E-Infrastructures
H2020-EINFRA-2014-2015

EINFRA-4-2014: Pan-European High Performance Computing Infrastructure and Services

PRACE-4IP

PRACE Fourth Implementation Phase Project

Grant Agreement Number: EINFRA-653838

D6.4

Deployment of Prototypal New Services

Final

Version: 1.0
Author(s): Luigi Calori, CINECA; Marcin Krotkiewski, UiO; Miroslaw Kupczyk, PSNC; Felip Moll, BSC; Janez Povh, ULFME
Date: 17.04.2017

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: EINFRA-653838**	
	Project Title: Deployment of Prototypal New Services	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: < **D6.4**>	
	Deliverable Nature: < Report >	
	Dissemination Level: PU*	**Contractual Date of Delivery:** 30 / April / 2017
		Actual Date of Delivery: 30 / April / 2017
	EC Project Officer: Leonardo Flores Añover	

* - The dissemination level are indicated as follows: **PU** – Public, **CO** – Confidential, only for members of the consortium (including the Commission Services) **CL** – Classified, as referred to in Commission Decision 2991/844/EC.

Document Control Sheet

Document	**Title: PRACE-4IP**	
	ID: D6.4	
	Version: <1.0 >	**Status:** ***Final***
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2010	
	File(s): D6.4.docx	
Authorship	**Written by:**	Luigi Calori, CINECA; Marcin Krotkiewski, UiO; Miroslaw Kupczyk, PSNC; Felip Moll, BSC; Janez Povh, ULFME
	Contributors:	Damian Kaliszan, PSNC Huub Stoffers, Surfsara Nial Wilson, ICHEC Oguzhan Herkiloglu, UHEM Fabio Hernandez, IN2P3/CNRS; David Hrbac, IT4I-VSB Nevena Ilieva, NCSA; Thomas Bönisch, HLRS Zoltan Kiss, NIIF Leon Kos, UL Kriakos Gkinis, GRNET Andrew Turner, EPCC Ioannis Liabotis, GRNET Andreas Panteli, CaSToRC Giovanni Erbacci, CINECA
	Reviewed by:	Stephen Booth, EPCC Florian Berberich, FZJ
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	17/March/2017	Draft 1	Adapted template for the contributors
0.2	30/March/2017	Draft 2	Merged all contributions and send for improvements of contributors
0.3	3/April/2017	Draft 3	Included all updates
0.4	3/April/2017	Draft 4	Submitted for internal review
0.5	14/April/2017	Draft 5	Revision
1.0	17/April/2017	Final	Sent for approval to PMO

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, New services, Urgent computing, Large-scale scientific instruments, In-situ visualisation, Repositories, Open source scientific libraries

Table of Contents

List of Figures

List of Tables

References and Applicable Documents

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

[2] J. Povh et al., Analysis of New Services, PRACE 4IP deliverable D6.3, 2015

[3] F. Hernandez, Revisiting bulk data transport over HTTP, PRACE 4IP white paper, available at www.prace-project.eu, 2017

[4] M. Kupczyk, D. Kaliszan, H. Stoffers, N. Wilson , F. Moll, Urgent Computing service in PRACE Research Infrastructure, PRACE 4IP white paper, available at www.prace-project.eu, 2017

[5] M. Krotkiewski, Dealing with small files in HPC environments: automatic loop-back mounting of disk images, PRACE 4IP white paper, available at www.prace-project.eu, 2017

[6] M. Krotkiewski, A. Panteli, Portable and flexible framework for in-memory data packaging and transfer, PRACE 4IP white paper, available at www.prace-project.eu, 2017

[7] N. Ilieva, Z. Kiss, B. Pavlov, G. Szigeti, Using gsatellite for data intensive procedures between large-scale scientific instruments and HPC, PRACE 4IP white paper, available at www.prace-project.eu, 2017

[8] N. Ilieva, B. Pavlov, P. Petkov, and L. Litov, GEANT4 on Large Intel Xeon Phi Clusters: Service Implementation and Performance, PRACE 4IP white paper, available at www.prace-project.eu, 2017

[9] L. Calori, R. Mucci, F. Stanek: Deploying remote visualization services in HPC environment: recipes for an Open Source software stack, PRACE 4IP white paper, available at www.prace-project.eu, 2017

[10] Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops, G. Min, B. Di Martino, L. Yang, M. Guo, G. Ruenger (Eds.) (Springer, Berlin Heidelberg, 2006) ISSN 0302-9743.

[11] See https://twiki.cern.ch/twiki/bin/view/Geant4/MultiThreadingTaskForce, A. Dotti at al.

[12] Camerlengo, Ozer HG, Onti-Srinivasan R, Yan P, Huang T, Parvin J, Huang K., From sequencer to supercomputer: an automatic pipeline for managing and processing next

generation sequencing data., AMIA Jt Summits Transl Sci Proc. 2012;2012:1-10. Epub 2012 Mar 19.

[13] Kawalia A, Motameny S, Wonczak S, Thiele H, Nieroda L, et al. (2015) Leveraging the Power of High Performance Computing for Next Generation Sequencing Data Analysis: Tricks and Twists from a High Throughput Exome Workflow. PLoS ONE 10(5): e0126321. doi: 10.1371/journal.pone.0126321

[14] SPRUCE resources, http://spruce.teragrid.org

[15] S.H. Leong, A. Frank, D. Kranzlmüller, Towards a general definition of Urgent Computing, International Conference on Computational Science, 2015

[16] V.V. Krzhizhanovskaya, N.B. Melnikova, A.M. Chirkin, S.V. Ivanov, A.V. Boukhanovsky, P.M.A. Sloot, Distributed simulation of city inundation by coupled surface and subsurface porous flow for urban flood decision support system, International Conference on Computational Science, 2013

[17] S.V. Ivanov, S. V. Kovalchuk, A.V. Boukhanovsky, Workflow-based Collaborative Decision Support for Flood Management Systems, International Conference on Computational Science, 2013

[18] E. Fiori, A. Comellas, L. Molini, N. Rebora, F. Siccardi, D.J. Gochis, S. Tanelli, A. Parodi, Analysis and hindcast simulations of an extreme rainfall event in the Mediterranean area: The Genoa 2011 case, Atmospheric Research, Vol. 138, 2014

[19] B. Balisa, M. Kasztelnikb, M. Bubaka, T. Bartynski, T. Gubała, P. Nowakowski, J. Broekhuijsen, The UrbanFlood Common Information Space for Early Warning Systems, International Conference on Computational Science, 2011

[20] D. Groen, J. Hetherington, H. B. Carver, R. W. Nash, M. O. Bernabeu, P. V. Coveney, Analysing and modelling the performance of the HemeLB lattice-Boltzmann simulation environment, Journal of Computational Science, Vol. 4, Issue 5, 2013

[21] S.H. Leong, A. Frank, D. Kranzlmüller, Leveraging e-Infrastructures for Urgent Computing, International Conference on Computational Science, ICSS 2013

[22] K.K.Yashimoto, D.J.Choi, R.L.Moore, A.Majumdar, E.Hocks, Implementations of Urgent Computing on Production HPC Systems. International Conference on Computational Science, ICSS 2012

[23] K.Kurowski, A.Oleksiak, W.Piątek, J.Węglarz, Impact of urgent computing on resource management policies, schedules and resources utilization. International Conference on Computational Science, ICSS 2012

[24] S.V. Kovalchuk, P. A. Smirnov, S.V. Maryin, T.N. Tchurov, V.A. Karbovskiy, Deadline-driven Resource Management within Urgent Computing Cyberinfrastructure, International Conference on Computational Science, ICSS 2013

[25] K.V. Knyazkov, D.A. Nasonov, T.N. Tchurov, A.V. Boukhanovsky, Interactive Workflow-based Infrastructure for Urgent Computing, International Conference on Computational Science, ICSS 2013

[26] Online announcement of the ICCS Workshop on Urgent Computing, June 2012, on the website http://dice.cyfronet.pl/events/iccs_urgent_computing_workshop

[27] Soden SE, Saunders CJ, Willig LK, Farrow EG, Smith LD, Petrikin JE et al... Effectiveness of exome and genome sequencing guided by acuity of illness for diagnosis of neurodevelopmental disorders. Sci Transl Med. 2014; 6:265ra168

[28] Willig LK, Petrikin JE, Smith LD, Saunders CJ, Thiffault I, Miller NA et al.. Whole-genome sequencing for identification of Mendelian disorders in critically ill infants: a retrospective analysis of diagnostic and clinical findings. Lancet Respir Med. 2015; 5:377-87

[29] Miller NA, Farrow EG, Gibson M, Willig LK, Twist G et al. A 26-hour system of highly sensitive whole genome sequencing for emergency management of genetic diseases. Genome Med. 2015 Sep 30;7(1):100

[30] Mudge S. (Editor), Methods in Environmental Forensics, CRC Press, 2009

[31] K. V. Knyazkov, D. A. Nasonov, T. N. Tchurov, A. V. Boukhanovsky. Interactive workflow-based infrastructure for urgent computing. Procedia Computer Science 18 (2013) 2223

[32] Scientific computing world web magazine: Importance of remote visualization for HPC: http://insidehpc.com/2015/02/hpcs-future-lies-in-remote-visualization/

[33] Nvidia blog VMD-NAMD: GPU based coupling simulation and visualization http://devblogs.nvidia.com/parallelforall/hpc-visualization-nvidia-tesla-gpus/

[34] Nvidia blog In-situ visualization http://devblogs.nvidia.com/parallelforall/interactive-supercomputing-in-situ-visualization-tesla-gpus/

[35] Paraview Home page: http://www.paraview.org/

[36] Visit Home page: https://wci.llnl.gov/simulation/computer-codes/visit/

[37] VMD Visual Molecular Dynamics home page: http://www.ks.uiuc.edu/Research/vmd/

[38] NICE Desktop Cloud Visualization: https://www.nice-software.com/products/dcv

[39] TurboVNC home page: http://www.turbovnc.org/Main/HomePage

[40] VirtualGL project page : http://www.virtualgl.org/

[41] Intel OpenSWR optimized OpenGL emulation http://openswr.org/

[42] CINECA Visualization service: http://www.hpc.cineca.it/services/remote-visualisation

[43] IT4I Anselm Cluster Visualization service: https://docs.it4i.cz/anselm-cluster-documentation/remote-visualization

[44] LRZ SuperMUC visualization service: https://www.lrz.de/services/v2c_en/remote_visualisation_en/overview_en/

[45] web based HTML 5 VNC access to LRZ Visualization service https://www.lrz.de/services/v2c_en/remote_visualisation_en/web_interface_en/

[46] HLRS Cray XC40 Hazel Hen visualization setup: https://wickie.hlrs.de/platforms/index.php/CRAY_XC40_Graphic_Environment

[47] IT4I Salomon Cluster Hardware description: https://docs.it4i.cz/salomon/hardware-overview-1

[48] noVNC Project home page : http://kanaka.github.io/noVNC/

[49] XPRA Project home page : https://xpra.org/

[50] Strudel Home page : https://www.massive.org.au/userguide/cluster-instructions/strudel; Strudel Web : https://www.massive.org.au/userguide/cluster-instructions/strudel-web

[51] Pyinstaller Project home page : http://www.pyinstaller.org/

[52] CINECA RCM help page : http://www.hpc.cineca.it/content/remote-visualization-rcm

[53] RCM old code base : https://hpc-forge.cineca.it/svn/RemoteGraph/branch/multivnc/

[54] RCM new GitHub repo of deployment recipes: https://github.com/RemoteConnectionManager/RCM_spack_deploy

[55] EasyBuild Project home page : http://hpcugent.github.io/easybuild/

[56] Spack HPC package manager, SC15 presentation: https://tgamblin.github.io/files/Gamblin-Spack-Lightning-Talk-BOF-SC15.pdf; code repository : https://github.com/LLNL/spack

[57] Guacamole Project home page : http://guac-dev.org/

[58] NVidia grid product : http://www.nvidia.com/object/grid-technology.html
[59] NVidia H264 hardware encoder : https://developer.nvidia.com/nvidia-video-codec-sdk
[60] H264 in TurboVNC /VirtualGL : https://github.com/TurboVNC/turbovnc/issues/19
[61] Comparison of In-Situ frameworks : http://www.mcs.anl.gov/~wozniak/papers/Workflows_2015.pdf
[62] ISAAC zero-copy in-situ framework: description : https://arxiv.org/pdf/1611.09048.pdf
[63] ISAAC zero-copy in-situ framework: code repository http://computationalradiationphysics.github.io/isaac/
[64] DAMARIS Project library for data processing and in-situ visualization http://damaris.gforge.inria.fr/doku.php
[65] ParaView : http://www.paraview.org/
[66] Catalyst library for in situ visualization: http://www.paraview.org/in-situ/
[67] Catalyst enabled applications: http://www.paraview.org/catalyst-adaptors/
[68] Catalyst code structure: https://blog.kitware.com/paraview-catalyst-enabling-in-situ-analysis-and-visualization/
[69] Catalyst python scripts: https://blog.kitware.com/anatomy-of-a-paraview-catalyst-python-script/
[70] Visit Libsim in situ visualization tutorial : http://www.visitusers.org/index.php?title=VisIt-tutorial-in-situ
[71] In-Situ visualization of Navier-Stokes Tornado effect : YouTube Hpc-Summer video
[72] GpuTech conference site : http://www.gputechconf.com/
[73] Mediterranean basin In-Situ simulation : YouTube HPC-Summer video
[74] Fusion Forge, Software for managing the entire development life cycle of projects, http://fusionforge.org
[75] HPC Forge Project from CSCS - https://hpcforge.org/
[76] HPC Forge, reference to PRACE 1-IP WP-8 project hosted in its service, 2013, https://hpcforge.org/softwaremap/tag_cloud.php?tag=PRACE-WP8
[77] GitHub organization and teams online documentation, https://help.github.com/enterprise/2.0/user/categories/setting-up-and-managing-organizations-and-teams
[78] GitHub organization, roles and permissions, online documentation, https://help.github.com/enterprise/2.0/user/articles/permission-levels-for-an-organization-repository
[79] GitLab organization, roles and permissions, online documentation, https://GitLab.com/GitLab-org/GitLab-ce/blob/master/doc/permissions/permissions.md
[80] GitLab groups, online documentation, https://GitLab.com/GitLab-org/GitLab-ce/blob/master/doc/workflow/groups.md
[81] GitHub pricing, https://github.com/pricing
[82] Git Autodeploy, software for automatically deploying the latest version of your github project, https://github.com/olipo186/Git-Auto-Deploy/
[83] BuildKite, automation of software development processes, http://buildkite.com
[84] GitLab CI, GitLab continuous integration tool, http://doc.GitLab.com/ci/
[85] Jenkins CI, open-source continuous integration software, https://jenkins-ci.org/
[86] Atalssian Jira, Project Management tool, https://www.atlassian.com/software/jira
[87] Redmine, Open Source Project Management Tool, http://www.redmine.org

