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

D6.3
Analysis of New Services

Final

Version: 1.0
Author(s): Janez Povh, Andrej Kastrin, FIS; Miroslaw Kupczyk, PSNC; Luigi Calori, CINECA; Marcin Krotkiewski, UiO; Felip Moll, BSC; Andreas Panteli, CaSToRC; Zoltan Kiss, NIIF; Nevena Ilieva-Litova, NCSA

Date: 30.11.2015

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: EINFRA-653838**	
	Project Title: PRACE Fourth Implementation Phase Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: < **D6.3**>	
	Deliverable Nature: < Report >	
	Dissemination Level: PU*	**Contractual Date of Delivery:** 30 / November / 2015
		Actual Date of Delivery: 30 / November / 2015
	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: Analysis of New Services**	
	ID: D6.3	
	Version: <1.0>	**Status: Final**
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2010	
	File(s): D6.3.docx	
Authorship	**Written by:**	Janez Povh, Andrej Kastrin, FIS; Miroslaw Kupczyk, PSNC; Luigi Calori, CINECA; Marcin Krotkiewski, UiO; Felip Moll, BSC; Andreas Panteli, CaSToRC; Zoltan Kiss, NIIF; Nevena Ilieva-Litova, NCSA
	Contributors:	Damian Kaliszan, PSNC Huub Stoffers, Surfsara Nial Wilson, ICHEC Oguzhan Herkiloglu, UHEM David Hrbac, IT4I-VSB Thomas Bönisch, HLRS Leon Kos, UL Kriakos Gkinis, GRNET Andrew Turner, EPCC Ioannis Liabotis, GRNET Thomas Röblitz, UiO/SIGMA2 Giovanni Erbacci, CINECA
	Reviewed by:	Herbert Huber, LRZ Thomas Eickermann, FZJ
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	29/September/2015	Draft	Adapted template for the contributors
0.4	2/November/2015	Draft	First composed version (for the service and WP6 coordinators)
0.5	8/November/2015	Draft	Second version for the for the service and WP6 coordinators
0.6	9/November/2015	Draft	Version for the for internal reviewer
0.7	18/November/2015	Draft	Version sent to service coordinators to provide a revision
0.9	23/November/2015	Draft	Version sent to MB/TB
1.0	30/November/2015	Final version	

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 Tables

References and Applicable Documents

[1] http://www.prace-project.eu
[2] 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.
[3] See https://twiki.cern.ch/twiki/bin/view/Geant4/MultiThreadingTaskForce, A. Dotti at al.
[4] 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.
[5] 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

[6] SPRUCE resources, http://spruce.teragrid.org
[7] S.H. Leong, A. Frank, D. Kranzlmüller, Towards a general definition of Urgent Computing, International Conference on Computational Science, 2015
[8] 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
[9] S.V. Ivanov, S. V. Kovalchuk, A.V. Boukhanovsky, Workflow-based Collaborative Decision Support for Flood Management Systems, International Conference on Computational Science, 2013
[10] 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
[11] 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
[12] 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
[13] S.H. Leong, A. Frank, D. Kranzlmüller, Leveraging e-Infrastructures for Urgent Computing, International Conference on Computational Science, ICSS 2013
[14] 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
[15] 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
[16] 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
[17] 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
[18] Online announcement of the ICCS Workshop on Urgent Computing, June 2012, on the website http://dice.cyfronet.pl/events/iccs_urgent_computing_workshop
[19] 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
[20] 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
[21] 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
[22] Mudge S. (Editor), Methods in Environmental Forensics, CRC Press, 2009
[23] 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
[24] Scientific computing world web magazine: Importance of remote visualization for HPC: http://insidehpc.com/2015/02/hpcs-future-lies-in-remote-visualization/
[25] Nvidia blog VMD-NAMD: GPU based coupling simulation and visualization

http://devblogs.nvidia.com/parallelforall/hpc-visualization-nvidia-tesla-gpus/
[26] Nvidia blog In-situ visualization http://devblogs.nvidia.com/parallelforall/interactive-supercomputing-in-situ-visualization-tesla-gpus/
[27] Paraview Home page: http://www.paraview.org/
[28] Visit Home page: https://wci.llnl.gov/simulation/computer-codes/visit/
[29] VMD Visual Molecular Dynamics home page: http://www.ks.uiuc.edu/Research/vmd/
[30] NICE Desktop Cloud Visualization: https://www.nice-software.com/products/dcv
[31] TurboVNC home page: http://www.turbovnc.org/Main/HomePage
[32] VirtualGL project page : http://www.virtualgl.org/
[33] CINECA Visualization service: http://www.hpc.cineca.it/services/remote-visualisation
[34] IT4I Anselm Cluster Visualization service: https://docs.it4i.cz/anselm-cluster-documentation/remote-visualization
[35] LRZ SuperMUC visualization service: https://www.lrz.de/services/v2c_en/remote_visualisation_en/overview_en/
[36] web based HTML 5 VNC access to LRZ Visualization service https://www.lrz.de/services/v2c_en/remote_visualisation_en/web_interface_en/
[37] HLRS Cray XC40 Hazel Hen visualization setup: https://wickie.hlrs.de/platforms/index.php/CRAY_XC40_Graphic_Environment
[38] IT4I Salomon Cluster Hardware description: https://docs.it4i.cz/salomon/hardware-overview-1
[39] noVNC Project home page : http://kanaka.github.io/noVNC/
[40] Pyinstaller Project home page : http://www.pyinstaller.org/
[41] CINECA RCM help page : http://www.hpc.cineca.it/content/remote-visualization-rcm
[42] RCM current code base : https://hpc-forge.cineca.it/svn/RemoteGraph/branch/multivnc/
[43] EasyBuild Project home page : http://hpcugent.github.io/easybuild/
[44] Guacamole Project home page : http://guac-dev.org/
[45] NVidia grid product : http://www.nvidia.com/object/grid-technology.html
[46] NVidia H264 hardware encoder : https://developer.nvidia.com/nvidia-video-codec-sdk
[47] DAMARIS Project library for data processing and in-situ visualization http://damaris.gforge.inria.fr/doku.php
[48] ParaView Catalyst in situ visualization : http://www.paraview.org/in-situ/
[49] Visit Libsim in situ visualization tutorial : http://www.visitusers.org/index.php?title=VisIt-tutorial-in-situ
[50] Fusion Forge, Software for managing the entire development life cycle of projects, http://fusionforge.org
[51] HPC Forge Project from CSCS - https://hpcforge.org/
[52] 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
[53] GitHub organization and teams online documentation, https://help.github.com/enterprise/2.0/user/categories/setting-up-and-managing-organizations-and-teams
[54] GitHub organization, roles and permissions, online documentation, https://help.github.com/enterprise/2.0/user/articles/permission-levels-for-an-organization-repository
[55] GitLab organization, roles and permissions, online documentation, https://GitLab.com/GitLab-org/GitLab-ce/blob/master/doc/permissions/permissions.md
[56] GitLab groups, online documentation, https://GitLab.com/GitLab-org/GitLab-ce/blob/master/doc/workflow/groups.md
[57] GitHub pricing, https://github.com/pricing
[58] Git Autodeploy, software for automatically deploying the latest version of your github project, https://github.com/olipo186/Git-Auto-Deploy/

[59] BuildKite, automation of software development processes, http://buildkite.com
[60] GitLab CI, GitLab continuous integration tool, http://doc.GitLab.com/ci/
[61] Jenkins CI, open-source continuous integration software, https://jenkins-ci.org/
[62] Atalssian Jira, Project Management tool, https://www.atlassian.com/software/jira
[63] Redmine, Open Source Project Management Tool, http://www.redmine.org
[64] Git, open source version control system, http://git-scm.com

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.
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
RI	Research Infrastructure
SCM	SCM - Source Control Management system, software for the managament 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 computer 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	Computation-based Science and Technology Research Center, Cyprus
CCSAS	Computing Centre of the Slovak Academy of Sciences, Slovakia
CEA	Commissariat à l'Energie Atomique et aux Energies Alternatives, France (3 rd Party to GENCI)
CESGA	Fundacion Publica Gallega Centro Tecnológico de Supercomputación de Galicia, Spain, (3rd Party to BSC)
CINECA	CINECA Consorzio Interuniversitario, Italy
CINES	Centre Informatique National de l'Enseignement Supérieur, France (3 rd Party to GENCI)
CNRS	Centre National de la Recherche Scientifique, France (3 rd Party to GENCI)
CSC	CSC Scientific Computing Ltd., Finland
CSIC	Spanish Council for Scientific Research (3rd Party to BSC)
CYFRONET	Academic Computing Centre CYFRONET AGH, Poland (3rd party to PNSC)
EPCC	EPCC at The University of Edinburgh, UK
ETHZurich (CSCS)	Eidgenössische Technische Hochschule Zürich – CSCS, Switzerland
FIS	FACULTY OF INFORMATION STUDIES, Slovenia (3rd Party to ULFME)
GCS	Gauss Centre for Supercomputing e.V.
GENCI	Grand Equipement National de Calcul Intensiv, France
GRNET	Greek Research and Technology Network, Greece
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 (3rd Party to GCS)
SIGMA2	UNINETT Sigma2 AS, Norway
SNIC	Swedish National Infrastructure for Computing (within the Swedish Science Council), Sweden
STFC	Science and Technology Facilities Council, UK (3rd Party to EPSRC)
SURFsara	Dutch national high-performance computing and e-Science support center, part of the SURF cooperative
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 (3rd Party to SNIC)
UnivEvora	Universidade de Évora, Portugal (3rd Party to UC-LCA)
UPC	Universitat Politècnica de Catalunya, Spain (3rd Party to BSC)
UPM/CeSViMa	Madrid Supercomputing and Visualization Center, Spain (3rd Party to BSC)
USTUTT-HLRS	Universitaet Stuttgart – HLRS, Germany (3rd Party to GCS)
VSB-TUO	VYSOKA SKOLA BANSKA - TECHNICKA UNIVERZITA OSTRAVA, Czech Republic
WCNS	Politechnika Wroclawska, Poland (3rd party to PNSC)

Executive Summary

In Task 6.2 of work package WP6 we analyse new services with particular focus on the prototypal implementations of these services at a pre-production level. At the end Task 6.2 will assess the functionality of these services and promote their adoption in the future if the analysis is a success.

Services which are analysed in this document are:

Service 1: The provision of urgent computing
Service 2: Links with large-scale scientific instruments
Service 3: Integrating tools and methods to simplify post-processing and visualization activities
Service 4: Provision of repositories for European open source scientific libraries and Applications

The work in Task 6.2 was coordinated by FIS. Partners working on this task were grouped in four subgroups, one for each new service.

In Deliverable D6.3 we consider all 4 services and describe the state-of-the-art in the corresponding area, the standards that the new services will rely on, the potential users of the new services and the relevant policies that influence the particular services. For each new service we also describe at least one pilot and define its main ingredients (specifications) as well as the main steps that will be done by the end of the project in order to implement this service at the prototypal level and perform its evaluation.

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. Examples of such services are:

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;
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;
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;
4. Provision of repositories for European open source scientific libraries and applications, to promote wide adoption, uniformity at consolidation of European products.

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 - Task 6.2 coordinator and started working at the beginning of June 2015:

1. Service 1: coordinator PSNC, other partners: CSC, SURFsara, UHEM, ICHEC;
2. Service 2: coordinator UiO-SIGMA2, other partners: CaSToRC, NCSA, NIIF;
3. Service 3: coordinator CINECA, other partners: HLRS, IT4I-VSB, UL FME;

4. Service 4: coordinator BSC, other partners: EPCC, GRNET, NIIF;

Service 1 is described in Section 2. At the end of this section, a description for a service pilot is provided for an urgent computing framework with a selected code from the PRACE benchmark suite which will be developed and deployed at selected Tier-1 and Tier-0 machines.

Service 2 is described in Section 3. Existing links between PRACE partners and the European Synchrotron Radiation Facility, the Large Hadron Collider, the Extreme Light Infrastructure and the Norwegian Sequencing Centre are described. For each instrument a pilot of a new link is presented.

Section 4 contains a description of Service 3. Improving session based VNC remote visualisation and pilot projects for parallel in-situ- and post-processing are presented.

In the last section we consider provision of repositories for European open source scientific libraries and applications. We concluded that GitLab is the best choice for the pilot implementation and present the detailed description on how to implement it as a pilot service.

2 Service 1: The provision of urgent computing

2.1 Description of the service

Urgent Computing (UC) is a relatively new and extensively explored approach to solve emergency tasks such as “digital” representation of various extreme and sudden weather conditions or natural phenomena (earthquakes, storms, floods) etc. using HPC resources. The UC term was introduced in 2006 by TeraGrid in the SPRUCE (Special Priority and Urgent Computing Environment) science project [6] to support broadly defined civil protection and crisis management.

