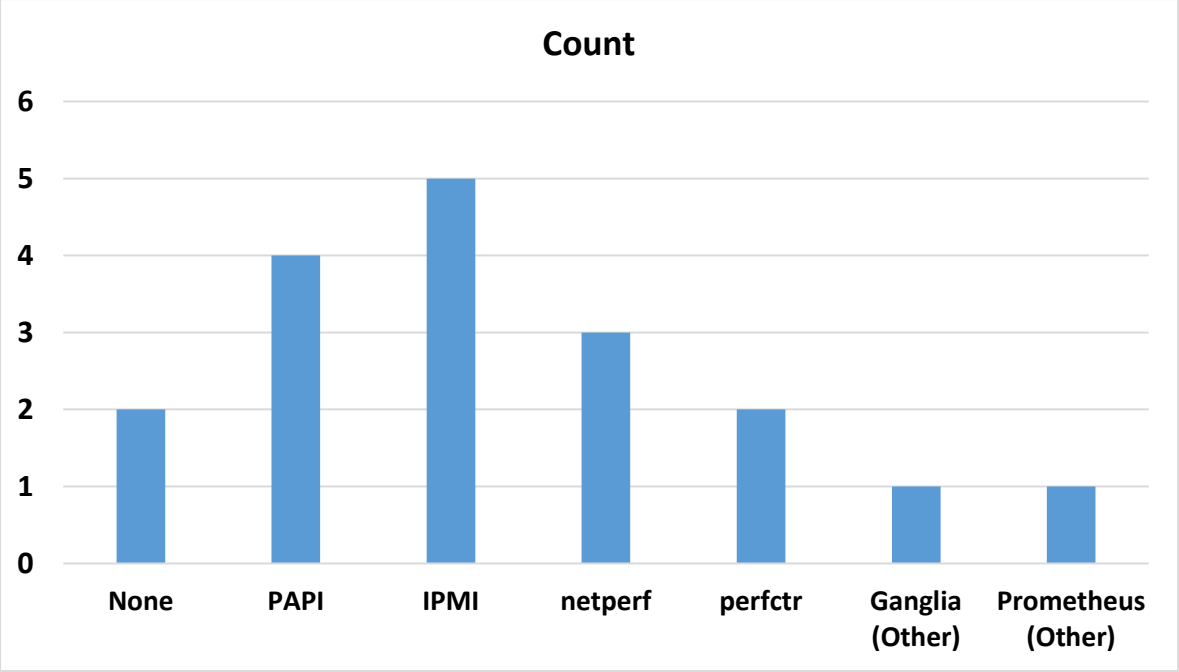

Figure 6 The minimum requirements of PRACE Tier-0/Tier-1 sites in terms of performance monitoring tools that should be installed on a system before it gets accessed by general HPC users