[88] Git, open source version control system, http://git-scm.com

[89] GitLab feature analysis, PRACE Internal Wiki, Zoltan Kiss, 2015, https://prace-wiki.fz-juelich.de/bin/view/Prace4IP/WP6/T62Service4UseOf, GitLab

[90] WP6 Service 4 - PRACE Repository Pilot documentation - https://prace-wiki.fz-juelich.de/bin/view/Prace4IP/WP6/T62Service4/

List of Acronyms and Abbreviations

aisbl	Association International Sans But Lucratif (legal form of the PRACE-RI)
BCO	Benchmark Code Owner
CoE	Center of Excellence
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture (NVIDIA)
DARPA	Defense Advanced Research Projects Agency
DEISA	Distributed European Infrastructure for Supercomputing Applications EU project by leading national HPC centres
DoA	Description of Action (formerly known as DoW)
EC	European Commission
EESI	European Exascale Software Initiative
EGI	European Grid Infrastructure
EoI	Expression of Interest
ESFRI	European Strategy Forum on Research Infrastructures
ESRF	European Synchrotron Radiation Facility
EUDAT	European Collaborative Data Infrastructure
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
GÉANT	Collaboration between National Research and Education Networks to build a multi-gigabit pan-European network. The current EC-funded project as of 2015 is GN4.
Geant4	Geometry And Tracking – a platform for simulation of the passage of particles through matter
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
GPU	Graphic Processing Unit
GUI	Graphical User Interface
HEP	High Enery Physics
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
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
HPL	High Performance LINPACK
ISC	International Supercomputing Conference; European equivalent to the US based SCxx conference. Held annually in Germany.

IN2P3/CNRS	National Institute of Nuclear and Particle Physics at CNRS
KB	Kilo (= 2^{10} ~10^3) Bytes (= 8 bits), also KByte
LINPACK	Software library for Linear Algebra
LRMS	Local Resource Management System
MB	Management Board (highest decision making body of the project)
MB	Mega (= 2^{20} ~ 10^6) Bytes (= 8 bits), also MByte
MB/s	Mega (= 10^6) Bytes (= 8 bits) per second, also MByte/s
MC	Monte Carlo
MFlop/s	Mega (= 10^6) Floating point operations (usually in 64-bit, i.e. DP) per second, also MF/s
MooC	Massively open online Course
MoU	Memorandum of Understanding.
MPI	Message Passing Interface
NDA	Non-Disclosure Agreement. Typically signed between vendors and customers working together on products prior to their general availability or announcement.
NGS	Next Generation Sequencing
PA	Preparatory Access (to PRACE resources)
PATC	PRACE Advanced Training Centres
PBS	Portable Batch System
PET	Positron Emission Tomography
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PRACE 2	The upcoming next phase of the PRACE Research Infrastructure following the initial five year period.
PRIDE	Project Information and Dissemination Event.
QOS	Quality of Service
RCM	Remote Connection Manager
RI	Research Infrastructure
SCM	SCM - Source Control Management system, software for the management of changes to source code computer programs.
SSH	Secure SHell
SVN	Subversion, a control version system software
TB	Technical Board (group of Work Package leaders)
TB	Tera (= 240 ~ 1012) Bytes (= 8 bits), also TByte
TCO	Total Cost of Ownership. Includes recurring costs (e.g. personnel, power, cooling, maintenance) in addition to the purchase cost.
TDP	Thermal Design Power
TRAC	Integrated SCM and Project Management. It provides an interface to Subversion or Git
TFlop/s	Tera (= 1012) Floating-point operations (usually in 64-bit, i.e. DP) per second, also TF/s
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
UNICORE	Uniform Interface to Computing Resources. Grid software for seamless access to distributed resources
VCS	Version Control System, also known as revision control or source control, is the software for management of changes to documents and source code computr programs
VNC	Virtual Network Computing

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	Cyprus Research and Educational Foundation, 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
ICM	Warsaw University, Poland (3rd party to PNSC)
INRIA	Institut National de Recherche en Informatique et Automatique, France (3 rd 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 (3 rd Party to SNIC)
LiU	Linkoping University, Sweden (3 rd 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 (3 rd Party to GCS)
SIGMA2	UNINETT Sigma2 AS, Norway

SLA	Service Level agreement
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
UC-LCA	Faculdade Ciencias e Tecnologia da Universidade de Coimbra, 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 (3 rd 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

In this deliverable, we present results obtained in Task 6.2 of Work package 6 of PRACE-4IP project. This task was focused to four new services that had potential to address some of the widely recognised needs in scientific computing: (i) providing protocols, policies and infrastructure for applications that need to be run in urgent situations, like flooding, fires etc.; (ii) establishing fast and reliable links between large scale scientific instruments and HPC centres from PRACE for fast and efficient data transfer; (iii) providing smart visualization tools, including guidelines, manuals and solutions for remote and in-situ data visualization; and (iv) providing repositories for European open source scientific libraries and applications, starting with long list of PRACE implementation projects deliverables.
The operational work on these services was following operational plan prepared within Deliverable D6.3. For each service, at least one pilot/prototype was developed and tested by several project partners. The deliverable has therefore four main sections – one for each service. Each section contains description of the services and the pilot(s), the evaluation of pilot(s), the final analysis of pros and cons and the proposal how to proceed with each service. This document is therefore also a basis for work in Task 6.2 of PRACE-5IP, where the matured pilots will be upgraded into regular services, some services will probably be closed and some new services will started being analysed.

1 Introduction

An efficient and state-of-the-art HPC infrastructure at European level should be ready to operate innovative services to address scientific, technological and societal challenges. In Task 6.2 of project PRACE-4IP we have examined four services. Following the interest of project partners with person months (PMs) assigned in WP6 Task 6.2, four groups of partners – one for each new service – were defined. The groups were coordinated by FIS and ULFME - Task 6.2 coordinators and started working at the beginning of June 2015. They were composed of:

1. Service 1: The provision of urgent computing services where the emerging computations results can help to issue critical decision-making paths in the case of a critical, national-scale emergency. The coordinator was PSNC and other partners involved in the task were: CSC, SURFsara, UHEM, ICHEC;
2. Service 2: The link with large-scale scientific instruments (i.e. satellites, laser facilities, sequencers, synchrotrons, etc.) providing a large amount of data and information which more generally require an improved support of data intensive applications. The coordinator of this task was UiO-SIGMA2 and other partners involved in the task were: CaSToRC, NCSA, NIIF and IN2P3/CNRS.
3. Service 3: Smart post processing tools including in situ visualisation to check and visualise dynamically the evolution of large volumes of data produced by simulations on extreme scale systems, where the data size represents a barrier for standard processing and visualisation methodologies. This task was coordinated by CINECA and the other partners involved in the task were: HLRS, IT4I-VSB, UL FME.
4. Service 4: Provision of repositories for European open source scientific libraries and applications, to promote wide adoption, uniformity at consolidation of European products. This task was coordinated by BSC and the other partners involved in the task were: EPCC, GRNET, NIIF.

In Deliverable D6.3 we have described operational plans for each service, related work and different related challenges. Specifications of pilots (prototypes) that were planned for each

service were also provided. In this deliverable, we present for each service the prototype(s) that was (were) developed for each service by Month 26. More precisely, for each service we:

1. shortly recall what is the service about:
2. present the pilot(s) - ;
3. provide the evaluations of all pilots within the service;
4. make propositions related to the service, where we propose answers to the following questions:
   - shall we continue with the service on full production level (regular service within WP6.1 in 5IP or 6IP) or shall we continue the pilot phase or shall we stop working on this service;
   - if we shall continue, how shall we proceed;
5. propose the main points of the policy, if we claim that the appropriate policy is needed;
6. propose first draft of KPI's;

During the finalization of Task 6.2 we have collected important technical details also in seven white papers, which passed internal quality control and are available on PRACE web page, see [1].

2 The provision of urgent computing

2.1 Description of the service

The work on Urgent Computing service was conducted due to the prospective and possible usage of PRACE Infrastructure in the future. It is foreseen to include this service into PRACE core service list in case of the agreement with the PRACE external user.

To precise the meaning of the particular parts of the Urgent Computing service we recall some terminology.

An **Urgent Computing User –** for the sake of this work it can be a real person from institution, not PRACE related, usually related to authorities, public service institutions (Police, Fire Departments, Crisis Management Departments if appointed, and so on). The specialist who is an expert in a specific urgent event and can thus decide if an urgent computation should be initiated and has the knowledge to evaluate urgent products to make recommendations to decision makers. The person should have been granted the access to the computing and data resources in advance, as contrast with the token based approach used by teragrid (http://spruce.teragrid.org/).

An **Urgent Use Case** is a description of a recurring issue, e.g. a flood, or a high impact issue, e.g. an explosion in chemical factory, which is expected to potentially result in extensive loss.

An **Urgent Event** is an occurrence at a point in space and time that can potentially create an extensive loss situation, which requires immediate attention. An urgent use case is thus a description of an urgent event.

An **Urgent Service** is the act of the activities, i.e. computation, decision making and coordination work, to fulfil the functions of an urgent system. In PRACE-RI, there will be the computation part of the urgent service chain.

An **Urgent Computation** is a computing activity that must start in short time, i.e. immediately or as soon as possible, to forecast the commencement and/or progress of an urgent event. An urgent computation is triggered by an urgent event.

An **Urgent Computing System** is a component within an urgent system that is in charge of the urgent computations. In PRACE-RI there will be selected Tier-0 systems (and Tier-1 systems), and selected data providers (eg. EUDAT, or alike).

An **Urgent Policy Definition** is the document stating the procedure corresponding to the agreement between parties (resource provider and end-user). In the scope of this work we focused on technical aspects of providing the service. Please refer to the White Paper ("Urgent Computing service in PRACE Research Infrastructure").

The corresponding parties and their relations are presented on the following figure.

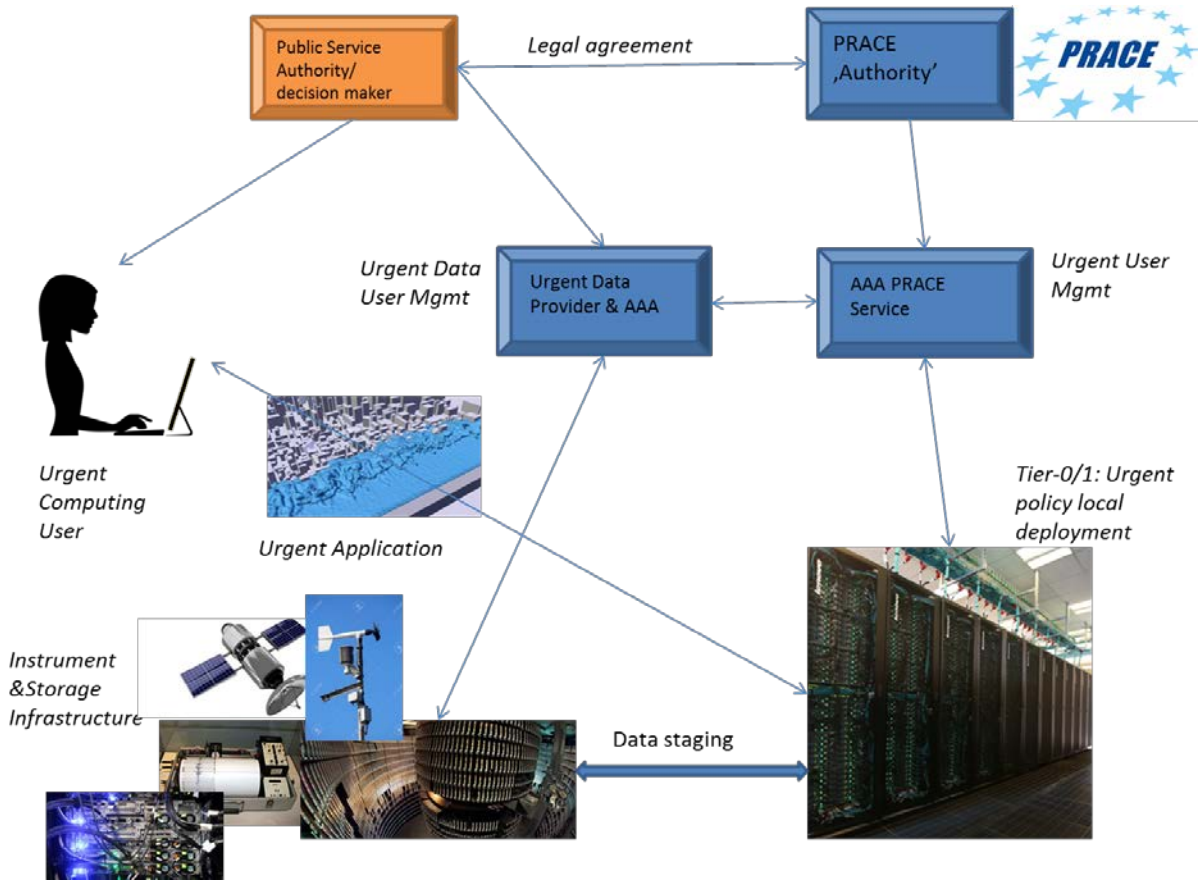


Figure 1: The relevant partners of urgent computing and their relations

We have investigated several possible scenarios and made the trade-off proposal on the PRACE-RI usage in urgent scenario. The ordinary use case must include the corresponding parties: Public Service Authority as the ordering party, PRACE Authority as the supplier of the service; UC User, AAA Management in PRACE, Urgent Application and the data source. The minimal agreement can state that the data staging is assured by the UC user itself directly on selected PRACE machine (without Instrument / Storage engagement at urgent time).