While UC may mean different things to different people [7], most authors on the subject would agree that the purpose of UC is to support time critical decision making by means of computation, simulation and visualization. Politically or morally grounded criteria also seem to play a role demarcating UC and it could be argued they should: In so far as UC makes use of publicly funded resources, the public interest and public good of the causes it is applied to, should be beyond reasonable doubt. Civil protection against extreme weather and climate phenomena or diseases ranks high among the foreseen areas of application. By contrast, the time critical decision making, supported by computational prediction that goes on in dedicated systems that process massive amounts of stock exchange data, are generally not considered UC [7].

Remarkably absent from papers advocating the use of HPC resources in urgent computing use cases is the awareness of how basic resources such as power and cooling are provided in most HPC centres. If advance reservation is a requirement for the scheduling environment, then surely it will also be pertinent that the hardware resources are uninterruptedly provided with adequate power and cooling during the reserved moment supreme. In HPC data centres it is quite common to differentiate between, on the one hand hardware that is essential for management of the system as a whole and for persistent storage of precious data, and on the other hand compute-only hardware and storage used as “scratch pad” for running jobs. The first category is provided with ample redundancy. For the second category, which is largest in power consumption, the redundancy provided by batteries is enough to bridge short power dips and to complete an automated graceful shutdown when the absence of the normal power feed passes a time threshold. Again: for facilities that are very much geared toward usage for research and development this is a cost efficient and energy efficient way to go about. If there

is a power outage, the users take their losses. Their running computations fail and will simply have to be rescheduled and rerun. Because power outages are exceptional, the business is much better served by investing in more capacity than in more redundancy for the exceptional case. But for urgent computing applications, this may not be enough.

2.2 Potential Users

Most European countries have meteorological offices that have dedicated compute resources at their disposal for doing short term weather forecasts. The practice of timely issuing "weather warnings" against high wind speeds, extreme rain, or snow, is well established. Prediction of floods in urban areas however is becoming a specialized area of research and development. Published use cases range from applications that run on a cloud virtual environment that has very modest hardware requirements [8], [9], to work that explores the added value of the integrating of high resolution simulation capabilities of supercomputers into a distributed research infrastructure for hydro-meteorology (DRIHM) [10] Other case studies present a solution that integrates reliable monitoring with on demand flood simulation triggered by the monitoring [11]. The computational capabilities required for the simulation models cannot always be deduced, as this is not necessarily the focus of publications.

UC applications also exist where the computational demands of the application itself are not significant but there exists a dependency on datasets which are large or (like weather forecasts) are regularly produced by HPC applications. For example, dispersion models for predicting the path and extent of airborne toxic material in the event of nuclear or chemical accidents require up to date weather forecast data as input but are often relatively low on computational requirements themselves. Also, archives of datasets such as weather forecasts and observational data can enable the reverse of this such as finding the source of outbreaks of airborne virus' such as foot and mouth disease in cattle [22].

UC, utilizing the computational capabilities of supercomputing, has also been suggested as a means to provide timely and clinically relevant assistance to medical experts performing complex surgery [12]. The added value of supercomputing resources becomes apparent when they are used to produce outcomes of complex simulations on a timescale that matches that of human engagement. In particular, using HPC resources it is now becoming possible to use individual whole genome sequencing (WGS) to guide personal strategies for disease diagnosis and therapeutic selection. There is increasing evidence that WGS can be useful in the acute care of infants with genetic diseases in neonatal and paediatric intensive care units [19][20]. Using advanced HPC technologies, clinical workflows can now reduce the time for WGS for emergency management of genetic diseases to 26 hours [21]. This process can thus be of real benefit in clinical environments and the relatively low demand rate but urgent nature of the computational requirements is an obvious use case for UC.

2.3 Current status

Existing publicly funded High Performance Computing (HPC) systems, along with cloud systems and grids, created for other purposes than UC, have all been proposed as UC platforms, because most domains of UC currently cannot afford dedicated systems [13]. The mission of PRACE is first and foremost to create and sustain a pan-European research infrastructure of state-of-the-art supercomputers, to cater to the needs of European academic research and development communities. So for piloting UC experiments in a PRACE context it would seem obvious and legitimate to focus on use cases that stem from European academic communities and have expressed an interest in using supercomputing resources.

Most authors proposing to use HPC resources at the same time also acknowledge that there are serious discrepancies between the requirements of a platform they would like to integrate and the ways in which HPC providers actually provide HPC today. The most noted discrepancy between what UC applications need and what HPC centres actually provide, is the fact that many HPC centres do not implement job pre-emption, nor advance reservation. The reason for this absence is simply that it does not fit the optimization of their "business model". The HPC facility has to cater for many academic clients with different affiliations. What they typically have to do, is maximize the job throughput of the machine as a whole in the most cost efficient and energy efficient way possible. There may be some modification of priority induced by the wish to give all users a fair share of the system's capacity, but there is usually only one quality of service for all.

But perhaps a number of UC applications can be accommodated with fairly little impact on the total throughput of the machine and without actually pre-empting jobs. Large HPC facilities usually implement more than one batch queue. Since there has to be support also for porting, developing, debugging, and so on, there is always a fair amount of short test work going on. The test activities would be frustrated if the waiting time between short test runs would be very long. At SURFsara, to accommodate a steady work flow for testers, backfill scheduling is applied to most resources but also a fairly small portion of the nodes is set aside to serve only the "short jobs queue". There are several ways to go about, but clearly tuning the parameters of such a queue and the number of nodes serving it, is a way in which the batch system can guarantee that N nodes can be made available within M minutes, without resorting to job pre-emption. Whether this is sufficient and feasible to accommodate an urgent application of course will depend also on the number of nodes needed by such an application as well.

The authors claiming that they can integrate supercomputers in their work will at best run state-of-the-art parallel simulation and analyses programs, not something that is more fault tolerant that other parallel programs that usually run on these machines. The overwhelming majority of all programs running on supercomputers are MPI or MPI and Open MP programs. While MPI can be used to run a parallel computation over several thousands of cores, the MPI framework is not at all fault tolerant. All the resources – the many cores, memory units, and communications channels - are all very tightly coupled in a running job. Operating systems and batch systems nowadays allow each task to be pinned on a particular core, for increased efficiency. However when an unrecoverable error occurs in a single core, in a memory unit, or in a communication channel, this usually implies that the complete job has failed and has to be rerun completely, or at least from the last saved checkpoint, if – user level – checkpointing is applied at all. Typically there is no redundancy in cores at all and no backup scheme for a failing core. Often, there is some redundancy in the interconnection between nodes. If a single channel breaks, theoretically this only leads to increased latency and loss of some bandwidth, so to loss of performance rather than failure. In practice however, most jobs break when a channel in the interconnect fails, if only by falling into a communications deadlock, as in MPI the responsibility of matching every receive with a corresponding send, lies with the application program, not the MPI library.

With the involvement of ever more tightly coupled components in a single system to serve a single computational job, the probability of breakdown of a single component, causing failure for the complete job, in principle increases. Yet in the last 10 to 15 years, many applications have been able to successfully scale up to utilize higher numbers of cores. To a large extent this is made possible by the increased quality of the hardware. Not only has the hardware industry been able to keep up with "Moore's Law" and increase the number of transistors on a single integrated circuit, but the quality and reliability of components generally also has been improved. Many applications have not shifted to more fault tolerant algorithms, but taken a

bit of a free ride on the hardware improvement. Still it does happen once in every so many jobs that a job fails because it was not prepared to handle a hardware exception. The recipe then is indeed to rerun the job. Because it happens only once per so many successfully completed jobs and the context is research, rather than a time critical step in a production process, or any form of urgent computing, this is not felt to be a problem. But it might be a problem, an unacceptable risk for urgent applications.

The requirements for the application and the "timescale of urgency" of any application that is accepted as a candidate to run on PRACE machines must be carefully scrutinized. The magnitude of a risk has a probability as well as an impact component. Most applications will need only relatively small number of nodes of a PRACE Tier-0 system. In some case the magnitude of risks could for example be held below the "unacceptability threshold" by greatly reducing the probability of hardware failure during the job. One way of doing that could be running jobs in duplicate on completely disjoint resource sets (nodes, switches) of a system.

Both in the medical application area as in the hydro-meteorological application area, urgent applications need to be supplied with lots of recently gathered data describing the problem at hand. And after simulation and analyses the final outcomes have to be available to the users. This is probably the case in any area of UC. It implies however that a reliable implementation has to secure availability of the input and output paths, to and from the supercomputer, and not just the availability of the supercomputer itself.

In HPC, most of the resources are made available for use only under several conditions such as scalability (ability to parallelize) to make the most effective use of the expensive hardware. Others, though, are secured by the policies the users need to be accustomed to before starting accessing the site's resources. One of the examples is based on firewall allowing the network traffic only from registered IPs. In clouds, though, the accessibility can be based on the users' requirements.

One of the main challenges in UC is the resource allocation step. In the current implementation of above mentioned SPRUCE [6], a human operator is designated as the final arbiter of job routing. However, this operator is provided with live data about the currently available capacity of candidate computing centres as well as statistical information such as processing speed, available programs and historical data about past UC assignments. This would provide more accurate insight about the real life performance.

It is safe to assume every partner computing centre in PRACE employs some form of internal monitoring mechanism. However, in order to assign a UC job to a centre, the status of systems might not be available in a readily exchangeable format. In any case, such a platform for exchanging information does not exist at the present. Proposing a standardized mechanism for gathering such exchangeable information would most likely serve to complicate internal policies and increase administrative burden. Nevertheless, it would be deemed reasonable to agree on a simple, least common denominator format for exchanging health and load status information of the computing, storage and networking among the partners in PRACE. As a side effect, this would also result in pressure to provide a higher standard of maintenance in computing centres.

Another important factor for determining the computing site selection would be data transfer and networking bandwidth. Due to the inherent unpredictability of the UC needs, it would be reasonable to assume that the data to be processed would not be present in the site of computation. One of the deficiencies in the current implementation of SPRUCE, as noted also by the designers, is the lack of focus on data transfer. For some type of job assignments, the allocated time may not be even enough for data transfer. Hence, minimizing data transfer time may play a major role for optimal UC efficiency. Some cases of natural disasters which

are among major triggers of UC requests, might compromise data transfer paths, such as fibre optic communication lines. Consequently, live monitoring of data connections and risk assessment should be a part of decision making process.

In the current implementation of SPRUCE, the privilege to initiate urgent computing requests is given to the scientists. This requires an optimal coordination among peers, preventing duplicated efforts and waste of limited resources. In addition, scientists do not have the authority to enforce policies based on their findings and are restricted to assisting government officials in making informed decisions. Likewise, determining the priorities of the subjects is beyond the responsibility of the scientists. Therefore, shifting the responsibility to initiate requests to the administrative policy makers would ensure proper coordination and lower communication overhead, which would be in contrast to the case in SPRUCE. SPRUCE, the first UC infrastructure, has an important deficiency regarding the submission of jobs. First, token holders initiate the UC signal via a portal and then the available resource is sought and finally the job can be submitted there. Such a long procedure might be accelerated if the job submission can be done directly on a portal.

Since PRACE HPC partners' centres have separate compute and data resources, it would be vital to have a common portal for UC needs. Because of urgent requests might be asked for UC needs at any time, the programs which will be employed for the computations must always be ready to run. Therefore, these programs must be updated and tested regularly to prevent any run failure. Since PRACE HPC centres are heterogeneous in the sense of installed hardware and software, we must ensure that these programs should be able to run on different architectures.

When a UC job is completed, the results must be analysed by the experts and the critical findings must be delivered to either local or administrative officials immediately for timely decisions such as evacuation. In order not to have any difficulties in transferring the results of UC, contact information of all officials must be presented with job submitters, scientists or HPC centre staff.

A shortcoming in SPRUCE, as mentioned in the literature, is the omission of work-flow formalism in UC requests [23]. As noted previously, one of the main concerns in UC that is not addressed in traditional HPC solutions is fault tolerance. One of the aims of work-flow based computation is dividing a job into small, relatively self-contained and manageable parts. When implemented properly, this would allow the user to process independent parts simultaneously, possibly on the different clusters if data transfer is feasible. This would also partially solve the fault tolerance problem, as only small parts of a job are restarted, as opposed to reprocessing of an entire monolithic job.

On the subject of work-flows, some authors in the literature [23] have claimed that allowing interactivity in work-flow processing would allow human decision makers to optimize the pathways, leading to earlier results and reduced computational costs, in addition to continuous flow of feedback from the computer about the job. Current support for interactive usage in HPC centres is discouraged due to perceived lower throughput; however, such usage might prove essential in UC.