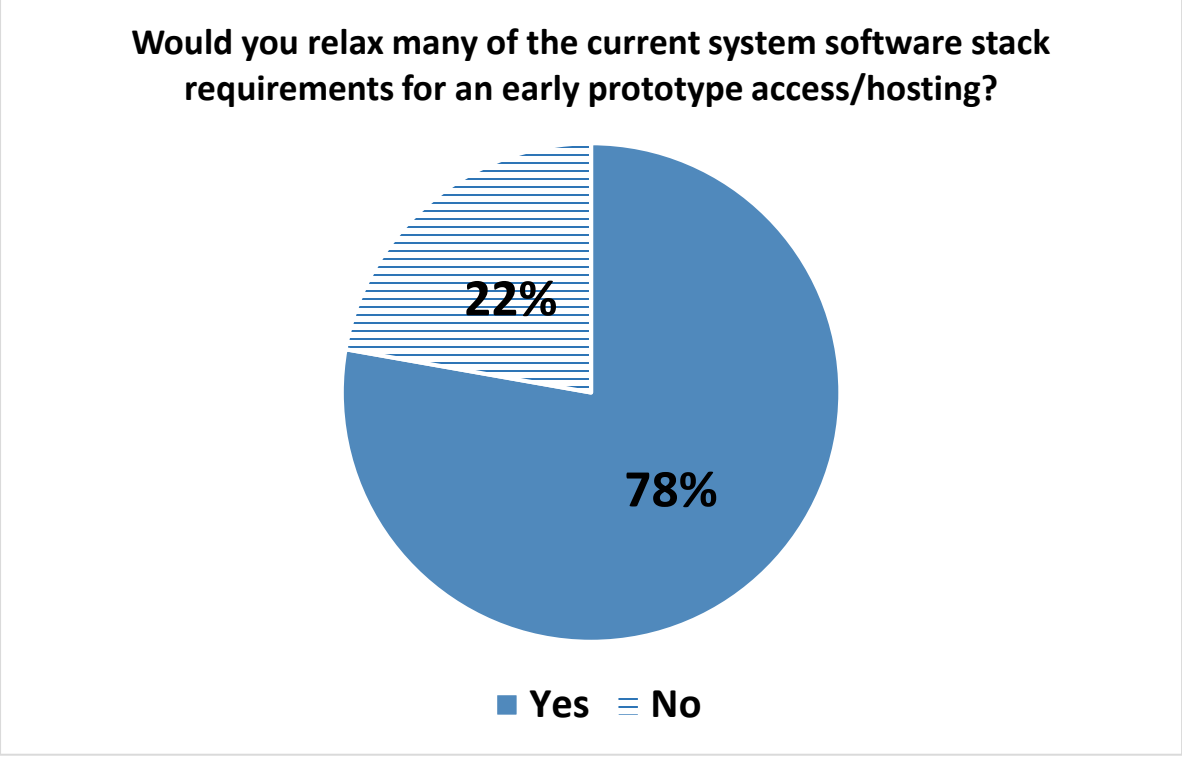

Figure 7 Relaxation policy of system software stack requirements at PRACE Tier-0/Tier-1 sites

Despite the fact that the prototype should resemble a production environment, this may not be possible in all cases for all aspects of the final product, or may not be required at all. Depending on what the purpose of the prototype is, a majority of answers, shown in Figure 7, suggest that some of the software requirements may be relaxed for prototypes, or even omitted. A good example of this may be the resource management system: if a prototype is being used only by a single team or person, access to the machine does not need to be managed. Prototypes may represent different Technology Readiness Levels (TRL) which should also be taken into account as should be the expectation that the pre-production sample hardware will have the same quality as the final product. Depending on the nature of the prototype, one must check what the subset of the production machine functionality should be in order to extract comparable results and what compromises may be permitted that would help to go through the test without impairing the results.

5 Description of benchmarks used for prototype evaluation

This section lists the details of the benchmarks used by PRACE Tier-0 and Tier-1 sites for evaluating the deployed prototype HPC systems before general users access them. The presented benchmarks cover a wide spectrum, ranging from micro-benchmarks (e.g. IOR, STREAM) over synthetic benchmarks (e.g. HPCG) to fully-fledged application codes (e.g. DL_POLY).

Responses to the survey conducted at Tier-0 and Tier-1 sites are given in the bar chart of Figure 8. This section also discusses the usage of the Unified European Application Benchmark Suite (UEABS) in prototype evaluation. UEABS is a collection of application benchmarks maintained by PRACE and listed in Figure 8 as a single benchmark. The detailed usage of single benchmarks within the UEABS is depicted in Figure 9.