The operational platform supporting a UC case implies that the systems are each in a state of "warm standby" for the real event. To be, and remain, in a state of warm standby:

- All needed application software must be pre-installed
- One or more data sets suitable for validation must be pre-installed
- There must be a validation protocol using pre-installed software and pre-installed data.

- Runs to execute the validation protocol must be scheduled regularly to verify that the applications keep performing as they should, especially after system changes, software upgrades on general purpose and computational libraries, etc.
- Validation runs can be regular batch jobs. A budget of sufficient core hours to perform these jobs regularly must be allocated.
- Since pre-installed software and data may be damaged by human error, or hardware malfunctioning, or even the above noted updating of other software components, there should also be regularly tested procedure to quickly restore – reinstall, relink, recompile, reconfigure, whatever applies – the pre-installed components.

While validation of the software can blend in with the regular production environment, the workflow of a real event will inevitably need more:

- Platforms supporting a UC case must have a mechanism to raise the emergency flag, which can be executed by a preselected class of users: UC users, UC operators.
- The raising of the emergency flag must enable the availability of the required compute and storage resources in due time.
- What the exact amount of required compute and storage resources actually is, and what exactly is due time, is of course project dependent, but should be agreed upon in advance by the involved parties supplying and using the platform.
- How the freeing of adequate resources in due time is implemented should be at the discretion of the platform supplying side. Some sides may want to use pre-emption of already running jobs, others may e.g. have a large enough dedicated partition for jobs with a fairly short wall clocks time that they can drain from regular job usage.
- Complete workflow scenarios, including the actual making available of resources in due time, must be regularly practiced as “dry runs” as well.
- The regular testing of workflow scenarios is potentially much more disruptive of normal production work than the above mentioned regular software validation and their frequency must be agreed upon in advance, at the intake of a UC project. The budget in core hours allocated for a UC project should be sufficient to cover the actual loss of economic capacity resulting from the agreed upon level of workflow scenario dry runs.

The aforementioned precautions and statement must be investigated, when put into the legal tender between PRACE and external Urgent Computing User (Institution).

2.2 Experiences with the prototypal service

To implement specific solutions we propose several possible scenarios of integrating the prospective functionality into PRACE-RI under the general agreement of the technical body of PRACE and PRACE “authority”. The power of PRACE-RI is distributed, so we have to expect a non-uniform framework for managing the new services. First, we focused on balancing pros and cons regarding available ideas. In parallel, we touched technical and local policy aspect on selected test sites (PRACE Exec Sites): PSNC, ICHEC, SURF Sara, BSC.

The technical outcome of the document affirms the feasibility of running of urgent computing application (UC App) on each considered PRACE-RI site. We can conclude the expected policy in three main steps:

1. Appropriate software installed on dedicated EXEC Host.

2. The User of UCApp can regularly operate in a normal PRACE Project mode, e.g., as DECI project.
3. When urgent situation starts, the role UCApp should be activated as quickly as agreed in the policy agreement. Here we must agree on the current scheduler state disorder.

Hint: PRACE AAA policy should be extended in the following manner (add a new objectClass in subtree *People*):

```
cn
deisaHomeSite
deisaNationality
deisaRegistrar
gidNumber
homeDirectory
objectClass    top
objectClass    person
objectClass    organizationalPerson
objectClass    inetOrgPerson
objectClass    posixAccount
objectClass    shadowAccount
objectClass    UrgentComputing
objectClass    deisaUser
praceAccountStatus
sn
uid
uidNumber
deisaDeactivated
deisaDeactReason
deisaSubjectDN CN=..
deisaUserProfile    'name of project_uc'
gecos
givenName
loginShell     /bin/bash
mail
ou
seeAlso   ou=…
shadowExpire
telephoneNumber
title
```

There were several testing scenarios provided in [4]. They include the proposal of configuration of all available PRACE local queuing systems (LRMS): PBS, SLURM, LSF.

2.3 Recommendations for the next implementation phase

The conclusion is that the service is worth having it on the PRACE-RI as the current offer. We can support including it into PRACE service portfolio. However, due to strict specification of Urgent Computing Application, they are not comparable to each other; we cannot provide and maintain the host specific configuration now. Each application has got their own execution host requirements, that is why we propose to stop technical investigation here, but we confirm that PRACE-RI is able to operate jobs having strong time and resource constraints.

Regarding the prospective draft of legal tender, we conclude that the future policy agreement must focus on:

1. Appropriate software installed on dedicated EXEC Host.
2. The User of Urgent Computing application can operate in normal PRACE Project mode, as DECI project,
3. Running of the Urgent Computing application does not spoil any PRACE-wide policy nor EXEC site policy.
4. Exploitation of the real Urgent Computing application will impose to possess the licensing agreement with the owner of such application and input data by EXEC Host or PRACE-wide.
5. KPI must be related exactly with the installed application, then all KPI measured values must refer to the value agreed in SLA upon agreement on usage of this application. Each application service must have distinct KPI (Table 1).

Urgent Computing efficiency (KPI template)	
Description	Mean time between submitting UC job and its start (run) including data staging.
Calculation	SUM(time(UC App pre_operation))/number_of_test
Inputs	Measured time
Outputs	Mean time
Time-interval	Everyday.Scheduled reports as mentioned in the agreement
Threshold	Must be defined in the legal tender (SLA) in advance.
Tools	OS tools (time, timex, LRMS accounting data).
ITIL Category	Service Design – Availability Management
KPI Lead	PRACE
Implementation plan	Must be implemented at the selected EXEC machine.

Table 1: Proposal of Key Performance Indicators for Urgent computing service

3 Links with large-scale scientific instruments

Simulations and processing of Big Data are crucial components of workflows that involve large-scale instruments, such as DNA sequencers, laser facilities, telescopes, and particle accelerators. Within the PRACE-4IP collaboration, Task 6.2 partners from CaSToRC, NCSA, NIIF, IN2P3/CNRS, and SIGNA2/UiO each worked closely with a different scientific community to understand the workflows, and design a new service that will facilitate processing and analysing of large-scale instrument data on PRACE infrastructure. We have identified two major needs of scientists and operators of the large-scale instruments:

1. Analysis of the experimental results requires access to compute power, which is often beyond what is hosted on the instrumental site itself. Therefore, there is a need for access to external HPC facilities that could offer some CPU time.
2. Workflows that deal with large-scale instruments often transfer the experimental data from the instrument site to the external HPC facility. Moreover, scheduling of data processing jobs is often a crucial part of the workflow. To carry out those tasks efficiently, securely, and reliably, adequate software tools are needed.

Within our effort, we have concentrated on understanding the second aspect of the problem. In this context, the modern science can fully benefit from recent advancements in HPC if it addresses several technical challenges:

1. Large-scale instruments produce enormous amounts of data, which needs to be transported to, and stored within the HPC facility. Efficient use of massively parallel, multi-user HPC resources requires software that is aware of the limitations of the transport channel, and the storage solution used on both sites.
2. The gap between bandwidth and computational performance has been growing, and in many cases, transfer of the data from the instrument to the HPC facility proves to be more time consuming than the analysis itself. Since network bandwidth is a scarce, shared resource, efficient and reliable data transfer becomes more and more crucial.
3. The data requires post-processing, analysis, and visualization, all of which are compute intensive. Proper coordination of transfer and computations is crucial from the point of view of efficiency and reliability of the workflow.
4. Experiments are usually augmented with numerical simulations, which require substantial computational power, but can also generate a comparable amount of data. In this sense HPC facilities are becoming large-scale instruments themselves.

Overall, a lot of effort has been invested to tackle the individual challenges. However, complicated workflows often involve many tasks that are not standardized and require large manual (human supervised) effort. Addressing some of the above challenges in an automatic manner on the level of the HPC facility can substantially improve throughput and resource utilization.
The goal of our collaboration with the scientific groups was to identify common requirements of a service that addresses some of the above-mentioned challenges. We have then looked through the existing technologies to identify suitable ways to address them. Finally, we have implemented and deployed several pilot solutions aimed at solving the most pressing challenges of the individual scientific groups.

3.1 Description of the prototypal services

Within this task, PRACE partners worked closely with the scientific communities from ESRF (synchrotron, collaborating with CaSToRC), LHC (accelerator, collaborating with NCSA), ELI-ALPS (laser facility, collaborating with NIIF), LSST (telescope, collaborating with

IN2P3/CNRS), and NSC (DNA sequencers, collaborating with SIGMA2/UiO). Each of the scientific groups works with a different instrument and follows a different data analysis workflow. This gave us with a unique opportunity to get a broad overview of the type of challenges the scientists face. The following requirements comprise a unified view of the workflows. It is important to emphasize that they stem directly from the way the users work with the instruments.

- The service should provide a versatile and automatic data transfer method to / from an HPC facility, preferably in the form of a transfer job scheduler.
- The scheduler should be capable of suspending and resuming transfer jobs to free the bandwidth during ‘rush hours’, when users expect low-latency interactive access to a resource.
- To address a wide user base, it is crucial that the service makes it possible to use a variety of data transfer protocols on both ends (e.g., gridFTP, scp, sftp, rsync). In general, it should provide a framework for the users to implement their own (non-standard) transfer procedure.
- Challenges of transferring and storing a large number of files on a network file system should be addressed e.g., by providing a way to package and compress the files into a single file. This can be mounted as a device in Linux (e.g., a tar file, or a file-backed, loopback mounted file system), and from which the individual files can be accessed.
- 'Live staging' is needed to hide the transfer time behind the computations or instrument run time. This will decrease the time to delivery and improve the overall user experience. In this context, a way to update and re-transfer changed files is needed.
- Integration with HPC infrastructure requires that the service provides a method for automatic submission of data processing jobs after the data transfer has been finished, using e.g., SLURM. Together with automatic data transfers this feature will improve the complete workflow, ensure better reliability, and deliver a predictable throughput.
- The service should provide informative progress and error reporting, both when it comes to data transfers, and computations. The reports need to be accessible by external tools (e.g., through file IO), or reported directly to user-provided API (e.g., REST API). Hence, reporting requires a general framework that can be customized by the users to suit their needs.
- The service needs to provide an interface or a framework that will allow external user tools to use its functionality, e.g., schedule transfer and computational jobs, modify the parameters, provide means of authentication, etc.
- Performing data transfers and job submission without user interaction implies some automatic means of authentication on all ends. Usually, this means that either passwords, or private keys must be stored somewhere on the system. This may cause a security risk, especially since valuable data may be involved, and since an external HPC infrastructure may be used for processing. Hence, it is important that the service is designed from the ground up with security issues addressed.

The above list of requirements is very broad, and in our research, we have concluded that no single existing software solution addresses all of them. In addition, while working with the scientific partners we have observed that each group has specific needs regarding each of the listed points, and each requirement has a different importance. For these reasons, we concluded

that it is not feasible to provide a unified service that would implement all the identified requirements in a flexible and useful enough way to convince the users to change their current way of work. By necessity, each relevant challenge is already addressed in some way in the existing workflows. A service that would implement all the above requirements would be complex, and would require substantial development to become production ready. In the end, it is not certain that the users would be willing to spend the effort to modify their existing workflows.
For these reasons, we chose to tackle a smaller subset of the requirements and provide the individual scientific collaborators with some tools that address their most pressing challenges. In our pilot deployments, we chose to concentrate on the following aspects:

- Automatic data transfers with job management
- Automatic submission of compute jobs to the HPC infrastructure
- Flexibility of data transfer protocol
- Dealing with large number of small files (transfer and storage)
- Security related to data transfer, data integrity, and authentication

To this end, the following solutions have been developed and piloted by the named PRACE collaborators:

1. Using gsatellite for scheduling of data transfer and compute jobs (NIIF, NCSA)
2. Dealing with large numbers of small files in HPC environments (SIGMA2/UiO)
3. New framework for data packaging and transfer (SIGMA2/UiO, CaSToRC)
4. HTTP for bulk file transfer over high latency network links (IN2P3/CNRS)

Technical aspects of the implemented pilots have been discussed in details in the submitted whitepapers [5]. Each partner's findings and experiences related to their pilot deployment are discussed in the Section 3.2 (Experiences with the prototypal services). Here, we briefly summarize the general concepts behind each solution.

3.1.1 *Using gsatellite for scheduling of data transfer and compute jobs (NIIF, NCSA)*

Being able to transfer large volumes of valuable data is a central point in linking large-scale scientific instruments with HPC infrastructure. Whether the data is created through experiments, or synthetically, it has to be reliably transferred, stored, and archived for future reference. In addition, scientific workflows include a data processing step, which depends on the availability of the data, and is often executed on a remote HPC resource. Automatizing the workflow requires tools capable of scheduling the data transfer in both directions, while interacting with the HPC scheduler to submit and monitor the compute jobs. In this context, we have analyzed the suitability of gsatellite - a data transfer and compute job scheduler - to enable fast transfer of multiple datasets with fault tolerance and a notification system. We discuss the results of our investigations on the example of two large-scale research infrastructures – accelerators (in the context of high-energy physics research and medical applications) and Extreme Light Infrastructure (ELI) [7]–[8].

3.1.2 *Dealing with large numbers of small files in HPC environments (SIGMA2/UiO)*

Processing of large numbers (hundreds of thousands) of small files (up to a few KB) is notoriously problematic for all modern parallel file systems. While the available storage solutions provide high and scalable bandwidth through parallel servers connected with a high-speed network, accessing small files is sequential and latency-bounded. Paradoxically, performance of file access is worse than if the files were stored on a local hard drive. We have developed a generic solution for large-scale HPC facilities that improves the performance of

workflows that process large numbers of small files. The files are saved inside a single large file containing a disk image. When needed, the image is mounted through the Unix loop-back device, and the contents of the image are available to the user as a normal directory tree. In the pilot deployment, the framework has been seamlessly integrated into a SLURM-managed HPC environment, and used to store read-only software modules created by administrators, and user-created disk images with read-only application input data. We investigate the benefits of this solution on the example of working with DNA sequencers at the Abel supercomputer facility in Oslo, Norway [5].