2.4 Relevant policies

All current PRACE-RI Tier-0 and Tier-1 systems utilize a batch oriented access model. Batch systems (BS) are used to control jobs run and resource usage efficiency. They are primarily responsible for starting/stopping applications, implementing a scheduling policy which decides which job to run next and places limits on resource usage, and often implementing some form of charging mechanism. The scheduling policy in particular varies widely between

sites in terms of decisions surrounding the size and duration of jobs, the priority associated with particular groups or individuals, whether different QOS levels are available or whether jobs can be pre-empted. To be useful for UC applications, all sites would need to implement a common set of minimal policies which would guarantee these UC applications could be initiated in a similar manner and get access to the required resources within a defined period of time. Thus, the enablement of UC within the PRACE-RI is largely a matter of agreement on and implementation of common policies rather than a technological problem.

Two groups of use cases can be distinguished: regular and event-driven. The former one is assumed to have regular computations on dedicated resources. In turn, the latter, for which examples can be: weather forecast, earthquake early warning system, storm surge computation is based on solutions using specific algorithms or strategies improvement, resources setup based on decision support systems.

Frequently, UC requires early decision support and human interaction with applications which stands contradictory to systems running tasks in batch mode. Many batch systems provide the ability to run interactive batch jobs where a user is allocated a set of nodes and granted shell access to them. However, this is frequently disallowed on all but a small development partition for efficiency reasons. Again, this is a policy decision as to whether interactive access is allowed for large, production runs or for extended durations.

Scientists are provided with transferable Right-of-Way tokens with varying urgency levels.

During an emergency, a token has to be activated at the SPRUCE portal by the token holder. These tokens, functioning as only session initiators, are pre-generated during non-emergency times, with specific associated restrictions. Tokens are entrusted to scientists upon request, but are transferable under the discretion of the holder. Upon activation, the token holder can grant high privilege access to one or more scientists for computing resources, within the limits of the token. In SPRUCE, the actual target for job execution is determined by the advisor, based on criteria mentioned in the previous section. The authors of SPRUCE specifically emphasize that the usage of a token only mandates access to a high priority queue on the target computing centre for the token lifetime. As such, the submitted jobs will still be subject to local policies, which may include providing "next-to-run" status or immediately pre-empting other jobs.

Whether a middleware layer such a SPRUCE or a strictly policy based approach would work for PRACE remains a question for investigation. As a first step, a majority of sites should be assessed for whether it would be possible to implement the batch scheduler policies necessary to provide the service. Once this is established, the end-user protocols and access methods will need to be decided.

There is no European level policy defining the terms of the Urgent Computing provision functionality. In most cases, it relies on the cooperation with the corresponding institution branches. The internal policy, if it exists, concerns the exchange of cpu-cycles between distributed branches of one institution – located in distinct cities.

2.5 Description of the pilot

WP6 aims to demonstrate the pilot installation of the environment ready to use for the Urgent Computing by the end of the project duration. The aforementioned prerequisites will be adapted to the next stage of the development process: design document creation, implementation and preparing the pilot testbed on a selected machine.

The pilot will be installed on a Tier-1 machine belonging to a PRACE-RI member site. The prerequisites will contain the configuration of the LRMS towards accepting the UC

application. The main work will focus on defining the batch scheduler policy decisions which includes:

- The attributing of the job intended to run as UC required.
- The permissions to submit UC jobs and the maintenance of access and update.
- Test of the following UC job scheduling strategies:
  - Grant highest priority so that it will run at soonest available opportunity.
  - Pre-empt/suspend currently running jobs to free resources immediately
  - Kill currently running jobs to free resources immediately
- Development of a suitable procedure to signal that the underlying use case for the UC application has been addressed and that no further jobs will be submitted in the short term.
- Evaluate whether manual intervention is desired or required.
- Development of a UC job invoicing model.

Since WP6 has not got access to UC applications till now, the test application will be taken from PRACE benchmark suite. All PRACE benchmarks are a kind of stress test for compute environments. Hence, it should be able to realistically emulate a real UC application with these codes. In addition, Task 6.2 will assess the behaviour of the selected application on a system with randomised malfunctions of different sub-systems.

Along with increasing computing power UC applications should carefully select appropriate methods and algorithms to help improving speed and reliability. Examples of such methods are chaotic relaxation and meshless methods as well as sparse grid combination techniques. The use of them in parallel programs is for instance a major topic of a working group from the MPI Forum. Another group - so-called - “adaptive algorithms” might also be used to improve application fault tolerance where restarting of computations is impossible due to lack of time. These algorithms are based on the fact that when some of the computing cores fail the task is able to continue and the final result can be accepted. On the other hand, partial approximate results can be obtained very quickly which are continuously improved during application run time.

The last class of algorithms derives from genetics where they help to refine the input (recorded) data and provide the best selection settings for a given constraint.

A more “brute-force” approach to fault tolerance will be to simultaneously run the same UC application at two independent, remote sites. Thus, any hardware faults on one system or network would simply mean that the results could still be obtained from the concurrent application run. An added benefit of this approach would be the possibility for verification of independent results from multiple sites. There will be tested the MPI middleware functionality for managing MPI processes in the fault tolerant environment (with implementation of eg. CHARM++, or AMPI[http://charm.cs.uiuc.edu]).

Since SPRUCE hasn’t been updated in the last 5 years, this software will not be considered for deployment on the PRACE infrastructure. The work of WP6 will therefore mainly focus on the review of the functionality features of the SPRUCE software stack and the assessment of the software stack in a test environment.

The ready-to-use pilot will offer the functionality based on the well-defined policy of running Urgent Computing applications. It is described in the following table:

Description	Planned End	Participants
Design document ready	Month 13	PSNC
Deployment of the LRMS rules regarding UC application managing	Month 16	PSNC
Evaluation of UC code against fault tolerant behaviour on one (or several) selected Tier-1 machine(s) with LRMS tuning and possible fault tolerant intercommunication lib usage.	Month 24	PSNC
Deployment of pilot on selected Tier-0 hosts	Month 27	PSNC

Table 1: Planned functionality of ready-to-use pilot

WP6 agreed to make tests also on Tier-0 machines during powering-up or down phase, so it will not affect the standard run-time machine environment.

3 Service 2: The link with large-scale scientific instruments

3.1 Description of the service

The importance of computational methods and processing of Big Data has grown to the extent that they are sometimes called – next to experiment and theory – the pillars of science. On the one hand, constant technological advancement leads to increased precision of scientific instruments used in experimentation. On the other hand, the results of the experiments cannot be analysed in practice without the use of large HPC facilities and advanced software tools. There are a number of different technical challenges that need to be overcome in order for the modern science to fully benefit from the technological advancements:

1. Large-scale instruments produce enormous amounts of data, which needs to be stored and archived for future query and reference.
2. In order to be useful the data requires post-processing, analysis, and visualization, all of which may require large computational power.
3. For decades, the gap between bandwidth and computational performance has been growing. Consequently, in many cases transfer of the data from the instrument to the HPC facility proves to be more time consuming than the analysis itself.
4. Experiments are often augmented with numerical simulations, which themselves require substantial computational power and can generate a comparable amount of data. In this sense HPC facilities and efficient numerical software together can be viewed as an instrument of their own, and are of fundamental importance in the process of validation and refinement of physical models.

In this section we describe Service 2, developed within task 6.2 of PRACE WP6. The service aims to improve the link between large-scale instruments and the PRACE HPC infrastructure by addressing some of the named challenges. This is collaboration between four partners from CASTORC, NCSA, NIIF, and UiO. Each of the partners works closely with a scientific community that has different goals and utilizes a different instrument. This provides a unique opportunity to get a broad overview of the type of challenges the scientists face. Subsequent sections describe in detail the effort undertaken by each of the partners.

3.2 Relevant Policies

Being able to handle huge volumes of valuable data is a central point of linking large experimental facilities with HPC infrastructure. Whether the data is created through experiment or simulation, re-creating it may be expensive. Hence, in all cases policies

concerning reliability of data transfer and storage must be implemented. When it comes to long term or permanent data storage guarantees, most countries impose strict policies on data annotation and reproducibility. It is compulsory to augment the data with detailed information on how it was obtained, and how it can be re-created, if possible. Such policies also make sure that the data can be later re-used by a different group of users. Finally, when dealing with sensitive information it is necessary to consider data security and restricted access policies. This aspect is complicated by the fact that such policies differ from country to country. On the service design level it is important to ascertain that the technology used will be modern, has a large user base, and will be supported in the foreseeable future. Within this WP6.2 project we strive to align our effort with the EUDAT project. We are in active contact with developers of B2STAGE. Working towards PRACE-EUDAT interoperability is our common priority. Moreover, this is in line with the general European policy of encouraging collaboration between PRACE, EUDAT, and EGI.

3.3 Pilot 1: Linking CASTORC with the European Synchrotron Radiation Facility

3.3.1 Motivation

Synchrotrons are circular particle accelerators, which accelerate a variety of charged particles to produce X-rays. They consist of a doughnut shaped vacuum pipe and magnets, which are placed around the vacuum pipe to form a ring. The ESRF, the European Synchrotron, is the world's most intense X-ray source. It is located in Grenoble, France, and it is supported and shared by 21 countries. At the ESRF, high energy electrons are accelerated to produce X-rays that are 100 billion times brighter than the X-rays used in hospitals. The ability of ESRF as a "super-microscope" to reveal the structure of matter makes it a tool of tremendous importance in the hands of scientists exploring material and living matter in a very wide range of fields, such as chemistry, material physics, palaeontology, archaeology and cultural heritage, structural biology and medical applications, environmental sciences, information science and nanotechnologies. Therefore, the use of such instruments will drive the future work in a wide spectrum of sciences.

The vast amount of data (~TB per hour) produced by the synchrotron almost 24 hours a day needs to be stored, analysed, and archived for future reference. In addition, some applications like the tomographic reconstruction software, which is used to render 3D images of heterogeneous materials, could substantially benefit from a large computing infrastructure. Thus, due to the high computational needs for the analysis of the gathered data, as well as the large persistent data storage requirements, many projects using ESRF will benefit from the PRACE HPC infrastructure. Another challenge that should be addressed is the efficient transfer of the data from the instrument facility close to the computing resource.

At the moment, ESRF user communities perform most of the required tasks/workflows regarding the gathered data transfer and processing in a manual way. Additionally, external users of ESRF are generally responsible to ensure availability of computing resources (usually in their home institutions) so that they can process the gathered data from the instrument. Therefore, the integration of this kind of instruments with PRACE HPC facilities will benefit a large community of scientists and will advance science in a wide range of fields.

3.3.2 Potential users

Each year around 6000 scientist from Europe's leading universities and research centres travel to Grenoble to use the facility. ESRF also has many resident scientists that perform

experiments. We have established a communication channel with some experienced ESRF users, who are interested in taking advantage of the results of this project. This collaboration led to the identification of four main classes of projects, which would benefit from access to PRACE resources:

- Protein Crystallography is a form of very high resolution microscopy. It enables the scientists to visualize protein structures at the atomic level and enhances their understanding of protein function.
- Microscopy Diffraction/CDI (Coherent Diffraction Imaging) is a “lens-less” technique for 2D or 3D reconstruction of the image of nanoscale structures.
- Spectroscopy is the study of the interaction between matter and electromagnetic radiation. Through the study of the interaction of matter with light a lot of information about its structure can be revealed.
- Tomography refers to imaging by sections or sectioning, through the use of any kind of penetrating wave. Projection data gathered from multiple directions is fed into a tomographic reconstruction software algorithm.

The first three of the above mentioned use-cases will readily benefit from Tier-1 systems: their computational needs are moderate, but they have large data storage requirements. On the other hand, tomography requires a lot of computing resources as well as big data storage and could benefit from a link with Tier-0 systems. Hence, for the purpose of the pilot service we focus on tomography.

3.3.3 Current status

Provided that thousands of scientists travel to ESRF each year in order to perform their experiments using one of the 43 experimental stations, all beamlines operate 24 hours a day, six days a week. Precise and extensive preparation is carried out by users at their home institutes prior their visit to ESRF. The workflow, which is followed by ESRF users to perform their experiments, is described below:

- The prepared samples are exposed to the X-rays and raw data is produced by the ID19 instrument. ID19 is currently the most productive instrument at ESRF and it is used for tomography experiments.
- The resulting raw data (~10GB per dataset) is gathered at a central storage system.
- Raw data is processed using the pyHST software, while the experiment is running at a local cluster, which has access to the central storage system. Roughly 15GB of data per input dataset are produced.
- The resulting data, along with the raw data is stored for 50 days at a local storage in order to be staged out by the users.
- Users stage out both raw and processed data through a temporary account on a Linux based machine, using FTP, SFTP or SCP.
- Users perform additional processing on the raw data at their home institutions using pyHST.