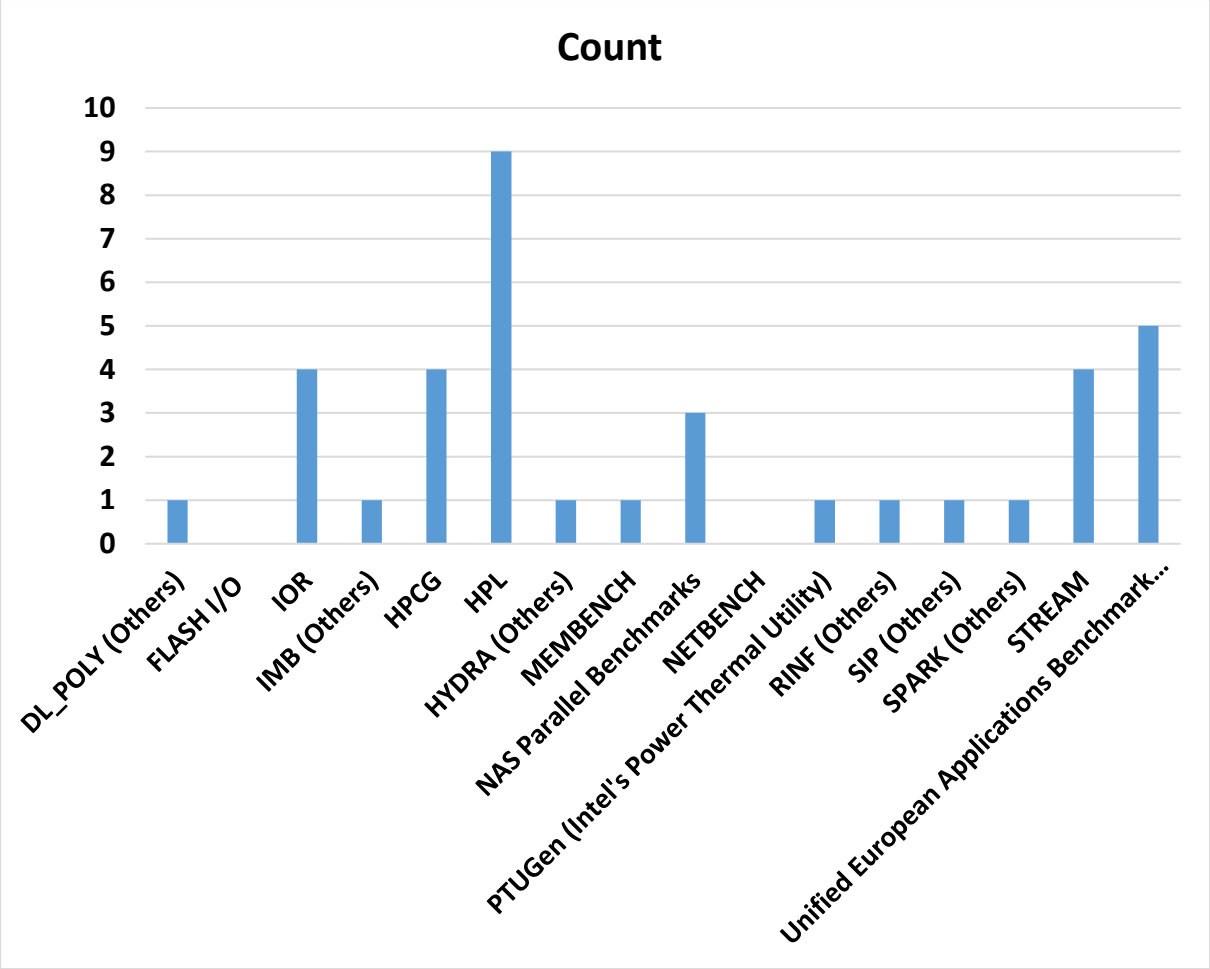

Figure 8 Usage of different benchmarks for prototype evaluation at PRACE Tier-0/Tier-1 HPC sites

The benchmarks listed in Figure 8 cover all main components of interest: floating point efficiency, memory bandwidth, performance of storage subsystems, and network performance. Table 2 briefly describes each of the benchmarks identified in Figure 8 - the components of the prototype it tests and provides corresponding references therein for further details.

Benchmark	Description
DL_POLY	DL_POLY is a general-purpose molecular dynamics simulation package developed at Daresbury Laboratory and can be executed as a serial or parallel application. It achieves parallelization using the replicated data strategy, which is suitable for homogeneous, distributed-memory, parallel computing systems [23].
FLASH I/O	FLASH I/O measures the performance of the FLASH parallel HDF5 (Hierarchical Data Format) output for tuning the I/O performance in a controlled environment. The benchmark recreates the primary data structures in FLASH and produces checkpoint and plot files. Since the

	I/O routines are similar to the routines used by FLASH, any performance improvements that are done with the benchmark program will be shared by FLASH [24].
IOR ***(Interleaved Or Random)***	IOR measures the I/O performance of parallel file systems at both the POSIX and MPI-IO level. Parameters such as the overall I/O size, individual transfer size, file access mode (single shared file, one file per client), data access pattern (sequential or random) are considered as input arguments and thus can be varied [25].
IMB ***(Intel MPI Benchmarks)***	IMB is a suite of benchmarks that perform various MPI performance measurements for point-to-point and global communication operations for a range of message sizes. The generated benchmark data allow evaluating: (i) the performance of the underlying high-end system, including the node performance, network latency, and throughput; and (ii) the efficiency of the used MPI implementation [26].
HPCG (High Performance Conjugate Gradients)	HPCG is a synthetic benchmark which generates and solves a 3D sparse linear system using a local symmetric Gauss-Seidel preconditioned conjugate gradient method. The computations and data access patterns are similar to commonly-used, real-world scientific applications, thus providing a good measure of application performance [27].
HPL (High Performance LINPACK)	HPL is used to rank systems in the TOP500 [28] list of the fastest supercomputers in the world. It solves a dense system of linear equations and shows the measure of achieved performance on a given system. The system of linear equations is represented as a matrix divided into small pieces, referred to as tiles, which are distributed across the processors of the compute nodes of the system [29].
HYDRA	HYDRA is a distributed HTTP benchmark tool capable of simulating myriads of agents sending rapid requests to a given server [30].
MEMBENCH	MEMBENCH measures memory bandwidth versus message size for unit and random stride cases [31].
NAS PARALLEL BENCHMARKS	This set of benchmarks targets performance evaluation of highly parallel supercomputers. They are developed and maintained by the NASA Advanced Supercomputing (NAS) Division (formerly the NASA Numerical Aerodynamic Simulation Program) [32].
NETBENCH	NETBENCH is used for measuring interconnect latency and bandwidth [31].
PTUGen (Intel's Power Thermal Utility)	This benchmark is developed by Intel for generating TDP (Thermal Design Power) like workloads [33].
RINF	RINF is used for testing the floating point and integer performance of various one-dimensional loop kernels. Depending on the access pattern, various aspects of the processor architecture and the memory hierarchy are tested (BADW-LRZ internal development).
SIP	SIP is a multi-threaded Strongly-Implicit Procedure (SIP) solver according to Stone [34], suitable for solving systems of linear equations resulting from a discretisation of partial differential equations. The iterative solver consists of an Incomplete LU (ILU)

	decomposition and a series of forward and backward substitutions. It is widely used in fluid mechanics and therefore is of practical importance [35].
SPARK	Benchmark developed by IBM research which covers a wide range of HPC, SQL, machine learning, etc. applications [36].
STREAM	Synthetic application-benchmark for measuring the sustainable memory bandwidth and the corresponding computation rate for vector kernels [37].
Unified European Applications Benchmark Suite (UEABS)	A set of fourteen application codes[3] taken from the pre-existing PRACE and DEISA application benchmark suites for forming a single set of scalable, currently relevant, and maintainable benchmarks which can be run on large-scale high-end systems [38].