3.1.3 *New framework data packaging and transfer (SIGMA2/UiO, CaSToRC)*

Preparation of data for network transfer often involves packaging (tar, compression, encryption), and usually requires extra disk space to store the archive. Assuring data integrity and completeness on the destination can be implemented by hash computations (e.g., MD5, or SHA) on both ends. All this preparatory work is usually done sequentially, and consecutively with the data transfer. To simplify the above generic workflow we have developed a portable and flexible framework for transferring of large amounts of scientific data. Its main objective is to prepare the data for transfer (e.g., tar, compress, encrypt, compute and verify hash), while delegating the network transfer to established external utilities (gridFTP, bbcp, scp, sftp, and others). The computations are done in memory, and on-the-fly, hence there is no need for extra storage. Compute-intensive preparatory tasks are executed concurrently with the network transfer, which decreases the overall execution time of the workflow. The framework is implemented in Python3, it is portable, and easy to extend. We provide a general-purpose stager application driven by configuration files, and a Python API, which has been used to implement custom workflows. The relevance of the developed framework is discussed in the context of the DNA sequencers at the University of Oslo, and of the ESRF accelerator community collaborating with CaSToRC [6].

3.1.4 *HTTP for bulk file transfer over high latency network links (IN2P3/CNRS)*

Moving massive amounts of data between geographically distant sites is a typical requirement of nowadays-scientific experiments. Long haul networks linking instruments and data processing sites are commonly used in the distributed platforms designed for processing the data handled by modern scientific experiments. Experimental data is typically collected at the site where the instrument is located and then transported to one or more data processing sites for archival and processing purposes. Specific tools have been developed over the years for efficient transport of data over high latency network links. Some of those tools extend a standard protocol (e.g. FTP) and other tools are custom-built for transporting data over network links in an efficient way. The standard *Hypertext Transfer Protocol* (HTTP) extensively used by web applications, has been traditionally avoided for bulk transfer of scientific data. In this work, we have developed several tools to evaluate a modern implementation of HTTP as the protocol for transporting data over long distance network links. We report on our experience in the context of transferring multi-terabyte data sets of raw images of the southern sky, which starting in 2022 will be taken every night by the imaging device of the Large Synoptic Survey Telescope (LSST) in the Chilean mountains [3].

3.2 Experiences with the prototypal services

3.2.1 *SIGMA2/UiO*

The supercomputing group at the University of Oslo collaborates with the *Norwegian Sequencing Centre* (NSC) - a technology facility that offers sequencing services on the HiSeq 2500, NextSeq 500 & MiSeq instruments from Illumina. Working with modern high throughput sequencers requires an HPC infrastructure, as well as robust procedures to efficiently transfer and process huge volumes of data. A single run on the current generation of Illumina instruments produces around 1TB distributed over roughly 100.000 small files. At NSC, the resulting data is transferred to a long-term backup storage. Non-sensitive data is also staged on the local Tier-1 Abel system for processing. For sensitive data analysis, UiO has developed a dedicated service (TSD), which provides users with a 'safe' computational resource. The procedure of moving the data into the facility is restricted to SFTP with two steps authentication using a password and a TOTP token. In the current setup, the server-side monitors the import directory and copies the incoming user files further into the secure facility. However, just by looking at the files it is impossible for the server side to know, whether the transfer is complete, or broken, which causes some technical difficulties.
When working with DNA sequencers, one of the most important bottlenecks is the large number of files they produce. Transferring individual files over the network takes a very long time and results in underutilized bandwidth. This is usually addressed by packaging of the files prior to the transfer, which requires additional space for the archive. In addition, unpacking and accessing lots of small files on modern network file systems is very inefficient. This is mostly due to high latency of metadata operations, which dominate the access time for small files. After the data has been transferred and staged, processing jobs are submitted to the HPC resource. At NSC this is currently done manually. The compute jobs are monitored, and the results are manually copied back to the home institution. The facility is going to be significantly extended in the coming years, therefore automatizing the workflows is important for their efficiency and predictability of the throughput.
Within this PRACE effort, we have developed and deployed two software solutions: a tool for mounting and unmounting of disk images to store large numbers of small read-only files on network file systems [5] and a flexible data transfer framework [6]. The aim of the developed tools is to increase reliability of the NSC workflows through the following improvements:

- improve the efficiency of data staging and processing by saving the small files inside loop-back mounted disk images
- automatize data preparation (packaging, encryption) and perform it in memory, on-the-fly, and concurrently with the network transfer
- signal to the server-side monitor that a transfer has completed by computation and verification of the data stream hash
- automatize the workflow by submitting a compute job after a successful transfer of the data

The disk images solution has been in production for almost a year on the Tier-1 Abel resource, where it's used to provide the software modules to the compute nodes. We have observed significant improvements of start-up time of large software suites with tens of thousands of files (e.g., MATLAB, Intel compiler, gcc), but also for short user programs that require many modules (e.g., some Python / perl / shell scripts). The other positive effect is a decreased pressure on the meta-data servers, and thus better overall file system performance, which is especially important on systems with many users, who run complex pipelines of short processing jobs (e.g., the bioinformatics community).

The data transfer framework has proven to be especially useful for the TSD users. We are currently integrating the developed tools into a GUI-based user software, and a web interface to TSD. Since most users of this platform run on Windows, portability of the framework is especially advantageous. Other benefits include the ability to simultaneously backup and stage the data to the HPC resource, verify transfer completeness, and automatically submit compute jobs. Currently, a disadvantage of the framework is inability to restart a broken transfer. Also, support for transfer tools is limited to what we needed to support in the pilot setting. Further development is needed in these areas.

3.2.2 *CaSToRC*

With 6000 users from all over Europe each year, the European Synchrotron Radiation Facility (ESRF) produces around a TB of data per hour, almost 24 hours a day. The data needs to be stored and archived for future reference. In addition, analysis of this amount of data (e.g., tomographic reconstruction) requires large compute resources. CaSToRC is currently collaborating with the software and data analysis groups at ESRF, which amongst others support users to perform experiments and analyse the collected results. A typical workflow consists of several steps. The prepared samples are exposed to the X-rays, and the raw data produced by the instrument is stored locally in a central storage system. After the experiment is finished the data is processed using the pyHST software on a local, GPU-accelerated cluster, and a digital volume representation is produced. The results and the raw data are stored for 50 days on an externally accessible storage. Users stage out the data using SFTP or SCP, or by attaching USB disks directly to one of the instrument workstations.
Now, ESRF user communities perform most of the tasks in a manual way. Moreover, since the locally available HPC resource provides limited compute power, scientists must often post-process the data themselves in other facilities. This imposes several difficulties and therefore several opportunities for improvements. Since some of the datasets consist of thousands of small files, automatic in-memory packaging before the transfer [6], and storing the data inside disk images inside HPC facilities [5] has been tested. The pilot involved transfer of the data to the Tier-1 system located at CaSToRC for analysis. The developed software, and the pilot results have been presented to key technical staff at ESRF. Through the collaboration we've received valuable feedback on how to further extend the service, as well as how a future collaboration could be established between ESRF and PRACE. In general, the feedback regarding the comprehensive handling of small files was very positive. The ESRF staff appreciated the idea of the pipeline transfer, and especially the fact that all computations on the transferred data are performed in memory, with no need for extra storage. The modularity and extensibility of the data transfer framework has been seen as an advantage. It has been proposed to extend the framework with ability to transform the datasets to HDF5 format on the fly, and prior to the transfer. As a disadvantage, the collaborators expressed reluctance to include the framework in their production workflows because the framework is still in the prototyping phase, while they have some tested, internally developed solutions. Regarding the EXT4 disk images, they were impressed with the performance improvements and they liked the concept of loop-mounting a file based disk image on a parallel file system.
During our collaboration, it was emphasized that ESRF has a great need for additional compute resources. Our collaborators gave a positive feedback for the developed tools, but they stressed that their main interest lies in obtaining access to external HPC resources to off-load their jobs during busy hours. The ESRF staff expressed their keen interest in an official collaboration with PRACE, possibly by the establishment of a service that would provide large institutional partners with compute time on the PRACE infrastructure. It has been suggested that two types of such 'institutional' access would be particularly useful:

- Close to real time access, while the experiment is running, to perform quality control of the experiment and adjust the experiment as it is happening. This type of computations would be performed by the operators of the instrument.
- Post processing computations, which would be performed after the completion of the experiment either by the users of the instrument, or by the ESRF staff.

3.2.3 *IN2P3 / CNRS*

The *Large Synoptic Survey Telescope* (LSST) under construction in Chile is scheduled to operate during 10 years from 2022, producing a total raw data volume of 15 TB per 24 hours' period, approximately 300 nights per year. Processing of the data will be performed by two centres, the *National Centre for Supercomputing Applications* (NCSA, USA) and the IN2P3 / CNRS computing centre (CC-IN2P3, France). Raw data coming out of the instrument will be recorded and archived in a data centre in Chile, near the acquisition site, and promptly transported to NCSA in USA, which will record the second archive copy. The full raw dataset will then be transported from NCSA to CC-IN2P3, which will host the third archive copy.
One of the most challenging aspect of the setup is the necessity to transfer large amounts of data daily over long distance, high latency networks. Since the data will be produced almost continuously, to move it in time the transfer bandwidth needs to be maximized. Existing bulk file transfer systems at the scale required by LSST are typically implemented using GridFTP as the underlying transport protocol (e.g., the LHC computing grid, Globus Online, etc.) and in some cases iRods. In this pilot service, we explore alternative implementations based on standard protocols likely to still be relevant during the 2020-2030 time frame. Specifically, the recently standardized HTTP2 protocol includes several built-in features, which makes it attractive for the use case we want to address. It is strongly supported by the industry, with several implementations already available.
The testbed system deployed at CC-IN2P3 is devoted to evaluate bulk data transport using the HTTP protocol over a shared, production transatlantic link of 10 Gbps with a RTT of 110 ms. The testbed consists of 4 hosts equipped with a 10 Gbps network card, which transfer data to/from NCSA in USA. Within the PRACE-4IP effort we have developed a set of tools for measuring the usability of a modern implementation of the HTTP protocol for memory to memory transfers [3]. We deliberately excluded disk I/O on both ends to focus on measuring the protocol performance. These are the lessons we learned from this research:

- The HTTP protocol seems to be a viable option, compared to well established tools for data transfer such as gridFTP, bbcp, scp, etc.
- The possibility to program both HTTP client and server is a valuable advantage for both infrastructure operators and end users, compared to using less flexible, monolithic tools.
- HTTP is very likely to remain relevant for the coming decade, as the Internet industry depends on it. If it were to become obsolete, it would be replaced with more modern tools which we could also consider for transporting data.
- Although for this research we used the HTTP implementation of both client and server built in the standard library of the Go programming language, there are similar implementations for any relevant programming language. It is possible to decouple the implementation of client and server. This is an advantage for the end users, which will allow them to embed data transfer capabilities into their own applications, regardless of the programming language.
- The ubiquitous nature of HTTP may become more relevant in the future where a large number of small devices, collectively known as the Internet of Things, will emit

significant amounts of data, which, like large scientific instruments today, may require HPC-like data processing capabilities. Being able to transport those data directly from the emitters to the data processing centers using standard protocols is an attractive feature of HTTP.

- Using encryption on top of HTTP via the standard protocol TLS is practical nowadays, provided you use recent hardware with built-in hardware-assisted implementation of encryption algorithms, as is the case for AES. This is not a limitation in practice since Intel CPUs commonly used by servers provide this capability since about 2010.

Interestingly, our research has shown that a recently standardized version of HTTP, named HTTP/2, which includes request multiplexing among other features, imposes some severe penalty in terms of throughput, compared to HTTP/1. The cause for this unexpected behaviour needs to be further investigated and understood.

3.2.4 *NIIF (Hungary) and NCSA (Bulgaria)*

ELI-ALPS laser facility

The *Attosecond Light Pulse Source* (ELI-ALPS, http://www.eli-alps.hu/) is part of an ESFRI project to build world leading next generation laser facility that aims to analyse interaction between light and matter. The instrument has not been put into production yet. Its estimated data production rate is 20Gbps. Parts of the data produced by the detectors is of no interest and can be discarded in an *online processing step*. Pre-processed data will be written to the local *offline storage* for scientific analysis.

The bottlenecks of the setup are the bandwidth between data acquisition systems and o*nline data analysis*, and the local computing capacity required to process data online or offline. Assuming an average load, local *offline data processing* (e.g., data analysis and Monte Carlo simulations) will be performed using the hosted HPC facility. The estimated computational capacity required on site reaches up to 21000 simultaneous cores, and data storage requirements are up to 3,5PB per year. However, the amount of user requests might fluctuate significantly, which can result in an occasional high load that will exceed the internal capabilities of the facility. The required capacities in terms of data transfer rate, storage and computation suggests the ‘overload’ might be a frequent case. Hence it is expected that during the busy periods there will be a need to offload the compute jobs to some external HPC infrastructure, possibly requiring contracts with several external companies, and collaboration with PRACE.

A reliable workflow including data transfer to the external site, processing of the data, and sending the results back to the facility needs to be implemented. Security is important as the experiment results might not be public in the early phases of processing. Within the pilot deployment we have tested GridFTP is as a candidate for transferring large amount of data, and *gsatellite* as an advanced tool for scheduling of *gtransfer* jobs that can complete the transfers reliably, and to ensure transactions are done without errors.

HEP and Hydron Therapy

Being able to produce and transfer huge number of simulated events is crucial when performing GEANT4 simulations for analysis of High Energy Physics (HEP) experiments, or for use in hadron therapy facilities. As a representative example of modern experiments in the field of HEP we consider the experiments performed on the Large Hadron Collider (LHC). A typical HEP workflow involves comparison of a sufficient amount of synthetic data created through simulations of particle collisions to experimental data. Synthetic data needs to be transferred to and stored on LHCG Tier-0 or Tier-1 centres to allow scientists to use it. Statistical uncertainty in all simulated physical quantities depends on the number of simulated events, which

necessitates large number of simulations, and produces huge volumes of data. The event generation step involving GEANT4 is the most compute demanding part of the workflow, and it is important that simulated events are generated, stored, and transported reliably. In addition, processing sensitive data (e.g. from hadron therapy treatment) must meet strict security policies, which in many cases additionally complicates the workflow.
The HPC approach used here is based on GRID technology. The largest research facilities in HEP (high-energy physics) use the Worldwide LHC Computing Grid (WLCG) infrastructure to perform such simulations. In this approach, different computers simulate different events. The events are fully independent, so no data transfer between different computers is needed. The problems along this way are two-fold:

- With the increasing data volumes, the resources become more and more insufficient, even with structures like WLCG.
- New architectures like those involving large number of Intel Xeon Phi accelerators are not used efficiently by the largest HPC community worldwide.