Limited available computing resources and lack of adequate expertise in scientific computing at the home institutions, as well as the big amount of data produced by the instruments impose complications in the workflow. Some of them are the lack of properly administered machines, limited access to large HPC systems, and limited knowledge of using appropriate tools, such as resource management systems. ESRF, in order to partly serve local and external users, hosts a small cluster and some data storage resources on-site. Maintaining those compute and data resources is not for free and the maintenance adds an extra overhead to ESRF. Also, new

instruments coming in the near future will produce data at the rate of several TB per hour, making the preservation and the transfer of data even more difficult.

3.3.4 Future development

The centralized and extensive data production nature of the synchrotron facility, in combination with the lack of adequate computing resources at the premises of ESRF, imposes several difficulties and therefore several opportunities for improvements. This project aims to address those limitations by implementing a pilot link between ESRF and CaSToRC. In the context of WP6.2 we will perform the following tasks:

- Analyse the requirements and user needs. (M10)
- Design the architecture of the pilot based on the requirements. (M12)
- Obtain pyHST software from ESRF and ensure that is running on our system. (M14)
- Implement an automatic transfer of the raw data to CaSToRC, while it is produced by the instrument. (M18)
- Automatically perform the initial processing at CaSTorC as soon as the raw data is available. Ideally this should be done while the experiment is running. (M22)
- Store the raw data locally to give the ability to the users to perform additional processing on demand. (M22)
- Evaluate the pilot in collaboration with ESRF users. (M25)

Some limitations that we will have to address, based on the above tasks, will be the following:

- Data transfer must be completed in a reasonable time prior to the end of the experiment.
- Adequate computing resources should be available to perform the initial computation while the experiment is running.

3.4 Pilot 2: MIC-oriented Multithreading for HEP and Health Geant4 Computations (NCSA)

3.4.1 Motivation

This service is piloting the use of emerging large scale Intel Xeon Phi based HPC facilities for a variety of applications ranging from LHC physics to hadron therapy for cancer treatment. Initially it will focus on demonstrating the service for LHC experiments at CERN. The LHC – being the world's largest and most powerful particle collider, and the most complex experimental facility ever built – provides a unique opportunity to link to PRACE HPC facilities and thereby accelerating compute-intensive steps of the analysis and simulation pipelines.

An essential part of the data analysis in all particle-matter interaction considerations are Monte Carlo simulations. Physical events are generated by numerical software using a theoretical model, with a complete set of detector parameters as an input and a set of final-state particles as an output. Results of the numerical simulations are compared with the data gathered during experiments in order to validate and refine physical models. The simulation of the detector response – propagation of the generated particles through the detector – is an extremely time and memory consuming procedure. Simulation of one proton-proton collision, which provides exhaustive information about all particles positions during the whole event, takes several core-minutes to compute, with a precision in the micrometer range and generating in average 1.5 MB of data. The number of generated events per year is ~ 10^{10}, which corresponds to ~15-20 PB of data. For the LHC experiments, the typical size of data

generated during experiments is tens of Petabyte per year, doubled by the MC simulation data of the expected signal and backgrounds, needed for disentangling detector effects from real physical processes and for correct interpretation of the results [2].

The fast and precise simulation of particle interactions with matter is an important task for several scientific areas, ranging from fundamental science (high-energy physics experiments, astrophysics and astroparticle physics, nuclear investigations) through applied science (radiation protection, space science), to biology and medicine (DNA investigations, hadron therapy, medical physics and medical imaging) and education. The basic software in all these fields is Geant4 (GEometry ANd Tracking) – a platform for simulation of the passage of particles through matter (https://geant4.web.cern.ch/geant4/).

Most of Geant4 applications, which will be actively used in the next years, are very demanding in terms of computing power and generated data volumes, and the demand will surpass the potential of the available resources. The designed service will enable the execution of Geant4 based simulations on HPC architectures with large clusters of Intel Xeon Phi coprocessors. Thus, it will open this emerging HPC infrastructure to large-scale scientific instruments (like LHC and all other accelerator complexes) but also to hadron-therapy facilities and radiation protection centres that in a long term may be viewed as a step towards the solution to resource deficiency.

3.4.2 Potential users

The research process in HEP strongly depends on the quality of MC generated events and their further processing within the Geant4 toolkit. In hadron therapy, the same procedure takes place both for treatment planning and R&D works, with a body-phantom replacing the detector. Also in the field of PET, Geant4 based GATE software is the standard for simulation of the detectors for further R&D purposes. The number of potential users of the service exceeds 10 000, estimated based on the number of Geant4 users. Though the scales of the tasks in different use cases are different, this user community has steadily growing needs. Therefore, any possibility for diversification of the resources and of the task-assignment to particular resources will be highly appreciated. With the “multi-MIC” version to be developed, the new service will provide such a possibility by opening an entire new class of HPC resources to this large user community.

There is an expression of interest on part of CERN-IT management and the MC Group of CMS in such a novel service, allowing a diversification of the employed HPC resources and operational use of an advanced HPC architecture – clusters accelerated with Intel Xeon Phi. The MC Group will readily participate in the pilot service deployment.

3.4.3 Current status

Geant4 exploits the object-oriented technology to achieve transparency of the physics implementation, as well as openness to extension and evolution. It encompasses a wide set of tools for all domains of detector simulation, including geometry modelling, detector response, run and event management, tracking, visualization and user interface. The Geant4 source code, libraries and user documentation are freely available.

In the context of Geant4, the present computational approach relies on GRID technology, employing an enormous number of “standard” x86-64 processors. Introduction of different accelerators – GPUs or co-processors – boosted interest in hybrid architectures and initiated large-scale efforts for adapting specialized software.

The largest research facilities in HEP (such as ATLAS, CMS, ALICE and LHCb at CERN) use the Worldwide LHC Computing Grid (WLCG). The WLCG employs a tiered structure of computing centres according to their characteristics. The problems with this approach are two-fold:

- With the increasing data volumes, the resources become more and more insufficient, even with distributed infrastructures such as the WLCG.
- One of the largest computational communities worldwide does not use new architectures equipped with large number of Intel Xeon Phi coprocessors.

3.4.4 Future development

The Geant4 Collaboration (http://geant4.web.cern.ch/geant4/collaboration/) has developed a single-MIC version of the toolkit. First tests indicate a potential for significant gain in terms of time and allocated computing resources: both the throughput and the speedup scale very well with the number of threads [3]. There are still only a few HPC clusters with a large number of Intel Xeon Phi coprocessors, but their number is expected to increase rapidly. The specifics of the designed service require sufficient storage space and a fast connection between the clusters and the potential users. To provide an efficient service, a multi-MIC version of the toolkit is needed, as well as an appropriate sufficiently flexible user interface, adaptable to parameters of the existing and future facilities.

The technical realization of the pilot service allows for two modalities. The experimental data from LHC is processed on a Tier-1 system. The synthetic data can be created either on the same Tier-1 (if equipped with Intel Xeon Phi coprocessors), or on several smaller clusters, with the results being uploaded and published on the Tier-1. The long-term storage of the generated data is a responsibility of the users. The simulated data mimics the format of the experimental data. The analysis protocol depends on the particular research problem and requires both types of data. No significant service-related data transfer from the research instrument (in this case, LHC) to the coprocessor cluster (alone, or as part of Tier-1/Tier-0 centre) will be needed (only configuration files). Large data transfer will take place between the coprocessor cluster and the Tier 1 centre. The same applies for the Hadron therapy centres.

The detailed development and implementation plan for the service pilot is as follows:

- Analysis of the requirements for data transfer and intermediate storage (month 12);
- Preliminary work, bringing the tool to the necessary maturity level – completion of the “multi-MIC” version of Geant4 toolkit, its validation against the CPU code, and investigation of its performance and scalability. This is currently done in close cooperation with Geant4 Collaboration (month 18);
- A pilot deployment of the code on a suitable computing infrastructure – for example, Sofia Cluster Avitohol, with 300 CPU Intel Xeon E5-2650v2, 300 co-processors Intel Xeon Phi 7120P, 9.6 TB memory and designed peak performance of 420 TFLOPS (month 22);
- A Monte Carlo event generation service for the LHC experiments. These events have to be certified, thus only the authorized MC Group will be involved in the tests of the pilot service (month 25).

3.5 Pilot 3: NIIF support for ELI-ALPS project with HPC from the early phases

3.5.1 Motivation

The Extreme Light Infrastructure (http://www.eli-laser.eu/) is an ESFRI project to build world leading next generation laser research facilities in Czech Republic, Hungary and Romania. ELI-ALPS (Attosecond Light Pulse Source) (http://www.eli-alps.hu/) is part of this infrastructure located at Szeged, Hungary, aiming to analyse interaction between light and matter. This technology can be used in material sciences, biochemistry, medicine, nanotechnologies, nuclear physics, astrophysics and cosmology.

The ELI project is now in the phase of building the infrastructure and planning the details of how scientists can perform their research on the distributed infrastructure. ELI is also seeking partner e-Infrastructures, which can directly or indirectly help to solve computing, hosting and storage challenges faced by such a large-scale infrastructure.

The PRACE pilot aims to support ELI-ALPS with a dedicated network setup between the PRACE Tier-1 LEO system and research facilities, HPC and Data resources as well as with knowledge of special technologies in order to better utilize the HPC resources. The following requirements were identified based on preliminary discussions:

- Both national and international high-speed data connectivity to the facility, and all its sites, considering the possibility of online off-site analysis (20Gbps throughput)
- High amount of multi-tier offsite storage
- Large, accelerated HPC resources on an ideal architecture for the tools that fits best for the purpose
- HPC knowledge to solve specific scientific challenges to analyse, simulate and visualize data

3.5.2 Potential users

The primary mission of the ELI-ALPS Szeged research facility is to make a wide range of ultra-short light sources accessible to the international scientific community. Laser driven secondary sources emitting coherent extreme-ultraviolet (XUV) and X-ray radiation confined in attosecond pulses is a major research initiative of the infrastructure. The secondary purpose of the facility is to contribute to the necessary scientific and technological developments required for high peak intensity and high power lasers. Application areas include:

- *Chemistry, Biology and Nanoscience.* Attosecond pulses enable a set of time-resolved intra-atomic and intra-molecular electron dynamics experiments, which provide insight into the temporal evolution of important molecular excitations and chemical reactions.
- *Medical applications.* Diagnostics and therapy are the twin targets of the medical applications of brilliant X-rays. Coherent X-ray beams facilitate phase contrast shadowgraphy or 3D tomography, which yields high-resolution insight into tissue density structure or tumour tissue.
- *Energy research.* The system can be a tool for real-time imaging and investigation of chemical changes, reaction pathways and kinetics on the atomic and molecular level in a time-resolved manner for materials and processes of advanced solar cell and battery applications.
- *High-power photonics.* ELI-ALPS offers a development test-bed environment for upscaling high-power short-pulse laser systems for industrial partners.

3.5.3 Current status

ELI-ALPS is currently preparing its new laser facility, while gathering scientific knowledge and personnel to start working with the instruments and equipment as soon as it is procured and installed. The scientific staff already started several researches to analyse different setups and suitable environments of the soon-to-be installed instruments. This research is ongoing and requires a large amount of effort along with proper research infrastructure. Preparatory scientific efforts need proper HPC and storage resources to start simulation and early trials of particle analysis inspecting interaction with laser emission. These are carried out by using theoretical simulations, or by building smaller test setups within laboratory environment, and doing computation on desktop-level computers. This means different types of ELI related research might be able to utilize HPCs to enhance them.

The facility will be deployed in mid-2016. This is in line with the span of the WP6 project, which will aid implementation of data transfers and analysis. The use cases and possible setup of instruments are currently being analysed. It is crucial to have a high-speed connection between ELI participant institutions to enable successful collaboration to share data and results. The construction of the ALPS site and buildings is coming the phase when a contract is needed with a provider, which can connect the system to the Internet with a cutting edge technology to offer a high speed, low latency connection. The planning of the infrastructure has already been started, and is currently in a phase of identifying suitable environment and local appliances needed to be able to handle, analyse and visualize large amount of data produced by detectors used in different research setups.

The tools and techniques available to support such research are also under investigation. One of the particular research fields is to simulate the action and interaction of particles within a few hundreds of cubic micrometres, which requires the use of millions of grid cells. The analysed period of time is only one picosecond; the area needs to be sampled millions of times within this small period. Research includes Particle-In-Cell analysis in one or 2 dimensions at once. Threads running in parallel are between 60 and 100. The prototypes of this research are currently under preparation in a non-HPC environment.

3.5.4 Future development

ELI and related projects will provide ELI-ALPS with an internal infrastructure to satisfy basic requirements. However, to deeper analyse the data it is compulsory to perform advanced computations, simulations, and visualizations using larger external infrastructures. Initial discussions suggest that this project can benefit a lot from both local and international infrastructure collaborations.