Table 2 Description of benchmarks used for prototype evaluation by PRACE TIer-0/Tier-1 HPC sites

Five out of nine PRACE Tier-0/Tier-1 HPC sites indicated to use UEABS for prototype evaluation, namely: *CINECA; CSC; BAdW-LRZ; GRNET; and PSNC*. The UEABS include, beyond up-to-date benchmark kernels and instructions on how to run them, datasets and input parameters typical of state-of-the-art application use cases and example benchmark results reported on PRACE Tier-0 systems. The PRACE PCP project also used a subset of UEABS, namely NEMO, Quantum Espresso, BQCD and SPECFEM3D, along with the classical HPL (High Performance LINPACK) benchmark used to asses performance of a dedicated system for solving a dense system of linear equations, providing a standardized Floating Point Operations Per Second (FLOPS) value.

It should be noted that although some of the UEABS benchmarks include support for accelerators, the inclusion of accelerated benchmark reports and instructions, namely benchmarks appropriate for GPUs and for Intel Xeon Phi, has been carried out during PRACE-4IP [39] and were not available during the timeframe the survey was conducted. The work on inclusion of accelerated benchmark reports and instructions is continued in PRACE-5IP and will become available in PRACE-5IP D7.5 expected by February 2019.

Figure 9 shows the usage of individual benchmarks included in UEABS by these 5 supercomputing sites (This data can also be found in the previously mentioned PRACE repository: https://repository.prace-ri.eu/git/hayk.shoukourian/5IPT3.git). The benchmarks marked with "*" sign were not used in the survey due to the above-mentioned availability.

Table 3 presents a brief description of each individual benchmark included in UEABS. Full details of these benchmarks are available via the UEABS repository [38].

[3] Initially twelve but became fourteen after merging back the PRACE accelerator benchmark suite.

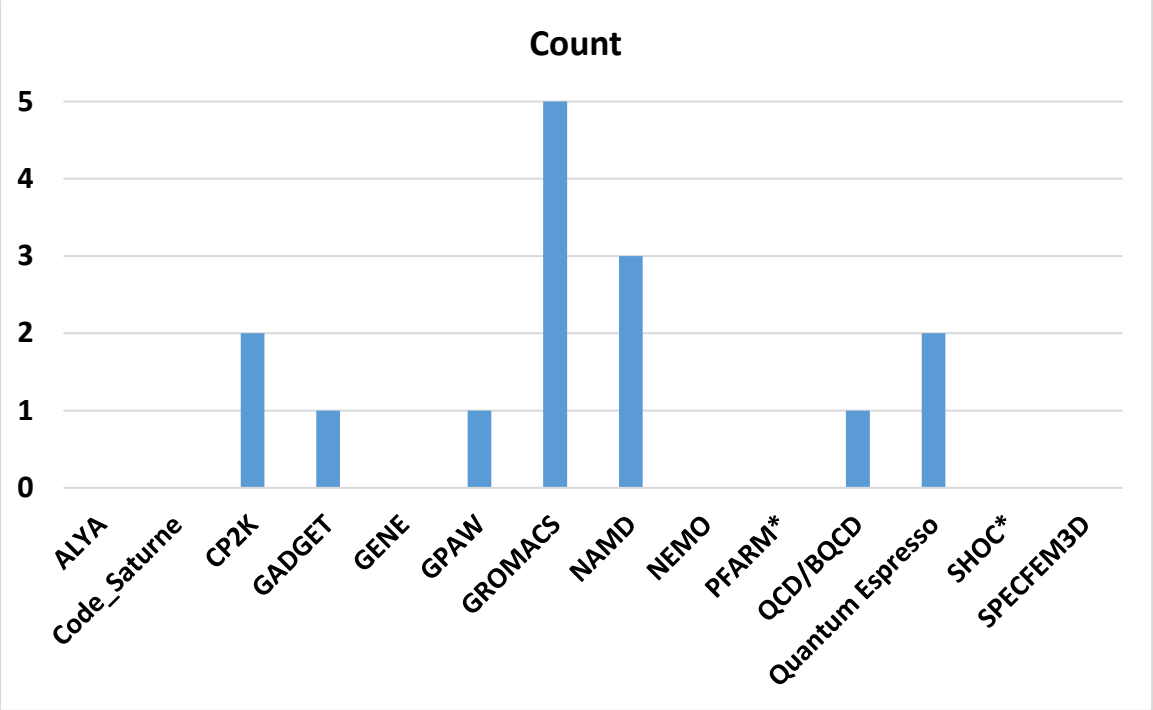

Figure 9 Usage of different benchmarks within UEABS for prototype evaluation at PRACE Tier-0/Tier-1 HPC sites