A service aimed at linking of GEANT4 workflows with PRACE infrastructure must address the primary bottleneck, which are the large computational requirements of the GEANT4 simulations. The pilot service aims at relaxing this bottleneck by running the simulations on powerful hybrid architectures with Intel Xeon Phi co-processors. In the future, exposing large computational capacity of PRACE systems through the developed service would greatly benefit many scientists working HEP, and in the field hadron therapy research.
Seamless integration of the service into exiting workflows demands automatic authentication, reliable and efficient data transfer, and automatic submission of simulation jobs to the HPC infrastructure. The submission mechanism should monitor the jobs and notify the user about its status. Automatic resubmission of the jobs should be possible in the case of service failures. In the hadron-therapy centres, using an automated workflow for data transfer and computations will provide better throughput and reliability, and hence allow the staff at these facilities (medical doctors and medical physicists) to focus on the medical aspects of the treatment planning and delivery. To automatize data transfer and job scheduling, within this pilot project we have tested *gsatellite* as an advanced tool for scheduling of transfer and compute jobs.

3.3 Pilot evaluation

Based on the requirements identified for both use cases, the workflows would benefit from a reliable data transfer job scheduler like *gsatellite* to enable fast transfer of multiple datasets with fault tolerance and a notification system. Combined with other wrapper tools, like *gtransfer*, the setup would allow transfer tasks to be more robust than using common transfer tools to handle transfer of each file separately. *gsatellite* is also a promising candidate for a job scheduler for services on hybrid architectures including Intel Xeon Phi co-processors.
The pilot deployment has shown that in both cases, the usage of *gsatellite* in linking HPC facilities to large-scale scientific instruments was justified [7]. The identified advantages and disadvantages have been summarized in Table 2. Importantly, *gsatellite* is already validated by PRACE, and using it in a framework requires no additional tools to be installed near the partner infrastructure. The framework using *gsatellite* provides reliable transfer of multiple large files between the scientific equipment and compute site, along with automated computations for specific use cases without user interaction. The framework is especially valuable when an input parameter or data changes, while the code itself not. Workarounds have been successfully implemented to cope with the weaknesses (automatic submission of jobs, etc.).
The responsibilities handled by *gsatellite* in the implemented services include:

- Transferring input data to remote HPC facility

- Scheduling compute job on HPC machine
- Monitoring remote job
- Transferring output file back home

gsatellite has been tested with *gtransfer* and *xrdcp*, but it can also be used with any other file transfer utility. An advantage of *gtransfer* is that it is supported by PRACE. SCP can be added to extend the service to potential users lacking GRID experience and infrastructure. In the pilot implementation, Globus toolkit, *gtransfer*, *UberFTP*, and *TGFTP* are installed on the service front-end and are accessible for the users.
In the first use case, the service design was inspired by Cloud technologies, i.e. sending the request and getting automatically the job completed. The workflow for the second use case is similar but it tries to solve the problem using only PRACE solutions (core software described in PRACE service catalogue). The presented framework still requires practical evaluation by users. For the tests, the available PRACE Tier-1 systems Avitohol and LEO can be used.

Strengts	**Weaknesses**
Gsatellite was evaluated by PRACE security team and found to be secure	Gsatellite is still very basic (compared e.g. to services in EUDAT B2SUITE).
Written in Bash – easy to modify, port, maintain	Transfer is user based (needs user keys or certificates).
Gtransfer is already an additional PRACE service	Transfer and compute are not integrated, need further integration
Able to use different transfer tools apart from Gtransfer. The variety of potential tools is beneficial for users lacking GRID infrastructure and certificates	Needs to "communicate" with preinstalled system scheduler in order to execute computation jobs
Provides scheduled and automatic data transfer to/from an HPC facility	Actual offloading to external resource is not yet automated
Actual workflow is achieved by submission of transfer and compute jobs	
Provides job monitoring, retries transfer and notifies user on error	
Uses standard X509 certificate based authentication, which is standard at PRACE and compatible with EGI, EUDAT, ELI infrastructures	
Supports encryption when used with Gtransfer	

Table 2: gsatellite - advantages and disadvantages

3.4 Conclusions and recommendations for the next implementation phase

Linking large-scale scientific instruments to the HPC eco-system is important for strengthening of the European excellence in fundamental research. Within PRACE-4IP, Task 6.2 we have collaborated with 5 scientific groups that use different instruments in their work. The goal was

to understand the workflows, and to identify common and individual requirements for a service that would facilitate working with the instruments on PRACE infrastructure. We have identified two main areas, in which scientists and infrastructure operators could benefit from a collaboration with PRACE:

1. Introduction of 'institutional' access to PRACE HPC infrastructure to offload data analysis work from the instrument centers, which often don't have the necessary capacity.
2. Development of modern tools that would automatize complex workflows involving data transfer and compute job submission and monitoring.

In PRACE 4IP we have mostly addressed the second aspect. We have identified the requirements of a unified service, which could be beneficial both for the users, and the infrastructure operators by improving the efficiency and reliability of the scientific workflows. The list of requirements is very broad; hence we have decided to concentrate on providing solutions to smaller sub-problems of the individual scientific groups. To this end we have developed and tested the following pilot services:

- Using gsatellite for scheduling of data transfer and compute jobs
- Dealing with large numbers of small files in HPC environments
- New framework data packaging and transfer
- HTTP for bulk file transfer over high latency network links

Our experience with the pilot deployments shows that the developed tools can become useful parts of scientific workflows in the future. However, deployment as a unified PRACE service is not feasible. In many cases the tools must be adopted by the users, and not by the infrastructure operators (e.g., data transfer framework, gsatellite). Hence, their adoption will depend on the willingness of the users to modify their existing workflows, and that takes a lot of time. The disk images framework can be deployed system-wide by administrators, but is only useful for cases where the data is stored in a large number of small files. Hence, deployment is only needed in certain cases. On the other hand, the newly developed HTTP transfer tools could become a viable alternative to established tools like GridFTP in the future, but they still require significant development. We conclude that, while it is important to further invest into the development of those tools in PRACE-5IP and advertise them to the users, with the exception of gsatellite cannot be deployed as a PRACE service.

The second identified area, which we have not analysed in detail in PRACE-4IP, is providing some sort of institutional access to the PRACE infrastructure. Three out of five collaborating instrument partners indicated their interest in obtaining regular access to additional compute power at the PRACE facilities. Operators of ELI-ALPS and ESRF, and scientists related to LHC have to perform regularly certain computational tasks, which are large in volume, but routine in nature. Thus, they do not qualify for resource allocation under PRACE Project Access calls, but the computational work is nonetheless decisive for the research. We suggest that during PRACE-5IP such a possibility is considered, and a formal framework for institutional access to PRACE is developed.

4 Smart post-processing, Remote and In-Situ Visualization

The main goal of service 3 within Task 6.2 was to define which services, policies, tools could be deployed to support post-processing activities related to HPC simulations running on PRACE HPC facilities.

We decided to focus on three topics, related to different aspects of the common problem of extracting useful information from the results of numerical simulation running on our HPC

centres. These three aspect corresponds to different "pilots" that are relatively decoupled but nevertheless could come into play within the same HPC and even be used within the same community or the same application in different stages:

1. **Remote Visualization**: is widely described in the introduction part of the Remote Visualization white paper [9]. This pilot deals with the task of providing end users, sitting in their office or homes, an environment that allows them to run, directly on the remote HPC facilities where their data is stored, their preferred post-processing work-flows, including interactive GUI and 3D graphical post-processing applications.
   We assume here that data transfer operations, especially over geographical distances, scale badly as the data grows and that it is much more efficient to move post-processing environment where the data is residing instead the other way around [32].
   Several technological solutions address this widespread needs and all of them try to be as much application neutral as possible, providing remotisation facilities in an application independent way.
   Many HPC centres already provide some form of this kind of service but there is not much standardisation, neither in technology nor in service policies.
   Within this pilot we have collected information regarding different deployed solutions, identifying use cases, performance indicator, common components and relevant policies.
   We have also selected a set of existing open source components, used within existing services and have tried to simplify both their usage as well as their deployment.
2. **In-situ visualization:** The main goal of this action is to explore one of the promising approaches for addressing the further scaling in data sizes: this approach is directed at avoiding or greatly reduce the need of saving the full data of a simulation to be post-processed later.
   The idea is to enhance the simulation code itself by adding ability to run much of the post-processing functions directly within the simulation code itself, being able to produce ready to use visual products (images) or greatly reduced significant data artefacts.
   While most of the approaches described in literature [55]–[57] try to minimise the impact on simulation code, none of them seems to be really application neutral. For this reason, our approach has been to select one technology amongst the most widespread and solid, and try to apply it to real application cases, trying to extract from this experience useful hints and best practice to promote adoption within other applications.
3. **Large data I/O optimisation:** this pilot address the feasibility to have general techniques to optimize I/O operations when large data have to be processed multiple times. A real case dealing with analysis and parameter optimization within the industrial fluid dynamics field has been addressed.

4.1 Description of the prototypical services

All pilots have in common the goals of reducing the turnaround time from simulation to insight, and rely on open source technologies and try to be as much as possible architecture neutral.
The first two pilots deals with visual interactive post-processing applications and both aims at simplify their deployment and adoption; they aims at providing an environment where the computational scientist find easy to use visual tools to explore and analyse the data.
Both pilots have been deployed on different clusters and use a deployment environment tailored to HPC environment. Specifically this includes in-situ visualization (pilot 2) which requires an environment similar to the one provided by remote visualization (pilot 1): in-situ experiments

rely on ParaView application being available and usable within a remote visualization service, they could be viewed as scaled up visual applications, leveraging on application neutral remote visualization services.

4.1.1 *Remote visualization service*

Existing visualization services deployed in HPC centres generally and PRACE specifically have been scouted and analysed, regarding used technologies, resources and policies. As noted, many deployed services rely on some common Open Source components. Specifically the open source stack TurboVNC [39]/ VirtualGL [40] has been considered as possible common ground for a prototypical service.
Additional components such as lightweight window managers and largely used visual applications (ParaView [64], Visit [70], Blender) have been considered as optional components of a visualization service deployment.
From the analysis of the different user access policies, it is quite clear that one important factor for user engagement, is the simplification of installation and start-up procedures for accessing remote visualization services.
For that reason, we have focused this pilot activity around evaluation and enhancement one custom solution, adopted by CINECA for one of its remote visualization service [42].
This tool, called Remote Connection Manager, is a lightweight python GUI interface, which wraps underlying VNC component for simplifying VNC client installation, remote session activation and bookkeeping.
The python code can be "compiled" using the cross-platform packager Pyinstaller tool [51]: it gets packaged along with VNC client and other dependencies, into a standalone executable that on start-up unpack itself into temporary area and then executes, avoiding the need of a real installation.
This small, cross platform tool has been evaluated outside CINECA, and released as Open Source. A refactoring activity has started to allow it to be easily deployed in different contexts by other HPC centres. We extensively describe the following software stack components:

- VNC session, start-up and connection
- Window manager component and setup
- VirtualGL OpenGL interposing and GPU remotization library
- Remote Connection Manager python wrapper GUI

Another issue that this pilot focused on is the provisioning of general deployment recipes for the different components of remote visualization services.
Different HPC deployment tools [55] have been evaluated as candidates for deployment tools. Spack [56] project, supported by LLNL has been selected as a basis for the automated deployment of all the components needed for deploying remote visualization service.
We have contributed updates of recipes for general use, while we have collected project specific recipes in a separate repository. The work is currently carried on in GitHub repositories [54].
This approach to deployment has been used in CINECA to provide remote visualization service on the newly installed Marconi cluster. Porting on Marconi has raised few issues as it is a Tier-0 machine with no GPU nodes:

- As there is no GPU on any node, VirtualGL component has to be substituted with another software only OpenGL emulation library. We have used Mesa with OpenSWR [41] support in order to provide Intel optimized support for 3D interactive applications such as ParaView that depends on OpenGL.
- As our goal was to allow VNC session to be hosted on any Marconi compute nodes, we would prefer not to require any modification to the available computing environment

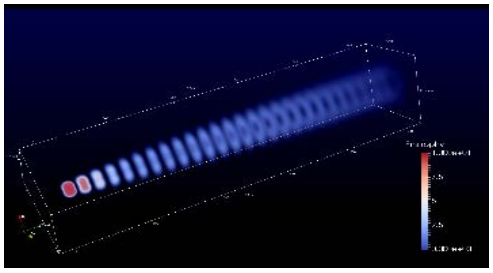
that currently does not include any X11 support. So, to address this requirement, the components required for remote visualization service has to be deployed at the unprivileged user level, so requiring a source build from scratch. Fortunately, Spack package manager was already providing user-level source recipes for most of the X11 stack, so a relatively low number of additional packages recipes had to be defined.

- A completely different impact on resource usage has been observed: Intel multicore OpenGL emulation is quite effective but require a substantial amount of cores to provide acceptable frame rate, even for rendering of relatively simple 3D scene. This has the practical implication that is quite difficult to host 3D application on interactive session on shared login nodes.