The developed service will combine utilities to transfer the data from the instrument to the computational site, and set up a software environment that will help in analysing the data. As an initial phase of this pilot (first three months), network connection requirements need to be defined and fulfilled. When an instrument and connection equipment are ready, there is a need for a service to deliver the data to an offsite data centre located close to an HPC centre. Advanced tasks will be then be carried out on the data in the HPC facility and the results will be sent back to the research facility. This requires a high-speed network connecting all ELI facilities and a data centre where large amount of data can be stored, along with HPC capacities, with which data can be analysed with the help of most suitable software optimized for the task, by using accelerators, if applicable.

In the second phase (five month period), use cases with a need of transferring and processing high amounts of data will be analysed. Several laser-related research groups need tools that are currently being evaluated as pilots. For each particular data processing algorithm, an

exhaustive analysis of the most suitable HPC infrastructure (including accelerator support) is required.

As a third phase (five month period), a detailed analysis of available Tier-1 HPC systems will be performed to match numerical applications with the most suited system architectures. Early research suggests that there might be special algorithms (e.g. Particle in Cell) that would benefit from GPU acceleration. X-ray imaging tools which might also benefit from acceleration are currently also under investigation.

In the last phase (four month period) application optimization and scaling options will be evaluated.

3.6 Pilot 4: Linking Next Generation Sequencers with PRACE (UiO)

3.6.1 Motivation

Nucleic acid sequencing is used to determine the order of nucleotides in a given DNA or RNA molecule. Possibly the best known DNA sequencing endeavour was the Human Genome Project: it took 13 years and was completed in 2003. Since then the introduction of Next Generation Sequencing (NGS) made whole-genome sequencing affordable and practical. NGS is not limited to human DNA any longer: it promises benefits for a range of applied fields, e.g., diagnosis and establishing links between genetic variations and disease, and for epidemiology and pathogen studies, but also contributes to the development of theoretical fields, e.g., comparative biology.

It is inevitable that NGS will have a large impact on the modern society, both when it comes to sequencing of human, and non-human DNA. Over the past decade sequencing capacity has been increasing exponentially. However, in order for the produced data to be useful it needs to be processed, analysed, and stored. A modern sequencer produces 1-10TB of raw data per week. Practice shows that the deployed instruments are never idle. Initial post-run processing of the data obtained throughout one sequencing run takes around one day on a large SMP machine with ~50 CPU cores, and doubles the data output. Subsequent downstream analysis of the data requires a large HPC infrastructure and know-how that already today is beyond what labs can afford.

As of today, there is no standard way to integrate the sequencing work-flow with large-scale HPC resources. Much of the work done with the sequencing data is not automatized and performed internally by the labs. Pilot service 4 addresses this problem.

3.6.2 Potential users

It is hard to estimate the number of potential users around all of Europe. However, the demand for services related to sequencing grows every year – both for scientific and commercial purposes. Putting effort into development of standardized tools and routines will certainly benefit this field.

For the purpose of designing a pilot service the IT centre of the University of Oslo (UiO) collaborates closely with management and research groups from the Norwegian Sequencing Centre (NSC, http://www.sequencing.uio.no/) - a national technology core facility offering services related to the newest sequencers available. As of today, for data processing and analysis NSC uses the Tier-1 system available at UiO - the Abel cluster. Due to its wide availability this HPC resource can only be used to process non-sensitive data. The planned expansion of the operations will on one hand result in many more instruments being deployed, which will sharply increase the HPC needs. On the other hand, because of a

recently awarded funding, human DNA sequencing in Nordic countries will play an increasingly important role in the coming years. This will require the deployment of secure, sand-boxed HPC systems and procedures certified to move and process sensitive data.

Within task 6.2 of WP6 we will concentrate on handling of non-sensitive data only. Successful deployment of the new service will in future perspective benefit projects that process human DNA, where data transfer and analysis must be controlled and automatized in order to give security guarantees.

3.6.3 Current status

A crucial step of the NSC work-flow is the conversion of large amounts of raw data produced by the sequencers to a format suitable for downstream analysis. Converted files are delivered to the NSC end-users for further processing. As of today, operational staff at NSC uploads the data generated by the instrument manually into the Tier-1 Abel cluster and starts the conversion by editing and submitting job scripts to the SLURM resource management system. This is done independently for every sequencing run. Post-processing of the data produced by each run takes up to a day on modern shared memory compute nodes, and itself results in a large amount of data spread over multiple small files. The data then needs to be stored and made available for downstream analysis.

NSC is in the process of deploying a Laboratory Information Management System (LIMS) purchased from GenoLogics. A LIMS is an essential tool used by the labs to manage the work-flow. In the case of sequencers it is used to track all laboratory steps starting from preparation of the samples, through monitoring of the sequencing progress, to management of the sequencing results, data movement, and data processing. Any available LIMS provides data exchange mechanisms that allow the user to start and monitor data post-processing. However, linking to large-scale HPC resources is not standardized in any way and often implemented ad hoc. In the considered case the procedure has several limitations:

- The LIMS-initiated post-processing can only be started on a local resource, which is insufficient when it comes to compute capabilities.
- There is no support for resource management systems, such as SLURM, which is a must in multi-user, multi-instrument environment with a constant high sequencing throughput.
- For the largest part the current work flow requires user supervision and manual actions, which is not going to function in the future, when the facility is expanded.
- The downstream analysis of the sequencing results is performed manually by the scientists. This work involves creation and submission of SLURM scripts that execute programs like *bwa mem*, *bowtie*, *samtools*, and a few others. The details of each analysis (e.g., programs used and their parameters) depend on the project, but the number of used programs is limited.

3.6.4 Future development

The goal of this project is to design a service that would automatize the sequencing work-flow and provide a seamless link between the instrument and the 'backend' HPC resource. Achieving this goal requires that the above mentioned limitations are addressed in a standardized, automatic, and robust manner. The importance of this approach is recognized by the NGS community and a number of recent studies concentrate on addressing the described issues [4][5]. Within WP6.2 we will concentrate on the first step in the work flow, i.e., implementing a dedicated API to enable LIMS to start the post-processing on the Tier-1 HPC facility automatically, to monitor its progress and inform the user about any potential

problems, and finally to import the processing results into LIMS. To the extent possible we will consider data security and the goal of being able to deal with sensitive data in the future. Implementation of the following service components is necessary:

- as the raw data is being produced by the sequencer, perform a data transfer to the nearby HPC facility for processing
- as data is made available on the HPC resource, automatically generate and submit SLURM jobs, and monitor the job's progress
- notify about job completion, or about any irrecoverable errors at any stage of the work-flow
- archive/store the raw and processed data for subsequent downstream analysis on the HPC resource, or transfer it back to the lab

In order to routinely process the expected tens of sequencer runs per week it is important that the solution is robust, and that data transfer and processing mostly avoids user interaction. This is a challenge given that the work-flow involves a pipeline of interacting/dependent software components and runs in a multi-user HPC facility. There are multiple points of failure, which need to be considered:

- network connectivity and congestion
- HPC resource availability and occupation
- errors within the sequencer data, or within the (post)processing pipeline

In the implementation we will attempt to use existing and proven software components. As a large part of the service is related to data transfer and storage, a natural choice are services developed within the EUDAT project. For the initial service the B2STAGE seems to provide the necessary functionality. The pilot service will be deployed on the Tier-1 Abel cluster located at the University of Oslo. The initial development plan is as follows:

- detailed requirements in month 14
- draft architecture in month 17
- analysis of existing services in month 20
- pilot implementation in month 23
- evaluation of pilot and future recommendations in month 27

4 Service 3: Integrating tools and methods to simplify post-processing and visualization activities

4.1 Description of the service

As the data size produced by numerical simulation grows at a much faster rate than I/O and network bandwidth, data transfer represents a serious performance bottleneck in any simulation work-flows that include human feedback. For the implementation of the pilot service we will attempt to use existing and proven software components.

Many HPC work-flows consist of several loops of the following steps:

simulation -> bulk data transfer -> post-processing -> visualization.

The first step happens deeply inside computing nodes of an HPC infrastructure, typically an HPC cluster. The last step involves the user who is most of the times physically located outside HPC centre, possibly geographically far from it and eventually connected with low bandwidth: the straightforward idea of performing post-processing on the user facility is

becoming more and more inconvenient as the data size and computing capacity is increasing at a faster rate than the available bandwidth.

In order to overcome this situation, different strategies have been adopted, from data compression to artefact extraction. Nevertheless, even when large network bandwidth is available, the traditional work-flow that involves data movement is becoming less and less attractive.

A current trend is therefore to perform as much as possible of the work-flow within a HPC computing centre, providing infrastructure for remote visualization. This way, just input control and video streams travels on the (possibly low) bandwidth user connection, relieving also user devices from the burden of performing heavy graphics operations. The need for HPC centres to provide (remote) visualization services is widely acknowledged and it is foreseen to be subject of increasing demand by the users [24].

Another factor that helps in providing visualization services is the increasing availability of GPU's within HPC infrastructure, already used as computing accelerators. This allows running large scale post-processing parallel visualization on the same HPC resources used for simulation, or even advanced coupling between simulation and visualization (in-situ visualization techniques) that completely avoid the data transfer step by performing post-processing and visualization directly within a simulation run [25][26]. This is confirmed by the fact that many HPC centres are proposing visualization services that comprise both a session based remote visualization service (VNC access) on GPU accelerated nodes as well as more advanced parallel rendering services based on several visualization tools. Nevertheless, visualization services do not seem to have fully entered into the common user work-flow, possibly because the procedure needed to use these services are still a bit intimidating for the average user (SSH tunnelling, VNC client installation, graphics resource reservation, parallel rendering setup, simulation code instrumentation).

Also, even if visualization services of HPC centres are often layered on similar infrastructure, use the same open source software components and provide similar functionalities, see Table 2, there does not seem to exist any publicly available collection of detailed installation and deployment recipes, not to mention automation deployment scripts. It seems that each centre has to rely just on build instructions provided by upstream component developers that have to be general and require work to be adapted and configured to the specific deployment environment. To our knowledge there is no repository collecting specific recipes for deployment of remote visualization services components in an HPC environment. In addition current visualization services often do not address latest visualization approaches such as in-situ and web based visualization.

In order to address these challenges and outline possible services we propose these activities:

- **Improve visualization services currently deployed by PRACE partners**, promoting usage of application neutral remote visualization services (VNC based) as well as widespread high-performance visualization tools in order to simplify deployment and to improve user experience.
- **Select specific pilot projects representative of data intensive use cases** applying advanced techniques such as in-situ visualization or Big Data processing.

The final goal would be to publish a mix of deployment recipes and example pilot usage documentation that will

- streamline the deployment of visualization service components within HPC centres,
- simplify the access to visualization technologies,

- provide real use case examples for promoting usage and best practice of deployed tools,
- document pilot usage of state-of-the-art emerging visualization techniques.

4.2 Improving session base VNC Remote Visualization

4.2.1 Description of the pilot

Goal of this pilot activity is to improve the overall user experience of the visualization services as well as reduce the effort required to deploy them on infrastructures provided by PRACE partners. The pilot activity will be grouped as follows:

- Survey activity aimed at the production of a state-of-the-art document that would be proposed for publication through PRACE channels. The document will
  - collect information about state-of-the-art remote visualization technologies specifically oriented to HPC centres requirements,
  - collect information about characteristic of visualization services deployed by PRACE partners, gathering information on technologies adopted, access policies, current usage statistics and perceived obstacles to greater usage,
  - evaluate and compare available services and technologies according to user requirements and deployment policies, possibly defining some benchmarks and quantitative performance evaluation.
- Implementation activity: some of the tools and components used by CINECA to simplify access to visualization services will be improved, evaluated by other partners and possibly deployed in another visualization service. This will be the base for the service pilot demonstrator.
- Deployment support: deployment recipes of the aforementioned components as well as of common visualization tools (Paraview [27], Visit [28], VMD [29], Blender ...) will be collected documented and possibly formalized using a state-of-the-art deployment tool. This material will be kept in a code re-visioning system, possibly using the PRACE repository developed in service 4.

4.2.2 Motivation

A preliminary survey regarding visualization services provided by HPC centres in Europe and USA suggest that the infrastructures used for the services have the tendency to cluster around few hardware and software architectures, some based on Open Source components.

The similarity of hardware infrastructure, usage models as well as software stack used suggest that there is a potential for sharing experience among the centres deploying substantially similar visualization services. Nevertheless, even if deployment environment exhibits a substantial grade of similarity, it's quite hard to find shared deployment recipes or publicly available documentation about components and installation procedures.

Looking at user documentation of several services access procedures, it seems that the users need to perform a substantial amount of manual steps to reach the very basic goal of having a GUI enabled desktop session running remotely on the HPC visualization infrastructure. This could be somehow intimidating, especially for the novice user.