Benchmark	Description
ALYA	The Alya System is a Computational Mechanics code capable of solving different physics (convection-diffusion reactions, turbulence, bi-phasic flows and free surface, thermal flow, quantum mechanics and solid mechanics, etc.), each one with its own modelization characteristics, in a coupled way. ALYA is written in Fortran 90/95 and parallelized using MPI and OpenMP.
Code_Saturne	Code_Saturne is an open-source, multipurpose Computational Fluid Dynamics (CFD) software package. It was originally designed for industrial applications and research activities in several fields related to energy production. Code_Saturn is based on a co-located finite volume approach that can handle three-dimensional meshes built with any type of cell (tetrahedral, hexahedral, prismatic, pyramidal, polyhedral) and with any type of grid structure (unstructured, block structured, hybrid). The code is able to simulate either incompressible or compressible flows, with or without heat transfer, and has a variety of models to account for turbulence. Dedicated modules are available for specific physics such as radiative heat transfer, combustion (e.g. with gas, coal and heavy fuel oil), magneto-hydro-dynamics, and compressible flows, and two-phase flows. The software comprises of around 350,000 lines of source code, with about 37% written in Fortran90, 50% in C and 15% in Python. The code is parallelized using MPI and OpenMP paradigms.

CP2K	CP2K is a freely available (GPL) program to perform atomistic and molecular simulations of solid state, liquid, molecular, and biological systems. It provides a general framework for different methods such as e.g. density functional theory (DFT) using a mixed Gaussian and plane waves approach (GPW), and classical pair and many-body potentials. It is well written, standards-conforming to Fortran 95, parallelized with MPI, and in some parts with hybrid OpenMP+MPI as an option.
GADGET	GADGET is a freely available code for cosmological N-body/SPH simulations on massively parallel computers with distributed memory. GADGET is written in C and uses an explicit communication model that is implemented with the standardized MPI communication interface. The code can be run on essentially all supercomputer systems presently in use, including clusters of workstations or individual PCs.
GENE	GENE is a gyro-kinetic plasma turbulence code and is highly scalable. The code is written in Fortran 90 and C and is parallelized with pure MPI. It strongly relies on a Fast Fourier Transform library and has built-in support for FFTW, MKL or ESSL. It also uses LAPACK and ScaLAPACK routines for LU decomposition and solution of a linear system of equations of moderate size.
GPAW	GPAW is a program package for electronic structure calculations based on the density functional theory (DFT) and the time-dependent density functional theory (TD-DFT). The density-functional theory allows studies of ground state properties such as energetics and equilibrium geometries, while the time-dependent density functional theory can be used for calculating excited state properties such as optical spectra. The program package includes two complementary implementations of time-dependent density functional theory: a linear response formalism and a time-propagation in real time. The program offers several parallelization levels. The most basic parallelization strategy is domain decomposition over the real-space grid. In magnetic systems, it is possible to parallelize over spin, and in systems that have k-points (surfaces or bulk systems) parallelization over k-points is also possible. Furthermore, parallelization over electronic states is possible in DFT and in real-time TD-DFT calculations. GPAW is written in Python and C and parallelized with MPI.
GROMACS	GROMACS is a versatile package to perform molecular dynamics, i.e. simulate the Newtonian equations of motion for systems with hundreds to millions of particles. It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a lot of complicated bonded interactions, but since GROMACS is extremely fast at calculating the non-bonded interactions (that usually dominate simulations) many groups are also using it for research on non-biological systems, e.g. polymers.

NAMD	NAMD is a widely used molecular dynamics application designed to simulate bio-molecular systems on a wide variety of compute platforms. A NAMD license can be applied for on the developer's website free of charge. Once the license has been obtained, binaries for a number of platforms and the source can be downloaded from the website. Deployment areas of NAMD include pharmaceutical research by academic and industrial users. NAMD is written in C++ and parallelized using Charm++ parallel objects, which are implemented on top of MPI.
NEMO	NEMO (Nucleus for European Modeling of the Ocean) is a state-of-the-art modeling framework for oceanographic research, operational oceanography seasonal forecast and climate studies. NEMO is used by a large community: 240 projects in 27 countries (14 in Europe, 13 elsewhere) and 350 registered users (numbers for the year 2008). The code is available under the CeCILL license (public license). The latest stable version is 3.6. NEMO is written in Fortran90 and parallelized with MPI.
PFARM	PFARM is part of a suite of programs based on the 'R-matrix' ab-initio approach to the variational solution of the many-electron Schrödinger equation for electron-atom and electron-ion scattering.
QCD/BQCD	The QCD benchmark is, unlike the other benchmarks in the PRACE application benchmark suite, not a full application but a set of 5 kernels which are representative of some of the most compute-intensive parts of QCD calculations.
Quantum Espresso	QUANTUM ESPRESSO is an integrated suite of computer codes for electronic-structure calculations and materials modeling, based on density-functional theory, plane waves, and pseudopotentials (norm-conserving, ultrasoft, and projector-augmented wave). It is freely available to researchers around the world under the terms of the GNU General Public License. QUANTUM ESPRESSO is evolving towards a distribution of independent and inter-operable codes in the spirit of an open-source project, where researchers active in the field of electronic-structure calculations are encouraged to participate in the project by contributing their own codes or by implementing their own ideas into existing codes. QUANTUM ESPRESSO is written mostly in Fortran90 and parallelised using MPI and OpenMP.
SHOC	The Scalable HeterOgeneous Computing (SHOC) benchmark suite is a collection of benchmark programs testing the performance and stability of systems using computing devices with non-traditional architectures for general purpose computing. Its initial focus is on systems containing Graphics Processing Units (GPUs) and multi-core processors, and on the OpenCL programming standard. It can be used on clusters as well as individual hosts. Also, SHOC includes an Offload branch for the benchmarks that can be used to evaluate the Intel Xeon Phi x100 family.