4.1.2 *In-situ Visualization experiments.*

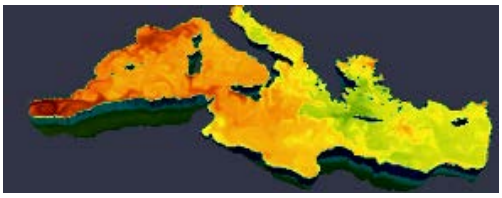
There are different approaches and libraries for in-situ visualization [61], many of them are promising approaches but do not have a large supporting community [64] or are quite new and specific to some use case [62]. As ParaView [65]–[69] is one of the most used visualization tools, we decided to focus on its in-situ built-in solution, named Catalyst [66]: this way consolidated experience with traditional post-processing visualization with ParaView could be reused within in-situ context.
Under the joint umbrella of Summer of HPC activity, we selected two real simulation codes as candidates for being instrumented using ParaView Catalyst to add in-situ visualization capabilities [67].
Catalyst is able to provide both interactive as well as batch visualization: the simulation code get instrumented with Catalyst and ParaView code calls that get linked with the instrumented simulation application. The ParaView side of the code see simulation code memory through user defined “adaptors functions” (Fortran and C++) as data sources [68].
The ParaView data structures mapped from simulation memory are exposed are available to ParaView filters to extract useful artefacts like reduced and filtered data or proper rendered images. These filters, arranged in the usual ParaView data-flow paradigm, are defined in a function (usually defined by a python script [69]) and are available during simulation time for extracting the needed artefacts.
Without code recompiling, the user can change the type of visualization pipeline to apply and the frequency of artefact extraction (typically, large data artefacts are extracted and saved less frequently than small image rendering).
The library allows interactive connection from a ParaView client to allow real-time interactive steering of the simulation.
The selected code were:

1. Baseline CFD simulation of extropy effects possibly leading to tornados.



This is a highly scalable MPI numerical simulation baseline CFD code. It has been instrumented to access its regular computational grid that is distributed amongst its MPI cores.
The instrumented code allow ParaView to interactively connect to the running simulation to visualize the current simulation variable to check for its evolution, see YouTube Summer of HPC movie [71].
The work will be also presented in the poster session of GpuTech conference [72] with title “Insitu visualization with ParaView Catalyst”.

2. Forecast of nutrient dispersion in the Mediterranean basin.



This is a 2.5 D Code that forecast various nutrient dispersion, it runs in a daily production chain in CINECA.

The code has been instrumented to allow ParaView connection to the running simulation and extraction of a subset of a subset of the more than 50 variable to be visualized, see YouTube Summer of HPC movie [73].

4.1.3 *Big Data I/O optimization*

This pilot deals with large data handling issues that could possibly arise in post processing work-flows:

- Big Data Applications consume a lot of I/O bandwidth
- Most often I/O not optimized
- Optimization often difficult due to different optimization targets for different I/O patterns in one application

Here an automatic optimization tool is proposed, based on machine learning – **S**emi-**A**utomatically **I/O**-tuning framework (SAIO), which is developed at HLRS.

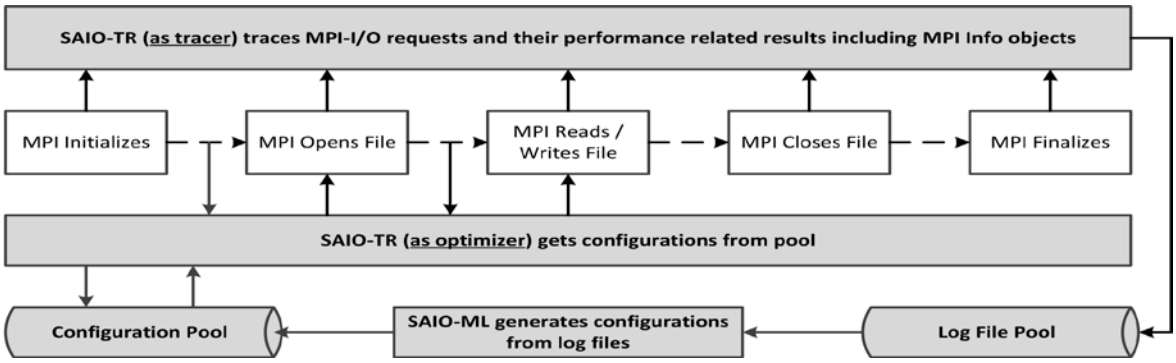


Figure 2: The schema of the proposed tools

The tool collects the data in real time execution from real use cases, so we need:

- Real-time I/O tracing and run-time optimization;
- Standalone machine learning process.

The following plot reports respective results of optimization applied to synthetic and real cases:

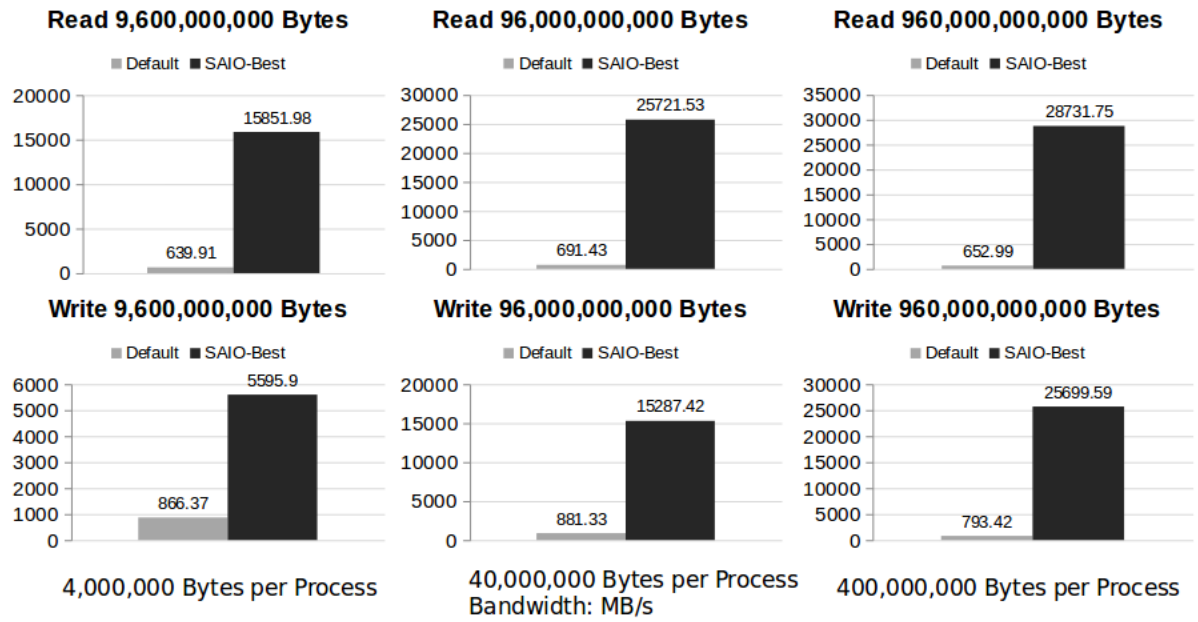


Figure 3: Synthetic case with 2400 MPI processes

The synthetic test case in shown in Figure 3. In all cases, SAIO finds better parameter sets for I/O than the default one which applications use if the user do not set any additional parameter for I/O. Figure 4 shows the application of SAIO to a post processing run of a user application. Here, 26% improvement in run time could be obtained.

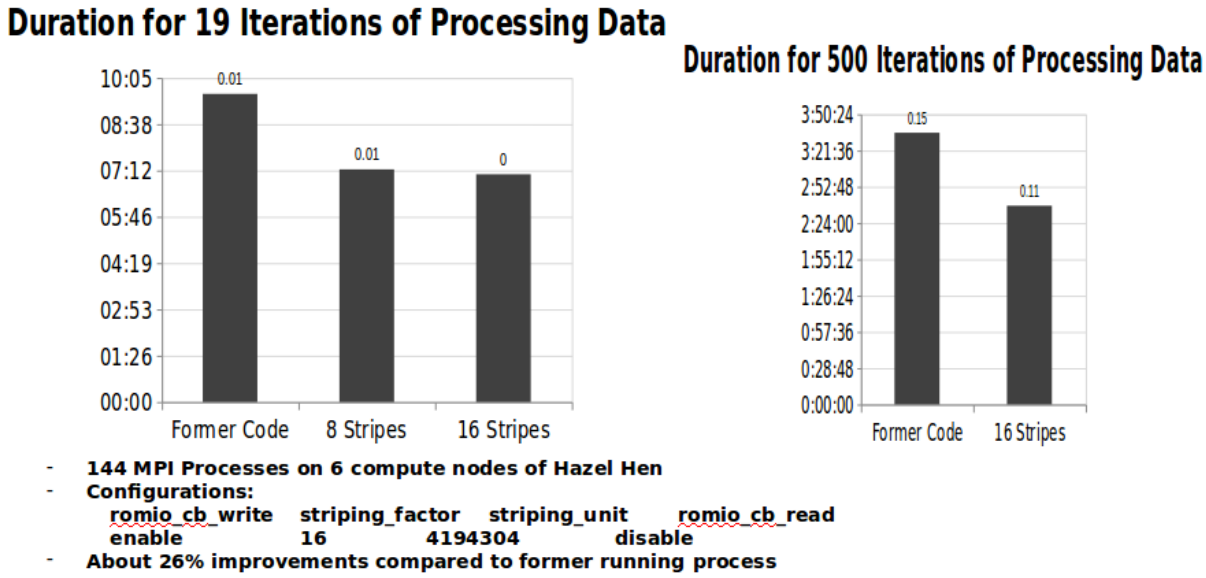


Figure 4: Real application with 144 MPI processes on six nodes of HazelHen

This pilot is presented in detail in a paper that has been submitted for publication at ISC 2017.

4.2 Experiences with the prototypical services

The pilots presented in this section are very different, regarding scope, diffusion and code maturity. A common approach that we used is the provisioning of repository of installation

recipes for all the base components used in the pilots. For Remote and In-Situ Visualization, we have used Spack HPC package manager [56].

4.2.1 *Remote Visualization*

CINECA Remote visualization service based on RCM (Remote Connection Manager) is used since for few years, but recently its user base has increased, reaching over 3 hundreds users on Galileo cluster.
As reported in the white paper [9], the system has been evaluated in different usage contexts. The most critical factor seems the available connection bandwidth with the remote user. Over LAN usage, there were almost no issue about performance, reports from WAN access seems to indicate that performances are perceived as satisfactory down to a good domestic ADSL (around 7 Mbits / sec). In few cases the reported severe usability issues where either connected to network congestion (domestic ADSL in the summer or small SME XDSL at work hours) or high latency connection (we had one bad feedback report when connecting from USA). The reports regarding access from other PRACE sites were satisfactory.
As reported in white paper [9], RCM is not used just for 3D interactive visualization but also as an alternative to ssh access that allows for GUI interfaces to work. Apart from user preferences, there are cases where GUI tools are really needed. For example, debugging sessions with TotalView or profile data analysis with Intel Vtune.
Some users are starting also to perceive it as an alternative access service: since Marconi cluster went in production, users started to ask for availability of RCM on it.
We have tried to evaluate the impact that a larger use of VNC sessions as an alternative to ssh could have over currently shared resources as login nodes. Regarding CPU usage, this is related to activity, so if the session is inactive, the CPU load is negligible.
In case of activity, there is an overhead related to image encoding on the node hosting VNC session as well as possible CPU load on the node (usually login frontend) that is supporting the ssh tunnelling.
This is again loosely related to activity, specifically the amount of screen memory that the applications are changing. This could have substantial variation, so precise measurement has not be carried on. Nevertheless, just looking at simple “top” report, the amount of CPU usage for ssh tunnelling was never greater than 20% of a core, usually lower than 5%. This means a worst case of more than 150 concurrent users on a single 36-core login Broadwell node, going up to 600 considering average usage.
Unfortunately, increasing in adoption has downsides: the number of ticket reporting problems and bugs has increased as well. The most frequent issues are:

1. Problems related to client platform incompatibilities, usually they arise on updates on Operating systems:
   - Recently Mac OS Sierra changed ssh client policy, user pass authentication problems.
   - Old Linux packaging included an X11 library that is incompatible with some new distributions like ArchLinux and Debian unstable.
   - On Windows, special characters in passwords are not supported.
2. Problems related to host environment and error reporting.
   - When login nodes are not available or not responding, the client does not report clearly the error.
   - Changes in login IP could need cleaning of local user session storage
   - When sessions are handled as batch jobs, user get unclear error messages if job scheduler is overloaded.
3. Problems related to applications
   - Some 3D applications (java 3d related) were not working properly under VirtualGL

- Marconi X11 / OpenGL software stack is not rock solid
- TurboVNC client-server reported to have issue with copy-paste and special keys.

The following table summarizes pros and cons of the Remote Connection Manager solution.

		Pro	**Cons**	**Improvements**
User experience	Client	The single executable client is handy for users, giving a zero-install experience	lack of web client-less (Http5) support, no Mobile client	Implement http client like Guacamole or NoVNC
	HPC site	Connections management made easy The interface hides tunneling complexity, job submission as well as VNC password setup	Lack web portal for session management	Deploy web portal submission
Deployment	Client	The cross-platform code compile single executable, including all dependencies, ready to run on bare OS	The wrapped executable is harder to debug; when things go wrong, no easy workaround	Distribute also source level client along with build instruction
	HPC site	User level deployment, no need for sysadmin install, apart from GPU drivers Use session storage in User Home Support different job schedulers Deploys on interactive login nodes	Window manager stack and application code difficult to build in user space: either rely on sys installed window managers or install really bare ones as Fluxbox	Use system software when container / VM technology is available Explore XPRA, that transport single windows and not entire desktops
Security	Client	The client does not store user pass, rely on consolidated ssh protocols	The wrapped client is not signed on windows and Mac OSX The build environment is not well defined for Windows and OSX.	Sign the wrapped app Use Virtualization tools like Packer / Docker to define client build environment
	HPC site	All session related step run in non-privileged mode, the “surface of attack” does not increase	VirtualGL Sessions running on same node are not completely decoupled:	Use container / VM
Stability	Client	RCM wrapped code support auto-update to ensure compliance to server	Bugs and errors related to Client OS Platform are difficult to reproduce and fix	Improve client code error handling Set up multi-platform testing environment for clients
	HPC site	Ensure proper batch job session handling,	Unstable when scheduler or login hangs	Improve server code error catching and handling
Performance		Satisfactory performance on LAN and good quality Home ADSL	Non state of the art full video compression (No H264)	Fund H264 integration in TurboVNC / VirtualGL [60].
Maintenance costs		All software stack components are release as Open Source	Maintaining and supporting cross-platform client code require considerable effort	Share the effort between HPC centers, using common code development, building and testing infrastructure
Flexibility	Client	Allow integration of other tools into client	Difficult to maintain different client versions	Set up automatic client build infrastructure
	HPC site	Support nodes without GPU	Do not support easy setup of predefined sessions	Implement hierarchical configuration files, let users override session and job configuration files
	Resource manager	Adapt to different job management policies (interactive login, reserved queues, normal queues)	Site configuration of session jobs is not flexible	

Table 3: Pros and cons for the Remote Connection Manager pilot

4.2.2 *In-Situ Visualization experiments*

The in-situ approach has proved to be feasible but it still require quite amount of effort for effective adoption by simulation codes developers.