4.2.3 Relevant policies

Different deployment policies could affect visualization services:

- **Deployment architecture**: visualization services could be supported by different resource layouts such as:
  - A dedicated graphics cluster, consisting of a completely separated number of visualization nodes equipped with GPUs sharing a connection network, storage resources and autonomous administration.
  - A dedicated partition of nodes with GPUs within a larger cluster, sharing storage, network and administration with the larger cluster; node access could be direct from front-end or login nodes or through a batch scheduler, either using standard queues or reservations.
  - In cases where many nodes of the cluster are equipped with GPUs for computation acceleration, visualization services could be treated as standard computational jobs and scheduled accordingly. In this case, care must be taken to implement a system environment which is also suitable for visualization work-flows.
- Different policies regarding **resource allocation** could affect visualization services that are, for their nature, quite far from the batch computational workload typical of HPC environments: resource allocation could range from:
  - Shared access with simple fixed resources limitation, similar to access policies of login nodes, limits are typically defined as
    - CPU Minutes per process/per session and
    - Memory GBytes per process/per session.
  - Full nodes allocation with reserved access, this could be handled by schedulers using advanced reservations.
  - Interactive visualization jobs scheduler (PBS, LL,) specifying accounting, standard resources (core number, Memory, GPU) as well as special ones such as X11 servers availability. This type of allocation is more flexible but a bit tricky:
    - Typical visualization work-flows often involve interactive usage of a series of processes. Care must therefore be taken to insure that accounting and limitation applies to the whole process tree of the visualization session. This is ensured for example by applying Linux CGROUPS policy enforcement (for PBS scheduler the latest release is required to support this feature).
    - Relying on standard scheduled jobs raise also the problem of proper handling of (hopefully minimal) delayed interactive session start due to the underlying queuing system.
- Other relevant policies regarding **user access** involve security related restrictions such as:
  - Visualization nodes general accessibility and firewall restriction (open accessible IP, white list protected IP, private IP, accessible through login node SSH tunnel, multiple fire-walling)
  - User authentication and authorization methods and interfaces (usual user/password login, certificates, web based job submission portal, others).

4.2.4 Potential users

Results of this pilot could be useful for different actors involved in either deploying or using visualization services provided by any (PRACE) HPC centre:

- The **survey** describing the different technologies adopted by PRACE members to deploy visualization services could be relevant in the scouting phase of a new service.

- The **visualization service prototype** could be evaluated for adoption as "out of the box" simple solution for visualization service deployment, simplifying access, especially for novice users.
- The **shared repository** of single components deployment recipes could be useful for speeding up the deployment process, avoiding common deployment pitfalls.

4.2.5 Current status and planned actions

Survey and evaluation activities

The activity to produce a state-of-the-art document has already started: a list of relevant technologies to follow has been compiled. The document will be kept up-to-date by an active technology watch. Two HPC relevant technologies will be closely tracked: one proprietary solution (NICE-DCV [30]) and VirtualGL [32]/TurboVNC [31] as an Open Source solution.

From a preliminary survey activity regarding remote visualization services deployed by PRACE partners it seems that these two technologies currently cover almost entirely the remote visualization components used.

The following table summarizes the current situation of deployed visualization services.

Site	Type	Nodes	Mem	Cores	OS	Rviz sw	Clients	Scheduler	Limits	Net access	User access	Reference Information
CINECA Pico	Login nodes with GPU	2 with 2 Nvidia K40	128	20	RHEL 6.5	RCM/TurboVNC/ VirtualGL	RCM, VNC	SSH access	No	Open, SSH tunnel on node	SSH User, Pass	[33]
CINECA Galileo	Dedicated portion of cluster	3 with 2 Nvidia K40	128	16	CentOS Linux 7	RCM/TurboVNC/ VirtualGL	RCM, VNC	PBS Pro	PBS CGROUPS	Open, SSH tunnel through login	SSH User, Pass	[33]
IT4I Anselm cluster	Dedicated portion of cluster	2 with Nvidia Quadro 4000	64	16	bullx OS 6.3	TurboVNC/ VirtualGL	VNC	PBS Pro		Open, SSH tunnel through login node	SSH keys	[34]
IT4I Salomon cluster	Dedicated portion of cluster	2 with Nvidia Quadro K5000	514	28	RHEL 6.6	NICE DCV	VNC	PBS Pro		Open, SSH tunnel through login node	SSH keys	[34]
LRZ SuperMUC	Dedicated portion of cluster	7 with Nvidia K20	128	16	SLES 11 SP3	TurboVNC/ VirtualGL	VNC noVNC oneClick	Slurm		restriction on client IP	GSISSH , tunneled ssh	[35] [36] [39]
HLRS Cray XC40 Hazel Hen	Dedicated front end nodes	14 with Nvidia quadro 6000	128GB-1.5TB	24	SLES 11	X0vnc TurboVNC/ VirtualGL	VNC	TORQUE/ MOAB	CGROUPS	restriction on client IP	UNICORE , SSH User,Pass X509	[37]

Table 2: PRACE Remote Visualization services

Further data will be collected to evaluate and compare available services and technologies according to user requirements and deployment policies, possibly defining some benchmarks and quantitative performance evaluation.

Pilot demonstrator

Since two years CINECA has developed and deployed in production on different clusters a simple tool, called Remote Connection Manager (RCM) [41] that streamlines access to VNC based remote visualization services that require session job submission and SSH tunnelling. The main goal of this tool is to simplify the access to CINECA remote visualization services for scientific users.

RCM has been used for accessing remote connections based on VNC server sessions on all the GPU accelerated visualization nodes of CINECA's Tier-1 clusters (Galileo, Pico, Eurora ...) and even on Tier-0 (Fermi) login nodes. The Remote Connection Manager is a multi-platform (Linux, Windows, OSX) client/server application, wrapping SSH and TurboVNC clients into a Pyinstaller [40] Python executable that takes care of VNC connection bookkeeping.

Clients can be packed in the form of a single executable that unpacks all the needed components for the underlying VNC client, relieving users from the need to install them. The RCM tool also takes care of automatically informing the user when a new version is available. The tool currently relies on a server part to be installed (user space) on the visualization server. This takes care of handling vnc-server sessions, either in the form of SSH processes or of batch jobs (it currently supports PBS and LoadLeveler as batch schedulers).

This tool will be the starting point for the pilot remote visualization service demonstrator. Being completely based on open source technologies (TurboVNC/VirtualGL) and being released open source itself, evaluation and eventual deployment on other site infrastructures will be facilitated. The preliminary activity of the pilot that is currently ongoing consists in other partners evaluating the client part accessing visualization services provided by CINECA. The second phase of the pilot will be the experimental deployment on some other partner's infrastructure to effectively evaluate if the tools fit the deployment constraints and access policies.

The first candidate for external deployment will be IT4I Anselm cluster. In order to ease the port and evaluation of the software, preliminary development steps are required and already started:

- Streamline deployment
  - Provide deployment documentation
  - Remove site-specific code from code-base
  - provide scripts for automatic installation of required components
- User interface improvements
  - Allow user customization of session submission profiles and deployment parameters such as start-up menu
  - Improve error reporting

Up to now the development happened on CINECA HPC-Forge Subversion repository [42]. A migration to GitHub is planned to ease the client build refactoring and to promote collaboration. When the PRACE Open Source repositories services provided in 6.2.4 will be available, the pilot sources will be probably migrated to this repository.

Deployment recipes repository

Visualization services are composed by a series of components that are made available to users by environment modules. For example, CINECA visualization service consists of:

- RCM stack components : RCM, VirtualGL, TurboVNC, including Turbojpeg and other runtime as well as compile time dependencies
- Lightweight desktop such as Fluxbox
- Open source, common visualization tools: ParaView, Visit, VMD, Blender, Vaa3d etc.

Most of these components are installed using deployment recipes based on in-house deployment tools. This almost prevents reuse of the recipes in other centres. Some PRACE partners such as IT4I are already using the EasyBuild [43] open source tool to automate computing cluster software installation and environment modules deployment. We plan to evaluate porting visualization service components deployment "recipes" from the current in-house tool to EasyBuild.

4.2.6 Future development of the pilot

Regarding the evolution of Remote Visualization Services, one essential aspect is the activity of technology watch, which is essential to early catch useful innovations stemming from commercial vendors as well as open source projects.

The current technology watch information will be used to evaluate new promising techniques in order to improve performance and usability of the service:

- Extending the range of devices Remote Visualization can support:
  - Mobile devices;
  - Web browsers VNC clients like noVNC [39] and Guacamole [44].
- Virtualization of desktop: the topics of desktop consolidation and GPU sharing are attracting vendor effort (Nvidia grid [45])
  - Network bandwidth optimization : H.264 video stream compression has been adopted in Remote visualization (NICE DCV NVidia NVENC [46])

4.2.7 Action summary

Description	Start	Planned End	Participants
Technology watch on Remote Visualization systems	Ongoing	Month 27	CINECA, UL
Characterization and evaluation of current and future visualization services in PRACE community	Ongoing	Month 27	CINECA, UL, HLRS, IT4I-VSB
Remote Connection Manager improvements	Ongoing	Month 18	CINECA
Remote Connection Manager evaluation and deployment outside CINECA	Ongoing	Month 20	CINECA, UL, IT4I-VSB
Repository of deployment recipes for visualization service components	Month 12	Month 24	CINECA, IT4I-VSB

Table 3: Planned implementation actions for remote visualization pilot

4.3 Pilot projects for parallel and in-situ visualization and other advanced post-processing

4.3.1 Description of the pilot

Activity pertaining to this pilot will explore techniques and tools to address visualization and post processing problems that do not fit well within a single-node, single-user, VNC connection that are commonly supported by off-the shelf visualization tools usually deployed in Remote Visualization services. Based on partner interest and use base, we have selected the following activities:

- VMD-NAMD coupling for in-situ visualization
- Visit as scalable parallel visualization tool
- **ParaView** ecosystem: a parallel and in-situ visualization tool
- **Large Data** post processing workflows

For each tool or technique a real use case will be selected and a corresponding solution will be implemented and evaluated. Implementation recipes will be recorded and shared into a pilot repository, similar to the one described previously in chapter 4.2.5. Evaluation results will be collected into the final pilot report.

4.3.2 Motivation

Limitations of the VNC, single node session approach are mainly due to scaling issues depending on either cores or, more frequently, memory request being too big for the single visualization node:

- The single node available for Remote Visualization is not able to handle some steps of the visualization pipeline, either because of memory limitation (the data to be visualized does not fit within the single node RAM) or because the computational load is too high for interactive feedback on a single node; this problem needs to be addressed by using parallel visualization techniques.
- The traditional workflow that involves running a numerical simulation saving results on disk and later reading results using post processing tools to analyse and visualize data. This is not satisfactory either because data results are so large that the input/output step is dominating the runtime of the workflow or because early human feedback is needed to steer the simulation; in all such cases the use of in-situ visualization techniques seems to be appropriate.
- Data involved in the analysis process is so large that interactive visualization is not suitable, so that the visualization has to be pre-computed in a batch process step, possibly involving BigData analysis techniques.

4.3.3 Relevant policies

As this pilot deals mainly with scaling up visualization applications, most of the issues already exposed in the previous visualization pilot, also apply here. All proposed experiments have potentially large resource requirements, so their requests should be scheduled as other computational batch jobs.

In both parallel visualization as well as in-situ usage models, the user should interact in real time with either the parallel server or the simulation application respectively; possible resource allocation policies as well as internal networking could be relevant. In the parallel

visualization case, it could be useful to wrap the parallel server job submission in a GUI helper client application that notifies the user on the availability of the resources.

For in-situ visualization, it would be really beneficial, when the underlying transport software that connects the simulation with the visualization client would support efficient attach and detach. The availability of this feature would ensure that the user client could attach to the simulation, when application steering is needed, while the simulation runs unattended without major performance penalties during the rest of the time. In large data analysis experiments, I/O operations are dominant. So it is likely, that storage allocation policies will play a peculiar role, possibly requiring ad-hoc re-configuration.

4.3.4 Potential users

Potential users of this pilot are be:

- Users accessing parallel visualization tools such as ParaView and Visit, leveraging HPC infrastructure to speed up their post-processing pipelines.
- Researchers willing to use in-situ visualization “out of the box” (like in the VMD-NAMD case) in order to be able to interactively check and possibly control their running simulation remotely.
- Researcher/developers willing to dig into presented simulation code instrumentation with the goal of adding similar functions to their own simulation codes.
- HPC researchers willing to address infrastructural implications on large data post processing workflows.

4.3.5 Current status, planned actions and possible future development

VMD-NAMD pilot

Current status

In the field of classical molecular dynamics, one quite popular tool for molecular data visualization is VMD. In the same field, the same institution (University Urbana Champaign) provides tools for numerical simulation such as NAMD.

These two solutions can already interoperate to support interactive molecular dynamics in-situ visualization and steering as explained above. CINECA has started the evaluation of coupling NAMD jobs running on compute nodes with VMD running on a Remote visualization node.