	SHOC is written in C++ and is MPI-based. Offloading for accelerators is implemented through CUDA and OpenCL for GPUs.
SPECFEM3D	SPECFEM3D simulates three-dimensional global and regional seismic wave propagation based upon the spectral-element method (SEM). All SPECFEM3D_GLOBE software is written in Fortran90 with full portability in mind and conforms strictly to the Fortran95 standard. It uses no obsolete or obsolescent features of Fortran77. The package uses parallel programming based upon the Message Passing Interface (MPI).

Table 3 Short description of benchmarks within UEABS [38]

As Big Data and Machine-Learning (ML) techniques become more and more prevalent in a broad range of domains (e.g. IoT, medical/health care, autonomous driving, etc.), computational centres are identifying needs of their users that require a mixture of both traditional and optimized for data-intensive workloads HPC hardware. It is therefore equally important to consider benchmarks tailored towards performance evaluation of applications utilizing Big Data and AI/ML techniques. Although these benchmarks were not explicitly indicated in the survey results, for the purposes of this deliverable, a list of such benchmarks is compiled and suggested in Table 4, outlining some important Big Data and AI/ML specific benchmarks that can be used for future HPC prototype evaluations. Some of these benchmarks, for instance, were used in the procurement process of the AI Bridging Cloud Infrastructure (ABCI) supercomputer [40], which provides cloud access to compute and storage for artificial intelligence and data analytics workloads. The ABCI system, commissioned by the National Institute of Advanced Industrial Science and Technology (AIST) in Japan, started its operation in August 2018 and delivers up to 550 PetaFLOPS of AI processing power (half precision) and up to 37 PetaFLOPS with double precision, and has a projected annual average system PUE of less than 1.1 [41].

Benchmark	Target Domain	Description
TPC Benchmarks	***Big Data***	Transaction Processing Performance Council (TPC) [42] is a non-profit consortium of various IT companies that aims to evaluate the performance of transaction processing and database systems. In contrast to the majority of stand-alone benchmarks, the TPC benchmarks are designed after actual production applications and environments. TPC benchmarks cover a wide spectrum of areas including online-transaction-processing.
BigBench	***Big Data***	This benchmark addresses the three V's of Big Data systems by presenting a data model that simulates volume, velocity, and variety characteristics of a system via the help of a synthetic data generator for structured, semi-structured, and unstructured data [43]. The current implementation for a Hadoop based environment can be found in [44] and allows for two different

		execution modes: 1) using a driver for simple and complete execution of the benchmarks; or 2) using bash scripts that allow for execution of certain atomic tasks.
BigDataBench	***Big Data & AI/ML***	An open source big data and AI benchmark suite [45] with the current version (4.0) providing 13 representative real-world data sets and 47 benchmarks [46]. The benchmarks represent seven workload types: 1) AI; 2) online services; 3) offline services; 4) graph analytics; 5) data warehouse; 6) NoSQL; and 7) streaming covering applications domains such are search engines, social networks, e-commerce, multimedia processing, and bioinformatics.
BigFrame	***Big Data***	A benchmark generator capable of generating various benchmarks tailored to a given set of data and workload requirements used in big data analytics [47]. BigFrame allows users to generate certain benchmarks tailored to their specific needs (e.g. data variety, data volume, etc.).
Graph500	***Big Data***	Graph500 is a rating of supercomputer systems, focused on data-intensive workloads [48]. The aim of the benchmark is to have multiple kernels accessing a single data structure (representing an undirected graph). There are three timed kernels used by the benchmark: the first one constructs an undirected graph; the second performs a breadth-first search on the constructed graph; and the third kernel performs multiple single-source shortest path computations on that graph.
AMP Lab Big Data Benchmark (Berkeley Big Data Benchmark)	***Big Data***	The benchmark measures the response time on various queries using different data sizes by utilizing operations like scan, aggregation, join, or complex User Defined Functions (UDFs) [49]. This benchmark, currently, provides quantitative and qualitative comparisons of five data warehouse systems: Redshift; Hive; Shark; Impala; and Stinger/Tez. The benchmark supports scaling to “thousands of nodes” [48].
HiBench	***Big Data & AI/ML***	A big data benchmark suite aimed at evaluating various big data frameworks in terms of throughput, speed, and system resource utilization [50]. The benchmark contains in total a set of 19 Hadoop, Spark and streaming workloads divided into 6 categories: 1) micro (containing sort, TeraSort, WordCount, etc.); 2) Machine-Learning (containing Bayesian Classification, K-means Clustering, Gradient Boosting Trees, Alternating Least Squares, etc.); 3) SQL (containing scan, join, aggregate workloads); 4) web search (containing PageRank and Nutch indexing workloads); 5) graph (containing NWeight algorithm); and 6) streaming (containing Identity, Fixwindow, etc.