In order to simplify adoption and experimentation, recipes for deployment of ParaView with Catalyst have been included in the previous RCM recipes repository.

	Pro	Cons	Improvements
Adoption	Catalyst leverage ParaView system for in-situ visualization	Single application library Does not apply to closed source code	Other approaches who concentrate on I/O as Damaris promises to be application neutral
Flexibility	Catalyst can be integrated in different code: Fortran, C++, Python examples available	Catalyst impose ParaView visualization paradigm	
Ease of use	The required changes in application code are minimal and can be integrated in the simulation build process with configuration options	Code has to be instrumented Adaptor code can be tricky, especially if simulation data is distributed in a way that is not compatible with ParaView need	
Performance	Catalyst reduce the need of I/O and can leverage application parallelism pro post-processing	The process of adapt simulation data to ParaView structures can be time consuming, especially when reorganization between cores is needed	Catalyst recently introduced zero-copy adaptor support
Overhead	If non activated, overhead is negligible	When activated, in-situ post-processing could use in a very inefficient way the resources that are tuned to the simulation code	Move to in-transit visualization paradigm, (in situ with coupled ParaView server)

Table 4: Pros and cons for the In-Situ Visualization pilot

4.3 Recommendations for the next implementation phase

Due to the nature of the pilots, and to the time constraints, none of the pilots have reached the maturity to be proposed as a common service to be put in production.
Different centres have still many differences in the environment and take decision regarding production software stack in a completely independent way, making hard time for centralized decisions to be "bought in" by the different centres.
A preliminary step before defining a production service could be setup of interest group and provisioning of shared repository of deployment recipes of all the open source components.

4.3.1 *Remote Visualization*

This kind of service is increasingly deployed and its importance is widely acknowledged. In CINECA the service will be supported and extended to forthcoming clusters. Currently other centres provide similar services, sharing similar technologies and policies.
Nevertheless, at present time, further effort is needed to formalize a request for the service to be inserted between required services; furthermore, it is unlikely that all centres would shortly adopt a single technology.
Our proposal is to continue the pilot phase focusing on following activities:

1. Set up an interest group on remote visualisation technology to promote knowledge sharing, technology watch and cross-evaluation of services;
   This group will track technology changes, especially ones regarding virtualisation and cloud technology as container and visualisation technology are finding their way through HPC centres and will likely have significant impact on remote visualisation services by opening new opportunities.
2. Set up a common repository of deployment recipes of the different Open Source components used in remote visualization systems: initial content will be formed by the current recipes for CINECA service. Components used in other PRACE centres for remote viz, will be included and an effort to harmonize deployment will be carried on, so to produce a consistent set of compatible components. Possible candidates are NoVNC, Xpra, Strudel.
3. If possible, continuous integration facilities in PRACE (see [84]–[85]) will be used for supporting automating building and testing of the collected components.

4. Goal of next phase would be to collect in the repository the deployment recipes of the remote visualization services in production in at least three PRACE centres.
5. It would be important to keep contact with security group, in order to assess deployed technology from the security point of view.

4.3.2 *In-situ visualization*

1. The evaluated technology requires application code instrumentation, so it does not directly fit into the general service pattern. Nevertheless, there is an increasing interest in this approach to cope with data upscaling. Our suggestion is to:
   - Set up an interest group on in-situ and large data visualization technologies, including web based technologies such as ParaView Cinema to promote knowledge sharing and technology watch.
   - Set up a common repository of deployment recipes of Open Source tools and libraries for in-situ and large data visualization. This should include test, tutorials and openly available example codes.
   - Promote inclusion of in-situ technologies in community driven, open source codes (one example could be Open Foam application).

4.3.3 *Large data optimization*

- Further testing in both core functionalities as well as in real world application;
- Provisioning of the core tool as well as tutorial as deployable recipes.

5 Provision of repositories for European open source scientific libraries and applications

5.1 Description of the service

Service 4 of Task 6.2 aims to provide a series of repositories for European open source scientific libraries and applications, and focuses in the wide adoption, uniformity and consolidation of European products.
This service must provide enough tools to satisfy a wide range of needs and requirements for different projects and interests but at the same time, it must help to consolidate European products providing uniformity and consistency.
The proposed solution is to deploy a modern, useful and featured tool for code repository that will serve as the core for the solution. Around this core, different complements will be deployed and will serve as key elements to achieve the required wide adoption and uniformity. Some of these complements will consist for example in a project management tool, a bug tracker, a continuous integration system or a knowledge database facility.
The core of the service has been decided to be based on GitLab given the analysis of current technologies and possible features studied in this PRACE-4IP project. Other components are based also on open source software and are TRAC, Redmine, Jenkins, CASino and EUDAT B2SHARE.

5.2 Pilot – PRACE Repository Services

The pilot will consist in the deployment of a main entry point webpage from where all the services could be accessed at:

https://repository.prace-ri.eu/

For the full service, the following features will be covered by its respective software:

1. Code Repository: GitLab PRACE
2. Project Management & bug tracking: TRAC
3. Project Management & bug tracking: Redmine
4. Continuous integration system: Jenkins
5. Continuous integration system: GitLab PRACE
6. Account management: LDAP
7. Single Sign On to all services: CASino
8. Knowledge database: EUDAT B2SHARE Services

Moreover, a Service policies section has been analysed and documented, which will be described in the next paragraphs.
The purpose of this pilot is to get an overview of how the solution would work and satisfy partner's needs with a real structure and with different accounts for different kinds of actors.
Previous solutions and state of the art were examined and commented in Deliverable D6.3 [2], so we will not extend here the reasons of why we have chosen each particular software, but instead, we will comment the functionalities provided by each one of these services:

5.2.1 *Code Repository: GitLab PRACE*

GitLab is the core of this service and a web-based Git repository manager with wiki and issue tracking features. GitLab offers hosted accounts similar to GitHub, but also allows its software to be used on third party servers.

1. Features
   - Git repository management
   - Code reviewer tool
   - Bug/Issue tracking tool
   - Activity feeds
   - Integrated Wiki
   - GitLab CI for continuous integration and delivery.
   - Open Source: MIT licensed, community driven, 700+ contributors, inspect and modify the source, easy to integrate into your infrastructure
   - Scalable: support 25,000 users on one server or a highly available active/active cluster
2. Roles and Permissions
   - Types of permissions
     Breakdown of permissions are accessible in GitLab documentation [79]. Structure is the same than the one presented for Owner-Admin-Write-Read permission for GitHub.
   - Groups
     Groups can have different members with different permissions. When multiple projects are assigned to the same group, the members will have the same permission for all the projects [80]. One can promote specific members of the group for specific projects by adding them also as a member of a project.
3. Interface and Access: the interface and access methods are exactly the same than the specified for GitHub.
4. Costs: Hosting GitLab on a server operated by PRACE:
   - Hosting fee / Costs: FTE / recurring cost is necessary for GitLab instance installation and first configuration, which is important before first usage of repository.

- Administration costs: required manpower effort approximately 0.3 FTE for managing users, groups and GitLab framework. This level of effort translates to 3.6 PM per year of service operation.

5. GitLab Pros, based on requirements of PRACE:
   - Not depending on an external enterprise service
   - Can really say it is a separate PRACE service/repository
   - Can be PRACE branded with look and layout changes for each subpage (CSS, HTML5), other PRACE services can be linked from / integrated
   - Absolute freedom of configuration, installation of application addons and freedom of group management and private repositories
   - Mobile apps
   - Option to integrate with LDAP / connect with PRACE LDAP and use PRACE userbase
   - Integration of data into other sites (e.g. PRACE web, training portal, etc.) is possible and customizable
   - Builtin advanced wiki features that can be updated with git
   - Powerful import features from GitLab
   - Gravatar integration allows using the same avatar used on github
   - Unlimited public/private repos without the need of upgrading plans
   - Integration option with GitLab ci to test, build and deploy code snippets
   - UI is very similar to GitHub, users are familiar with it.
   - Possibility to use federated login, like edugain (using SSO method), auth edugain members seamlessly
   - Advanced Jira Support, Jenkins support
6. GitLab Cons:
   - One-time effort of deployment and configuration
   - Requires operation effort to run (these two however might be covered as a planned WP6 effort)
   - Requires hosting (there are numerous PRACE services hosted by PRACE partners independently from IPs)

5.2.2 *Project Management & bug tracking: TRAC*

Trac is an enhanced wiki and issue tracking system for software development projects. Trac uses a minimalistic approach to web-based software project management. It helps developers write great software while staying out of the way. Trac should impose as little as possible on a team's established development process and policies.
It provides an interface to git, an integrated Wiki and convenient reporting facilities.
Trac allows wiki markup in issue descriptions and commit messages, creating links and seamless references between bugs, tasks, changesets, files and wiki pages. A timeline shows all current and past project events in order, making the acquisition of an overview of the project and tracking progress very easy. The roadmap shows the road ahead, listing the upcoming milestones.
Main components are:

1. Wiki subsystem: TracWiki: built-in Wiki
2. Version Control subsystem:
   - TracBrowser: Browsing source code with Trac
   - TracChangeset: Viewing changes to source code
   - TracRevisionLog: Viewing change history

3. Ticket subsystem:
   - TracTickets: Issue tracker
   - TracRoadmap: Tracking project progress
   - TracReports: Writing and using reports
   - TracQuery: Executing custom ticket queries
   - TracBatchModify: Modifying several tickets in one request
4. Other modules:
   - TracSearch: Full text search in all content
   - TracTimeline: Historic perspective on a project
   - TracRss: RSS content syndication
   - TracAccessibility: Accessibility keys

5.2.3 *Project Management & bug tracking: Redmine*

Redmine is a flexible project management web application. This can accomplish similar points of TRAC features. We are going to propose to include both softwares in a productive version of the service since different partners are using different tools.

Some of the main features of Redmine are:

- Multiple projects support
- Flexible role based access control
- Flexible issue tracking system
- Gantt chart and calendar
- News, documents & files management
- Feeds & email notifications
- Per project wiki
- Per project forums
- Time tracking
- Custom fields for issues, time-entries, projects and users
- Git integration
- Issue creation via email
- Multiple LDAP authentication support
- User self-registration support
- Multilanguage support
- Multiple databases support

5.2.4 *Continuous integration system: Jenkins*

Jenkins is a self-contained, open source automation server, which can be used to automate all sorts of tasks such as building, testing, and deploying software. In this aspect is similar to the features provided by GitLab integrated CI, but it is a dedicated and very powerful tool widely used by many projects. So it is important to provide this tool.

5.2.5 *Continuous integration system: GitLab PRACE*

GitLab has integrated CI (continuous integration) and CD (continuous deliver) to test, build and deploy code.

- Multi-platform: builds can be executed on Unix, Windows, OSX, and any other platform that supports Go.
- Multi-language: build scripts are command line driven and work with Java, PHP, Ruby, C, and any other language.
- Stable: your builds run on a different machine than GitLab.
- Parallel builds: GitLab CI splits builds over multiple machines, for fast execution.
- Realtime logging: a link in the merge request takes you to the current build log that updates dynamically.
- Versioned tests: a .gitlab-ci.yml file that contains your tests, allowing everyone to contribute changes and ensuring every branch gets the tests it needs.
- Pipeline: you can define multiple jobs per stage and you can trigger other builds.
- Autoscaling: you can automatically spin up and down VM's to make sure your builds get processed immediately and minimize costs.
- Build artefacts: you can upload binaries and other build artefacts to GitLab and browse and download them.
- Test locally there are multiple executors and you can reproduce tests locally.
- Docker support: you can use custom Docker images, spin up services as part of testing, build new Docker images, even run on Kubernetes.

5.2.6 *Account management: LDAP*

In this case we used LDAP in order to provide authentication and account management. The deployment has been based on a standard OpenLDAP installation with default schemas. In order to ease the management of the LDAP tree, LDAP Account Manager (LAM) has been also deployed in the server. LAM provides a nice web interface from where you can add or remove users, manage groups, etc. This tool is very convenient since it is also able to execute particular scripts when users or groups are created or modified, thus creating base directories, projects, etc.

5.2.7 *Single Sign On to all services: CASino*

One of the main concerns of having multiple components is the fact that the user must log in on each service if a special system is not configured. This is the purpose of the Single Sign On (and sign-out) software. This software will allow the user to enter his username and password only once, and then be automatically logged on each of PRACE repository services. When logged out same thing must happen, the user is logged out of all PRACE repository services.

The elected solution has been one of the easiest to install available nowadays, it is CASino. Some tests has been made with more complex ones, like Jasig CAS, but complexity was not worth the purpose of the pilot.

5.2.8 *Knowledge database: EUDAT B2SHARE Services*

One extra point to include in the PRACE repository service was a demand from WP3 and WP4 to have a place to write and publish whitepapers, documentation, public documents, etc. It turned out that all of these deeply analysed requirements were perfectly provided by EUDAT services, specially by the B2SHARE tool. In the All Hands Meeting in Athens, on Feb. 2017, a presentation by an EUDAT collaborator was been made to WP6 and demonstrated that software was perfect for the needed purpose.

B2SHARE is a user-friendly, reliable and trustworthy way for researchers, scientific communities and citizen scientists to store and share small-scale research data from diverse contexts.

These are the features of EUDAT B2SHARE:

- Store: facilitates research data storage
- Preserve: guarantees long-term persistence of data
- Share: allows data, results or ideas to be shared worldwide
- integrated with the EUDAT collaborative data infrastructure
- free upload and registration of stable research data
- data assigned a permanent identifier, which can be retraced to the data owner
- data owner defines access policy
- community-specific metadata extensions and user interfaces
- openly accessible and harvestable metadata
- representational state transfer application programming interface (REST API) for integration with community sites
- data integrity ensured by checksum during data ingest
- professionally managed storage service – no need to worry about hardware or network
- EUDAT user support
- monitoring of availability and use

At this point we have taken an overview of the main functional components of the pilot. We are going to explain the non-functional parts regarding policies.