We have verified the ability to attach and detach the VMD graphical user interface to the still running NAMD job. We have also verified the ability to interact with the simulation by interactively adding user defined forces for steering the molecular dynamics. Preliminary evaluation also does not show relevant impact on the performance of the simulation job.

These functionalities seem to be provided "out-of-the box" by the binary installation of both tools.

Planned actions

An example with a small publicly available data set will be deployed to help user adoption. Real use cases will be considered to evaluate usability and performance and to gather user feedback.

VISIT pilot

As indicated by preliminary surveys, Visit will likely be one of the key tools that should be part of any visualization service.

Current status

- CINECA binary installation on Galileo and Pico cluster, supporting GPU accelerated remote rendering (deployment script available)
- IT4I-VSB Salomon cluster installation, supporting parallel rendering acceleration in SW by Intel PHI boards (ongoing scripted deployment)

Planned actions

- Real case study, performance evaluation and comparison:
  - CINECA multinode GPU accelerated as well as non-accelerated parallel rendering,
  - IT4I-VSB largely parallel performance evaluation on PHI boards
- Detailed description of deployment (build, configuration, modules) and documentation

Future development

- In-Situ, either through the native LibSIm [49] or Visit + Damaris [47] evaluation, specifically
  - Deployment and test of out-of-the-box examples
  - Instrumentation of a real (simple) user pilot code for in-situ visualization on one (Tier-1) cluster
  - feasibility test for porting on one Tier-0 machine

ParaView pilot

ParaView is one of the most used scientific visualization tools and is already provided by most of the visualization services provided by PRACE partners.

Current status

ParaView is currently deployed in the production environment of CINECAs Tier-1 visualization services (Galileo and Pico clusters) as single node binary, standalone version that cannot use multi node MPI based parallel execution. An MPI enabled version of ParaView enabling multimode parallel hardware rendering on NVIDIA GPUs with the recent OpenGL 2 rendering driver has already been tested.

Planned actions

- Document all the required OS setting, installation and deployment recipes required for the different installation testbeds
- Deployment on non GPU nodes relying on a OpenGL software only implementation (suggested OSmesa Gallium)

Future development

- Set-up of an In-Situ experiment with Paraview Catalyst API [48] and a representative user simulation application.
- Software rendering supported by Intel Xeon PHI boards
- Consistent performance evaluation and comparison of different setups with different data sets with different I/O formats to evaluate effective scalability. The tests will be collected and shared as Python scripting macros.

Large Data pilot

With the evolution of HPC systems, several scientific groups doing large scale simulations are running more and more into Big Data Analytics issues. Those issues are currently mainly of the Big Volume type. Variety and Velocity of data are not playing a dominant role here.

Typical problems are for example the search of climate data for specific patterns which are indicators for specific events like super storm etc. In Computational Fluid Dynamic (CFD), one example is the detailed analysis of turbulence phenomena which is done by different methods, for example statistical ones. Those analyses require multiple walks through the data sets which have a size in the order of 70 to 400 TB. Obviously, technologies and methods for high data throughput and intelligent approaches for such kind of analysis steps are very important.

Current status

Some of HLRS power users in the engineering field have already started facing this kind of problems and are experiencing serious performance limitation in their currently deployed workflows.

The Large Data Pilot conducted on the HLRS infrastructure will try to optimize this real user problem that can be important for the whole computational engineering community. Specifically it will try to improve performance and methodology for the post processing of a fan calculation. Goal of the fan simulation is a detailed understanding of the occurring turbulence phenomena in combination with aero acoustic issues.

The high resolution calculation jobs for this fan problem runs tens of hours on a large HPC configuration using ten thousands of processor cores and producing an output of about 80 TB. The result data are then further analysed. For this analysis, several statistical methods are used first. Depending on the outcome of the statistical analysis several further examinations are performed, for example a search for peaks in the frequency spectrum. Each of those steps requires a walk through the whole data set. Thus, the result data is read about 10 times within a relatively short time frame.

Planned actions

HLRS plans to perform the following steps:

- Analysis of the current analysis method
- Evaluation of different system software optimization approaches to improve I/O performance
- Evaluation of new approaches to improve reading cycles
- Eventually, proposal for a specific hardware platform for those post-processing cases
- Potential users

4.3.6 Action summary

Description	Start	Planned End	Participants
VMD and NAMD deployment and evaluation for in-situ visualization of classical molecular dynamics	Ongoing	Month 18	CINECA
ParaView parallel deployment and evaluation	Ongoing	Month 18	CINECA, UL
ParaView in-situ visualization pilot	Month 15	Month 24	CINECA, UL
Visit parallel deployment and evaluation on Xeon PHI cluster	Ongoing	Month 18	IT4I-VSB, CINECA
Large data pilot	Month 12	Month 24	HLRS

Table 4: WP6.2 Service 3: Planned implementation actions

5 Service 4: Provision of repositories for European open source scientific libraries and applications

5.1 Description of the service

Service 4 of this subtask 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 software repository tool 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. The core has been decided to be based on GitLab [60] given the analysis of current technologies and possibilities referenced in this document.

GitLab is a well-known software which is a web-based Git [64] tool used worldwide by thousands of developers. Its nice interface provides a high user satisfaction and increases the productivity of the projects. It is also provided with a wiki and issue tracking features.

Modularity is another feature of GitLab. Multiple plugins can be connected to provide and extend functionality. For example, authentication with different providers and systems like Google, LDAP, CAS, etc. is possible with minimal effort. Further features of this tool are a customizable interface and auto-test tools for compilation.

GitLab has been compared with other services like GitHub or Subversion in terms of features and costs. GitLab has been the best option despite other alternatives were also very good. Further details are described in section 5.2.4 of this document.

A brief state-of-the-art analysis has been performed and some HPC sites have been asked to report their experience with different repository tools. Finally, during the first work phase of this task many partners including CoEs, PRACE projects or internal PRACE work packages have been queried to get their first impressions and requirements for the proposed service.

5.2 Pilot – GitLab

5.2.1 Description of the pilot

The pilot will consist in the deployment of a GitLab solution and the configuration of a set of accounts for symbolic organizations and repositories. Moreover a wiki for documentation and an issue tracker tool will also be configured to satisfy technical project needs. An auto-test tool, federation services, and project management tools will be also evaluated to complete the solution.

The purpose of this pilot is to get an overview on how the solution will work with a real structure and with different accounts for different kinds of actors. An evaluation of the prototype and final conclusions must be done to see if it is feasible to deploy such an infrastructure. We will also see the weaknesses and strengths of the solution that could be used to improve the service in a possible future production implementation.

5.2.2 Relevant policies

While implementing the prototype we must take into account some relevant policies regarding data ownership and access to that data. Each individual repository should be allocated to a natural or legal person who is legally responsible for the data contained there and should have a mechanism for creating project specific access policies. That is, given a project, next questions must be answered:

- Who is the party responsible for the project?
- Who can access the data of the project?
- What are the permissions of each user?
- Who/how to manage user accounts?
- Who is given an account and under which legal terms?

Also, policies for the data types that are allowed to be uploaded must be defined. Basic security flags will be analysed and documented to create a secure and reliable service. All this questions and matters will be precisely defined during the implementation phase of the prototype.

5.2.3 Potential users

Centres of Excellence

Potential actors that could use the service specified in the prototype, are the Centres of Excellence around Europe. We've contacted task 6.3 leader (Link with other e-infrastructure and Centre of Excellence) that provided contacts for the following CoEs:

- EoCoE Energy oriented CoE
- BioExcel Life Science CoE
- NOMAD Novel Materials Development
- MAX materials design at the exascale
- EsiWace - Earth System Modelling for Weather and Climate
- ECAM e-infrastructure for simulation and modelling
- POP Performance Optimization and Productivity (www.pop-coe.eu)
- CoeGSS Global Systems Science

For each one we've started to ask for feedback on what kind of services they would expect for a solution like the herein proposed. At the time of this deliverable, we have received feedback only from POP and EoCoE:

- POP is not interested in a code repository since the nature of its work does not require such a service, but maybe it could be useful for storing some program traces.
- EoCoE demonstrated a big interest in a GitLab solution. They are planning to deploy an account to GitLab to host private codes and extract public parts in order to make them widely available. Their interest is focused in two main aspects. First, they want a quick and easy account creation and second, they are interested in the possibility to host private projects. If both requirements can be fulfilled, they would be interested in using this service in the long term.

Research institutions/universities

The following institutions have been contacted in order to understand what current software repository technologies they are using today. The results show that there's equality on usage

of Subversion and git, being GitLab used on more sites than GitHub. As code tracking tool, TRAC is a widely used technology.

Institution	Wiki	Code
Tulane University	TRAC	TRAC
University of California	TRAC	TRAC/Subversion
Princeton University	/	GitHub/Git
Shanghai Jiao Tong University	Mediawiki	GitLab/git
CNR-ISTI HPC Lab	GitLab Wiki	GitLab/git
INRIA	Fusion Forge	SVN
CINECA	TRAC	SVN

Table 5: WP6.2 Service 4 - Research Projects/Institutions platforms for Wiki and Code

Research Projects

None of the 3 research projects that we contacted (PI-s of NIIF: ProPep, APGTPBL, HyDiG) uses any code repository solution. Anyway, all of them demonstrated their interest on adopting such a solution since it could make their work flow easier and also enhance their productivity.

Existing and similar services

A web search for similar projects discovered a very interesting initiative named is HPC-Forge [51] . HPC-Forge aims to provide HPC services for collaborative support for HPC users. It's based on Fusion Forge [50], started at CINECA and was later adopted by CSCS. Our analysis concluded that this is a very similar project on what we want to achieve.

HPC-Forge is a software development infrastructure; a collection of services to support the software development process:

- Source control management: A system that provides a central place where the team members can store and access their entire source code base.
- Requirements management: A system used for recording and tracking product feature requests.
- Bug-tracking: A system used to record and track errors and feature requests.
- Automated build: A system that builds the application every night by automatically executing the required build procedure steps at the scheduled time, without any human intervention. Automated testing: The tools that team developers and testers use to verify software and to detect and prevent software problems, such as functionality errors, reliability problems, performance problems, or security vulnerabilities.
- Regression testing: Any tool or combination of tools that can automatically run all of your existing tests on your entire code base on a regular basis (preferably at night, as part of the automated builds). Its purpose is to help you identify when code modifications cause previously working functionality to regress, or fail. For example, the regression system may be a script that runs one or more automated testing tools in batch mode.
- Data repository: A storage area (upload/download) to provide access to 'publish' releases, documentation, test data.

HPC-Forge chosen services

- Source control management: Subversion (http://subversion.tigris.org)
- Requirements management, Bug-tracking: Trac (http://trac.edgewall.org)
- WebDAV as data repository software

- A web-based interface (portal) to access and create service instances, manage users and their permissions.
- Hudson (http://hudson-ci.org/): Open source “continuous integration” (CI) server. A CI server can do various tasks like:
  - Check-out source code
  - Build and test the project
  - Publish the results
  - Communicate the results to team members

Apart from source control software, bug tracking and automated build and testing, HPC-Forge also provides a data repository tool based on WebDav. We are not going to consider such a tool in our repository because its functionality is not one of our prototype goals.

Another good thing of this project is that PRACE WP8 from PRACE-1IP and -2IP used HPC-Forge and hosted up to 20 projects on the site, but technologies became obsolete and this software stack was discontinued, see [52].

PRACE 4-IP Inter-collaboration

PRACE 4-IP WP4 and WP7 work groups demonstrated its interest on deploying Codevault over GitLab. A first prototype deployed on GitLab.com has been tested by SURFsara in order to evaluate the benefits of such a tool.

Currently different ways of structuring and organising repositories are being discussed within different work groups in order to find the best approach. One proposed scheme is to have a defined naming scheme for repositories, e.g.

- <Activity>-<Name>

Specific examples of this would be:

- Training-AdvancedOpenMP
- AppBenchmark –VASP
- KernelBenchmark -Spectral

Each of these would be an individual repository within a PRACE GitLab installation or a PRACE GitHub Organization.

In summary, different PRACE 4-IP workgroups will cooperate (e.g. WP6, WP4 and WP7) in order to evaluate the final structure of the repositories.

As we can see there is a wide range of technologies and solutions used in different sites, be it PRACE, CoEs or other entities. HPC-Forge was a solution had provided a good interface, but is now deprecated and abandoned due to technical reasons Experience with the take-up of HPC-Forge also shows the importance of collaboration with different institutions to get the best possible tool and wide adoption.

5.2.4 Analysis of existing solutions

We have analysed many solutions like SVN, Google Code, GitLab, GitHub, etc. but finally only GitLab and GitHub, both commercial and free versions have been considered. These are the most used tools and since we want to promote wide adoption we must stick to them. Both are very good solutions, with a lot of features, very extensible, and fit very well to our initial requirements.