		workloads).
CloudSuite	***Big Data***	A benchmark suite for cloud services [51]. This benchmark suite covers a broad spectrum of applications including data analytics, data serving, media streaming, large-scale and computation-intensive tasks, web search, graph analytics, and data caching. It also includes benchmarks that perform extensive data usage with tight latency constraints (e.g. real-time video streaming, etc.).
Yahoo! Cloud Serving Benchmark (YCSB)	***Big Data***	An open-source benchmark designed to benchmark the basic operations (as insert, update, read, delete, etc.) for major No-SQL key-value database systems such are HBase, Hypertable, Cassandra, MongoDB, etc. [52].
Standard Performance Evaluation Corporation (SPEC)	***Big Data***	SPEC is a non-profit corporation aimed at establishing and maintaining standardized benchmarks and tools for performance and energy-efficiency evaluation of high-end compute systems [53]. SPEC has set up a Big Data Working Group that will further improve research in methodologies for Big Data system benchmarking [54].
Low-precision General Matrix Multiplication (GEMM)	***AI/ML***	A matrix multiplication algorithm that takes n^3 multiply-accumulate instructions for $n \times n$ sized matrices. The term "low-precision" indicates that the input and output matrix entries are integers of at most 8 bits. The scalar type is uint8_t. To avoid possible overflow, the results of more than 8 bits are usually internally accumulated and at the end, only the significant 8 bits are kept [55].
DeepBench	***AI/ML***	Open source benchmarking tool measuring the performance of basic operations that are important in training deep neural networks. It uses different neural network libraries, such as NVIDIA's cuDNN and Intel's MKL, to benchmark these basic operations on different hardware platforms [56].
MLPerf	***AI/ML***	Benchmark suite aimed at measuring the performance of software and hardware tailored towards applications relying on machine learning techniques [57]. MLPerf aims to provide a set of benchmarks that would allow measuring the performance of a given system for both training and inference. The benchmark suite is still under development[4] and currently provides seven benchmarks for: 1) image classification; 2) object detection; 3) speech recognition; 4) translation; 5) recommendation; 6)

[4] At the time of this best practice guide's publication.

		sentiment analysis; and 7) reinforcement learning [58].

Table 4 Short description of some emerging Big Data and AI/ML specific benchmarks

Additionally, researchers from ETH Zurich have recently developed a benchmark suite, referred to as Deep500 [59, 60], for the assessment of deep learning capabilities of a given HPC system. According to ETH researchers, Deep500 is a distributed and reproducible benchmarking suite freely available on GitHub [61] that offers: *(i)* customizability, i.e. allows to benchmark a combination of different deep-learning codes; *(ii)* detailed execution analysis capability by providing a rich set of evaluation metrics; and *(iii*) validation of convergence, correctness, accuracy, and performance [62].

6 Main KPIs used for prototype evaluation

Survey participants from PRACE Tier-0/Tier-1 HPC centers were also asked to indicate the main metrics/ Key Performance Indicators (KPIs) that their site relies upon when evaluating prototype systems. Figure 10 presents this data.