1. **Service policies:** We divided the service policies in five subgroups and discussed about the best choices to take into consideration. We give here the description of each policy type and then in the next sections we make the recommendations of how to apply them.
2. **Access policies:** Defines who/what projects can access the system and from what countries/institutions. Take in mind that different privacy policies can be applicable coming from different countries.
   It is necessary to define a guideline of which projects can be granted access to the PRACE repository and which don't.
3. **Authentication Policies:** The concerns of this guideline must specifiy whether users should use passwords to authenticate to the service, or whether they should use certificates, or both methods should be allowed. Also the definition of how often do they expire, when an access is revoked, what are the default permissions on different projects, who can be the manager of different projects, etc.
4. **Accounting Policies:** What information can we store, how much time, under what laws and what we require to create an account. I.e. how we demonstrate that a request is a valid one?
5. **Usage Policies and Data Policies:** Once a user has been granted access to the repository, there should be some policies in order to explicit define what kind of information can be uploaded. Data should be scientific, technical or management content. An example of not wanted content could be “photographs of last All-Hands meeting”. So from this documentation must define intended and prohibited uses and respond to questions like: Is it possible to upload any kind of data? What are the allowed trademarks, copyrights, data owners, permissions? What happens with private projects?

6. **Terms and Conditions – Samples:** It is also important to define a standard terms and conditions regarding the service level agreement that the service provides: reliability, uptime time per cent, data and project backups, virus/malware scan, etc. We provide a first draft in the pilot documentation [90].

5.3 Experiences with the prototypal services

After having deployed all the commented software and decided about the main points of the policies, we got some feedback of interested partners and projects. Thanks to that we are in position to evaluate the pilot.

We present a brief PROS&CONS table of what we think are the main important points of the election that we have taken.

5.3.1 *Feedback from Codevault:*

We also received some feedback from Codevault team that started to use the service. Their main concern is about the access to the repository. They see the service as a great tool but they want to be sure that all the students can access the codes that are uploaded. Also, there is the concern of how to manage the accounts of project managers that has to be able to define read permissions for ones, and write permission for others on the given repositories. There are other minor problems that they detected and that must be taken into consideration for the transition to a production service. We will enumerate them in the next section.

5.3.2 *Feedback from interested partners:*

We raised a survey to PRACE mailing lists and asked to different CoEs and projects. We got positive interest from different projects and after the presentations in the AHM on February 2017, more people got interested in the usage of the service. Main directly contacted partners are:

- BioExcel CoE – KTH Sweden
- EoCoE Energy oriented CoE – CEA France
- NOMAD Novel Materials Development – MAX-PLANCK Institute, Germany
- MAX materials design at the exascale – CONSIGLIO NAZIONALE DELLE RICERCHE, Italy
- EsiWace – Earth System Modeling for Weather and Climate – DEUTSCHES KLIMARECHENZENTRUM GMBH, Germany
- An e-infrastructure for software, training and consultancy in simulation and modelling – EPFL Lausanne, Switzerland
- POP, Performance Optimization and Produictivity CoE – BSC, Spain
- CoEGSS Global Systes Science – HLRS, Germany
- PRACE Mailing List
- Codevault
- Project Alya - BSC

The conclusions of the survey are shown below based on 9 responses:

Question	PUNCTUATION (over 5)
General service interest	3.8
Public Code repository feature	4
Private Code repository feature	3.1
Continuous Integration tool	3.3
TRAC Project Management/Issue Tracking	3
Redmine Project Management/Issue Tracking	2.2
Special functionalities you would like	Two-factor authentication
Nº of projects you would host	2
Nº of users that would use the service: from	1 to 100

Table 5: Survey about PRACE Service Repo Pilot

From our point of view, the conclusion is that the service has potential to be and the selected technologies are suitable for the needs of the asked people. The less punctuated one has been the Redmine Project Management tool, but since there are interest from some partners and it is not a difficult tool to maintain, we will recommend to keep it.

5.4 Recommendations for the next implementation phase

Based on evaluation and the feedback we received the following propositions are made.
Having a PRACE Repository Services in PRACE-5IP like the one described in this document is a good point. We recommend to continue with the server and to start the transition to a production and a regular service in PRACE-5IP. There are lots of people interested in hosting their codes in a central PRACE facility, and this service is exactly that.
Moreover, the benefits of having multiple tools will provide a wide adoption since users will be able to choose what they are used to or what fits better their needs.
Open Source must be the premise of this repository and open licences must be recommended or a must to upload contents in this service.
Finally, all the components of the service must be equally branded to be shown as a PRACE service. In this way we will get the uniformity of one single service that will promote the seamless usage to all collaborators.
Based on all the feedback received, comments, questions, etc. we do the more specific and technical considerations that have to be added and taken into account in the productive service:

- Provide backup: In the pilot we didn't provide a solution for backing up data. It is mandatory to have one.
- Provide High Availability (HA): Granting a good uptime is mandatory for a wide used service. A HA solution should be installed and the system should be designed to provide this feature. Our recommendation is to use at least two servers with shared storage and an HA software like Pacemaker.
- Capacity of the service: It can start with low resources since we have seen that there is no much consumption when having all the systems up&running. But of course we could not test an intensive use. Scalability is a must in this case.
- Provide PRACE network connectivity: Some users commented that it would be useful that PRACE Gitlab, Jenkins, and other software to be connected to the internal PRACE Network. This would give the possibility to for example gather codes from interned

restricted login nodes. This is a very interesting feature that has to be added to a productive service.
- Provide different machines for CI: In order to test codes compilation and testing in different architectures, it would be needed to get some centers to collaborate and provide some login nodes to act as slaves of CI systems (Jenkins and GitLab).
- Access committee must be created: An access committee is one of the most important parts of this service. This committee must ensure that the users given access are trustworthy ones, and must keep track of the accounts and users permissions. It must also decide on political aspects of the service.
- Operator team must be created: After the committee has granted access to a project, an operator must create the accounts and inform the user that he has been granted. All of this can be automatized but there should be a responsible.
- Administration team: Upgrades, maintenance, and problem resolution must be assigned to another team.
- Usage manuals must be created for each service.
- Knowledge Database: From all the software analysed, the one that best fits the needs of the service is B2SHARE. A collaboration must be started with EUDAT and PRACE to see the possibilities of integration, and if it is possible, to add the PRACE brand to EUDAT product.

5.4.1 *Service policies:*

From the pilot team we recommend the creation of an access committee for the PRACE Repository Services service that evaluates the feasibility for the inclusion of the project/content into the repository and that decides about the main concerns of what data can be uploaded and what is the best use of it.
This access committee must also provide the steps to consolidate the PRACE repository as a valuable resource for the institution and for the research in general. That means that has to manage the service and align it with the needs addressed from the PMO.
The main aspects that this committee should agree on are:

5.4.2 *Access policies*

We initially provide some basic ideas on which to decide if a project must be granted access or not:

- All PRACE projects have access to the repository by default
- All PRACE researchers/staff can ask for new projects
- All PRACE users can ask for an account
- External PRACE collaborators can ask for an account
- CoE can ask for an account
- Authentication Policies

In the pilot, the authentication used has been password method. We added the feature to the LDAP server to store certificates in every personal account but we did not implement on each service. The recommendation from the pilot is to keep passwords as an authentication method as it is an easier way of usage for everybody. A global change password page must be provided then.
The passwords are recommended to have an expiration of 2 years, this will be controlled in a standard LDAP field and a warning script could be deployed to send a warning to the user.

Accounts must be personal and projects will have r/o and r/w users, with r/w users being responsible for the repository.

5.4.3 *Accounting Policies*

The property and license of the code that is hosted in PRACE repository should be defined when the general terms are written when entering in production. We propose to force the usage of Creative Commons, Copyleft, GPL licences or similar that grants the freedom of distribution, share and modification whenever the original authors are cited.
The duration of the project, the backups, the size quota, etc. has to be defined by the access commite when allowing the project to be hosted in the service and it has to be registered in some database, possibly making a technical extension to the service if the number of projects increases, creating a form for asking a project and an automatic creation and maintenance system.

5.4.4 *Usage Policies and Data Policies*

Defining usage and data polices corresponds to the PMO and the requirements will appear once the service starts to be used intensively. At this point, it is difficult to imagine all possible scenarios, so in general terms we suggest a guideline that will help to decide in further occasions:

- The uploaded contents must support research, and especially European research.
- It is not permitted to use the repository and services for non-research and personal or company profit only.
- The service is intended to share the knowledge within research world and to provide value to the community. This means that the authors, owners, licenses, copyrights, etc. must be named and respected but the idea is to share and open the contents.
- Projects that deviate from these policies can still be approved to use the repository under the supervision of the access committee

5.4.5 *Terms and Conditions - Samples*

It is also important to define a standard terms and conditions regarding the service level agreement that the service provides: reliability, uptime time per cent, data and project backups, virus/malware scan, etc. We provide a first draft in the pilot documentation [90].

5.4.6 *How to proceed with the transition:*

We propose the following steps.

1. Create an access committee that will take political decisions about the service
2. Define the final policies of the service
3. Define the procedures of accounts and projects creation and management
4. Assign new teams for implementation/operation
5. Start collaborating with EUDAT in defining the PRACE project in B2SHARE
6. Start the design of a reliable service with HA, Backups, and the following components:
   - Gitlab and Gitlab CI
   - Redmine
   - Trac
   - Jenkins

- SSO
- Accepted solution for accounts

7. Implement and test everything
8. Add Codevault as first member for general service usage
9. Add WP3 as first member for B2SHARE service usage
10. Get feedback, improve and test
11. Final announcements

5.5 First draft of KPIs

In order to keep track of good usage of the service, here is a first proposal of KPIs that we should track:

1. Number of projects hosted in each service
2. Number of total vs active users
3. Storage of data hosted
4. Accesses per day per service
5. Type of uploaded projects (public/private)
6. Number of read-only vs read-write users per project
7. Number of incidents
8. Invested time on management
9. Uptime of the service

6 Conclusion

In this deliverable we have presented results of work done within Task 6.2 of Work package 6 of PRACE-4IP project. We have considered four new services that were the PRACE's answer to raised needs in research or even wider community: (i) The provision of urgent computing services where the emerging computational results can help to issue critical decision-making paths in the case of a critical, national-scale emergency; (ii) The new links with large-scale scientific instruments that are providing a large amount of data and information which require an improved support of data intensive applications; (iii) The smart post processing tools including remote and in situ visualisation and (iv) The provision of repositories for European open source scientific libraries and applications.

The partners that have had assigned PMs in this task were grouped into four groups – one for each service. The main part of each service was development of at least one pilot/prototype to experiment the service in real environment. In total there were ten pilots which provided clear picture about opportunities and weaknesses related to issues underlying each service. Additionally, we have proposed also a first draft of KPI that could be used to monitor the services.

For the urgent computing service, we need first a selection of applications that have status "Urgent". They have to be installed on dedicated EXEC Host, respecting all licencing issues. The user of each such application can operate in normal PRACE Project mode, as DECI project, etc. But when urgent situation starts, the user can start running the urgent computing application. The status of other applications, i.e. the protocol how the other running jobs are rescheduled shall be clear for each EXEC host and defined in appropriate policy. In the deliverable we explained few possibilities for such policy. We also proposed KPIs related with each urgent application. All KPIs must refer to the values agreed in SLA upon agreement on usage of this application. Important outcome of Service 1 is also that PRACE partners are capable offering urgent computing services but the real needs for such services are small so further development of this service makes sense only if we detect partners with real such needs.

In the service about linking large-scale scientific instruments to the HPC ecosystem we have identified two main areas, in which scientists and infrastructure operators could benefit from a collaboration with PRACE:

1. Introduction of 'institutional' access to PRACE HPC infrastructure to offload data analysis work from the instrument centres, which often don't have the necessary capacity.
2. Development of modern tools that would automatize complex workflows involving data transfer and compute job submission and monitoring.

We have decided to concentrate on providing solutions to smaller sub-problems of the individual scientific groups. We have developed and tested the following data transfer services:

- Using gsatellite for scheduling of data transfer and compute jobs
- Dealing with large numbers of small files in HPC environments
- New framework data packaging and transfer
- HTTP for bulk file transfer over high latency network links

Our experience with the pilot deployments shows that the developed tools can become useful parts of scientific workflows in the future. In many cases the tools must be adopted by the users, and not by the infrastructure operators (e.g., data transfer framework, gsatellite). Hence, their adoption will depend on the willingness of the users to modify their existing workflows, and that takes a lot of time. The newly developed HTTP transfer tools could become a viable alternative to established tools like GridFTP in the future, but they still require significant development. While it is important to further invest into the development of those tools in PRACE-5IP and advertise them to the users, with the exception of gsatellite which cannot be deployed as a PRACE service.

Three out of five collaborating instrument partners indicated their interest in obtaining regular access to additional compute power at the PRACE facilities. They do not apply for the resource allocation under PRACE Project Access calls, but the computational work is nonetheless decisive for the research. One of our conclusions is also that during PRACE-5IP a possibility for institutional access of large scale scientific instruments to PRACE infrastructure should be investigated.

The work on third service showed that different centres have very different practices related to smart visualization tools. We suggest as a preliminary step (before defining a production visualization service) to setup an interest group which would: (i) promote knowledge sharing, technology watch and cross-evaluation of services; (ii) provide a shared repository of deployment recipes of all the open source components for smart visualization and (iii) promote inclusion of smart, especially of in-situ visualization technologies in community driven, open source codes.

We suggest that in the PRACE-5IP project we collect in such repository the deployment recipes of the remote visualization services that are in production in at least three PRACE centres. Another important task is maintaining contacts with the security experts, in order to assess deployed technology from the security point of view.

The pilot developed within the last service turned out to be matured enough to go to production as regular WP6 service. We suggest to make this transition within Task 6.2 of PRACE-5IP. Open Source must remain the premise of this repository and open licences must be recommended or a must to upload contents in this service. Finally, all the components of the service must be equally branded to be shown as a PRACE service. In this way we will get the uniformity of one single service that will promote the seamless usage to all collaborators.