GitHub

GitHub is a web-based interface for accessing the git protocol. It provides a rich interface and a good set of features. This solution is only free of charge for open-source projects. GitHub repositories provide the following services and features:

- Repository push access over HTTPS (username and password) and SSH (mediated via key pairs)
- Web interface to repositories
- Issue/bug tracker/feature requests for each repository
- Wiki for each repository that supports a wide range of editing formats (e.g. Markdown, reStructuredText, Org-Mode)
- Activity monitoring/reporting and repository log
- Hooks to perform actions on external services (e.g. HTTP POST, AWS, Twitter, Jenkins) based on repository events
- Multiple options for automated deployment

GitHub supports also the concept of Organizations. An organization can contain an unlimited number of user accounts and an unlimited number of public repositories. A GitHub Organization provides the following services:

- Create unlimited number of teams to control access to repositories and delegate organization control
- Invite exiting GitHub users to join a repository
- Report on Organization activity
- Collect all repositories that are part of the Organization

Full documentation of Organizations can be found on the GitHub website [53]. Individual user accounts within the Organization can be granted different roles with regard to the Organization and with regard to individual repositories within the Organization. Each role allows a different set of permissions to be enforced for the account. These roles are organised through Teams:

- Owner Team has full access to pull from, push to and alter settings for all repositories in the organisation and alter settings and membership of the organization.
- Admin Team (for particular repositories) has full access to pull from, push to and alter settings for the specified repositories.
- Write Team (for particular repository or repositories) has access to pull from and push to specified repositories.
- Read Team (for particular repository or repositories) has access to pull from specified repositories.

The full set of Team roles and permissions are described on the GitHub website [54]. For many developers, the primary method of accessing GitHub will be through the git command line interface. The Eclipse IDE also integrates with GitHub. Windows and Mac clients are also available for GitHub if a user does not want to use the command line or install Eclipse. GitHub supports pull of data over HTTP (all repositories are public unless you use the paid-for service) and push of data over HTTPS and SSH. There's also a website available to use all the features and administer the repositories.

Costs related to GitHub

- Capital Costs

  - Access to GitHub for Organizations is free - you only pay if you wish to host non-Open Source software.
  - If we wished to have provision for private repositories the GitHub pricing plan can be found at GitHub pricing webpage [57].
  - As a quick guide: for 10 private repos in the Organization you would pay 25 USD/month. This comes to 300 USD/year/10 repos capital costs.

- Personnel Costs:
  A small amount of recurring effort is required in an oversight role of the Organization to ensure that the correct users have the correct permissions and requests for access are actioned. Depending on the SLA that PRACE wishes to attach to such a service these costs will vary. An estimation of the staff effort required to provide a basic level of service with the following service level is **1.2 PM/year**:

  - Requests for access and for changes in permissions have a response time of 1 week.
  - An ongoing effort level of 0.1 FTE (0.5 days a week) would be a generous amount of effort to maintain such a service and would lead to a high quality of service for users that would encourage uptake and foster confidence in the community. The service could be run on less effort but with detrimental effects on uptake and community perception.

GitHub Advantages

- Organizations are simple to create (takes less than 5 minutes/each)
- Rich set of roles and permissions to control Organization and repository access
- Existing GitHub accounts can be used by users
- Existing repositories can be added to Organizations
- No requirement to host own service
- Familiar GitHub interface and repository access methods
- Only public repositories allowed – encourages Open Source development
- Repositories also include:
  - Online wiki for documentation and project information
  - Online issue tracker
  - Online repository analysis tools for tracking activity
- Supports git and svn repositories
- Infrastructure will persist beyond the lifetime of PRACE (4IP) project
- Low level of PRACE staff effort required to administer

GitHub Disadvantages

- No access to private repositories (unless using paid service)
- Cannot be linked to custom authentication methods

GitLab

GitLab is 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. In the following section we analyse GitLab hosted on the cloud, and GitLab hosted in-house.

GitLab 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 Roles and Permissions
    - Breakdown of permissions are accessible in GitLab documentation [55]. Structure is the same than the one presented for Owner-Admin-Write-Read permission for GitHub.
    - 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 [56]. One can promote specific members of the group for specific projects by adding them also as a member of a project.

The interface and access methods are exactly the same as for GitHub.

Costs related to GitLub

Hosting GitLab on a server operated by PRACE:

- Hosting costs are necessary for GitLab instance installation and first configuration which is important before first usage of repository.
- Administration costs might be larger than administering GitHub. Required manpower effort is 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.

Available frameworks:

- Community Edition: Free to download and operate
- Enterprise Edition: It takes some extra fee for extra features something like Multi LDAP server support, LDAP group synchronization, Audit log. The price of license depends of the support level.

GitLab.com offers three level of support:

- BASIC
    - Fee: $39 yearly per user (purchased in packs of 10 $390 per pack)
    - Extras: next business day support, Permission management, Extended authentication
- STANDARD
    - Fee: $49 yearly per user (purchased in packs of 100 $4900 per pack)
    - Extras: 24/7 Emergency support, High availability (Zero downtime upgrade), Live upgrade assistance
- PLUS
    - Fee: $149 yearly per user (purchased in packs of 100 $14900 per pack)
    - Extras: Premium support, Dedicated Account manager

Hosting repository on GitLab servers (online at GitLab.com) would involve the following costs:

- Hosting fee: zero because GitLab do this on Amazon Web Service and Azure
- Administration fee: same as GitHub. 0.1 FTE => 12PM per year
- Extra fee (if necessary): $9.99 yearly per user in pack of 20 users for next-business day support (GitLab.com Bronze Support)

GitLab advantages:

- 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 add-ons and freedom of group management and private repositories
- Mobile apps
- Option to integrate with LDAP/connect with PRACE LDAP and use PRACE user base
- Integration of data into other sites (e.g. PRACE web, training portal, etc.) is possible and customizable
- Built-in 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
- Has an enterprise edition with a similar set of features as GitHub enterprise for much lower prices
- 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

GitLab.com advantages:

- Runs on Enterprise Edition of GitLab
- Don't need installation only sign up above for a free account
- Free unlimited public and private repositories, unlimited collaborators, issue tracking and wikis
- 10GB disk space/project
- Unlimited total disk space

GitLab disadvantages:

- One-time effort of deployment and configuration

- Requires operation effort to run (deployment and operation however might be covered as a planned WP6 effort)
- Requires hosting (there are numerous PRACE services hosted by PRACE partners independently from IPs)

GitLab.com disadvantages:

- Repositories maintained by GitLab which means dependency
- Unable to customize deeply, like custom plugin installation, etc.
- Unable to integrate to external Auth providers (e.g. PRACE userbase, Edugain, etc.)

Automated build/deploy services

A comparison between auto-build and auto-deploy technologies has also been done. All the analysed technologies can be connected to GitLab or GitHub.

Git-Auto-Deploy [58]

This service acts like a web server and supports executing shell commands whenever each repository changes. It is compatible with both GitLab and GitHub frontends.

Buildkite (formerly BuildBox) [59]

Automation and collaboration tools for building and shipping software are available.

Jenkins [61]

Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by providing powerful ways to define your build pipelines and integrating with a large number of testing and deployment technologies. As a reference, it's being used by IT4/VSB.

Built-in continuous integration tool

GitLab CI [60]

It is a continuous integration tool, with support to do various tests and deployments of updated code. It is available for both Community and Enterprise edition. There is no similar functionality in GitHub.

Project Management Tools

There are also project tools that can be integrated with the VCS frontend service.

Atalassian JIRA [62]

Integration with GitHub and GitLab Enterprise Edition is possible, but it is a payment solution. It has a project planner, bug/issue tracker, software release control, report tool. It's widely used and good rated tool. Costs can be found at its website. Its mean price is about $60 USD per user/per year.

Redmine [63]

This is not a real solution for GitHub, but it has integration with GitLab. Redmine provides Gantt tool (Project Management), Bug Tracker, Wiki, etc. It's free and it is self-hosted solution.

5.2.5 Specifications of the service

Thanks to the deep analysis performed on previous sections and the feedback received, we concluded that the best option would be to implement a prototype using the following solutions and technologies:

- Source control management: A system that provides a central place where the team members can store and access their entire source code base.
- Requirements management: A system used for recording and tracking product feature requests.
- Bug-tracking: A system used to record and track errors and feature requests.
- Automated build: A system that builds the application every night by automatically executing the required build procedure steps at the scheduled time, without any human intervention. Automated testing: The tools that team developers and testers use to verify software and to detect and prevent software problems, such as functionality errors, reliability problems, performance problems, or security vulnerabilities.
- Regression testing: Any tool or combination of tools that can automatically run all of your existing tests on your entire code base on a regular basis (preferably nightly, as part of the automated build). Its purpose is to help you identify when code modifications cause previously working functionality to regress, or fail. For example, the regression system may be a script that runs one or more automated testing tools in batch mode.
- Project Management: TRAC and Redmine will be implemented and evaluated in the pilot to see which the most efficient tool for our needs is.

Chosen HPC services

- Source control management: GitLab self-hosted version.
- Issue management, Bug-tracking: GitLab integrated system.
- Automated build and testing: GitLab CI. We can also try Jenkins.
- Project Management: TRAC and Redmine.

The decision of using GitLab self-hosted edition is based on the fact that recurrent costs are fixed without surprises. There’s no per project size limitation and the extension of features is not a problem. Also, the integration with a future PRACE federation will be possible at no economic direct costs.

Anyway, moving from GitLab to GitHub or other technologies has been studied, and tools exist to help on such process.

In terms of authentication, a simple LDAP server will be configured and set as alternative authentication method. Other methods like Google Oauth will also be provided.

Finally, using GitLab integrated tools in place of external tools will provide more consistency and will decrease administration costs as well as provide the benefit of uniformity of the repository.

5.2.6 Prototypal implementation

For the first implementation of the prototype some site should provide two virtual machines and a public IP address plus a proxy, in order to be able to set up a highly available service pilot. In case only a basic prototype should to be deployed, only one virtual machine and a public IP address will be needed.

Connection to the PRACE network is not required for basic authentication, but it would be worth to have it in order to test LDAP integration with x.509. Further investigations on certain plugins to enable this authentication method will be performed by WP6.

The implementation phase is divided into 5 different tasks:

- T.1 Policies Definition, we will define and detail all related policies of the services, from access policies, to usage and data policies, accounting, etc.
- T.2 Served Deployment, consists in configuring a test virtual machine and the setup of the platform where the services will be installed.
- T.3 Services Deployment/Configuration/Testing, is the major phase of the prototype implementation. Here we will install all the services commented above, we will configure them, and we will test them.
- T.4 Communication: We must continue the communication between work packages, projects and CoEs in order to get their feedback and requirements.
- T.5 Evaluation of the prototype: Will involve different partners, and will be a phase where projects will upload code and test the platform, giving feedback to WP6 Task 2.

An initial assignment of tasks for each subtask has been done.

In the planning only the implementation of the pilot is taken into account. Extra work like deliverables, meetings, teleconferences, etc. are not included.

Task Name	Resource Name
T.1 Policies definition	
T.1.1 Access Policies	EPCC
T.1.2 Authentication Policies	EPCC
T.1.3 Accounting Policies	EPCC
T.1.4 Usage Policies	EPCC
T.1.5 Data Policies	EPCC
T.2 Server Deployment	
T.2.1 OS Installation & Configuration	GRNET
T.2.2 Network & DNS configuration	GRNET
T.3 Services Deployment/Configuration/Testing	
T.3.1 GitLab	NIIF
T.3.2 Redmine	GRNET
T.3.3 Trac	GRNET
T.3.4 GitLab CI	EPCC
T.3.5 Jenkins	EPCC
T.3.6 LDAP server	BSC
T.4 Communication	
T.4.1 CoEs	GRNET
T.4.2 WP4-WP7	NIIF
T.4.3 Other projects & institutions	NIIF
T.4.5 Usage Guide	GRNET
T.5 Evaluation of the prototype	
T.5.1 CVS System	BSC
T.5.2 Project Management tool	BSC
T.5.3 Continuous Integration system	BSC
T.5.4 Policies	BSC
T.5.5 Access and Authentication	BSC

Table 6: WP6.2 Service 4: Project Plan

Taking into account that resources are not dedicated full time a day, an estimated schedule that should be accomplished is presented below:

Task Name	Start Date	Completion Date
T.1 Policies definition	16/11/2015	16/02/2016
T.2 Server Deployment	16/11/2015	23/12/2015
T.3 Services Deployment	02/01/2016	01/05/2016
T.4 Communication	16/11/2015	End of Project
T.5 Evaluation of the prototype	01/02/2015	End of Project

Table 7: WP6.2 Service 4: Project schedule