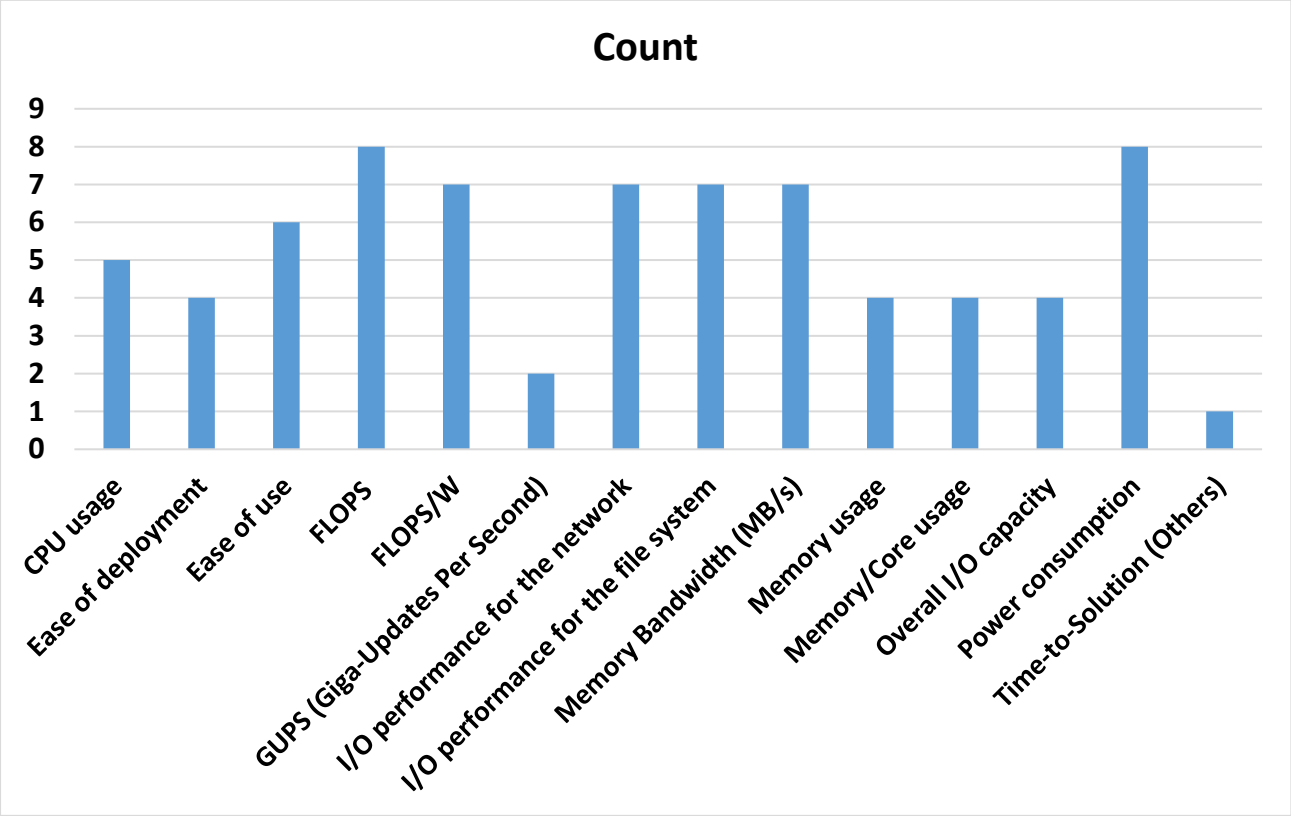

Figure 10 Usage of different KPIs for prototype evaluation at PRACE Tier-0/Tier-1 HPC sites

The survey results show that Floating Point performance and power consumption are the most popular metrics when evaluating prototypes. All but one respondent has reported these two metrics.

Other metrics, which are used widely among respondents (7 respondents out of 9), include the ratio of FLOPS per Watt and I/O related KPIs (network, filesystem, and memory subsystem bandwidth ranked equally). A limited number of participants reported metrics that are either harder to uniquely quantify, such as ease of deployment (4 out of 9 respondents) or problem-specific (4 out of 9: memory usage, memory/core usage; 5 out of 9: CPU usage).

7 Conclusions

High Performance Computing (HPC) prototyping is an important activity assisting the evaluation of new design concepts and technologies that aim to address the functionality shortages present in the existing state of the art solutions. However, the procedures concerning its planning, commissioning, and evaluation are not straightforward. This document provided a guideline for HPC prototype and/or demonstrator owners, based on previously assessed user requirements and prototyping activities done by PRACE Tier-0/Tier-1 HPC sites. More specifically, this document extended the previous PRACE-4IP WP5 efforts in developing a best practice guide for prototype planning and evaluation with:

a) a checklist for system and development tools that should be installed on a system before it gets accessed by general HPC users;
b) sets of benchmarks (including synthetic benchmarks) prepared in cooperation with PRACE-5IP WP7, the application-focused work package, which among other activities is in charge of code enabling activities, publication of Best Practice Guides and the development of the Unified European Applications Benchmark Suite (UEABS); and
c) Key Performance Indicators, as identified by PRACE Tier-0/Tier-1 HPC sites, that are most commonly used for prototype evaluation.

Some of the analysis presented in this document is based on the results of online survey that was distributed among 9 PRACE Tier-0/Tier-1 HPC sites. The most important points to note are:

- in prototyping environments, most sites identified job schedulers/resource managers and the availability of common HPC libraries such as MPI as the most important components to have available on the prototype;
- some of the currently existing software requirements may be relaxed for prototypes or even omitted to enable the early access;
- the adoption of UEABS as a benchmark by prototype evaluators is currently at ~50%, i.e. 4 out of 9 respondents stated they used at least one UEABS kernel during their prototyping. Furthermore, all respondents use HPL during their prototype evaluation, and all respondents complement UEABS (when used) and HPL with additional micro-benchmarks such as STREAM, HPCG, or IOR;
- in terms of Key Performance Indicators, the most popular metrics used for HPC prototype evaluation are those that can be unambiguously defined. Examples include FLOPS, FLOPS/W, total power consumption, bandwidth to file system, memory, and network. Conversely, metrics that are more difficult to define unambiguously or that cannot be easily determined quantitatively are less frequently used. Examples here include ease of system deployment, CPU usage, memory usage, and others.