



**SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures**

**INFRA-2012-2.3.1 – Third Implementation Phase of the European
High Performance Computing (HPC) service PRACE**



PRACE-3IP

PRACE Third Implementation Phase Project

Grant Agreement Number: RI-312763

**D6.3.2
Second Annual Technology Report**

Final

Version: 1.0
Author(s): Michael Rambadt, JUELICH
Date: 23.6.2014

Project and Deliverable Information Sheet

PRACE Project	Project Ref. №: RI-312763	
	Project Title: PRACE Third Implementation Phase Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: < D6.3.2 >	
	Deliverable Nature: <DOC TYPE: Report / Other>	
	Deliverable Level: PU*	Contractual Date of Delivery: 30 / 06 / 2014
		Actual Date of Delivery: 30.06.2014
EC Project Officer: Leonardo Flores Añover		

* - The dissemination level are indicated as follows: **PU** – Public, **PP** – Restricted to other participants (including the Commission Services), **RE** – Restricted to a group specified by the consortium (including the Commission Services). **CO** – Confidential, only for members of the consortium (including the Commission Services).

Document Control Sheet

Document	Title: Second Annual Technology Report	
	ID: D6.3.2	
	Version: <1.0 >	Status: <i>Final</i>
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2007	
	File(s): D6.3.2.docx	
Authorship	Written by:	Michael Rambadt, JUELICH
	Contributors:	Agnes Ansari, IDRIS Gabriele Carteni, BSC Giuseppe Fiameni, CINECA Bartosz Kryza, PSNC Zoltan Kiss, NIIF Philippe Prat, CINES Thomas Röblitz, UiO Ilya Saverchenko, LRZ Frank Scheiner, HLRS Björn Schembera, HLRS
	Reviewed by:	Andreas Schott, RZG Florian Berberich, JUELICH
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	26/May/2014	Draft	Initial version
0.2	06/June/2014	Draft	Contributions for, gsatellite, iRODS, New File Transfer technologies (update)
0.3	10/June/2014	Draft for internal review	Contributions for iRODS (update)
0.4	17/June/2014	Final Draft	Included review comments
1.0	23/June/2014	Final version	

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure, Second Annual Technology Report
------------------	--

Disclaimer

This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement n° RI-312763. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements. Please note that even though all participants to the Project are members of PRACE AISBL, this deliverable has not been approved by the Council of PRACE AISBL and therefore does not emanate from it nor should it be considered to reflect PRACE AISBL's individual opinion.

Copyright notices

© 2014 PRACE Consortium Partners. All rights reserved. This document is a project document of the PRACE project. All contents are reserved by default and may not be disclosed to third parties without the written consent of the PRACE partners, except as mandated by the European Commission contract RI-312763 for reviewing and dissemination purposes.

All trademarks and other rights on third party products mentioned in this document are acknowledged as own by the respective holders.

Table of Contents

Project and Deliverable Information Sheet	i
Document Control Sheet.....	i
Document Status Sheet	ii
Document Keywords	iii
Table of Contents	iv
List of Figures	vii
List of Tables.....	vii
References and Applicable Documents	vii
List of Acronyms and Abbreviations.....	ix
Executive Summary	1
1 Introduction	1
2 DECI Portal	2
2.1 Status	2
2.2 Planned Extensions.....	2
3 Inca	3
3.1 Description	3
3.2 Status	3
4 Service Certification.....	4
4.1 Status	4
4.2 Planned Extensions.....	4
5 DMOS development and deployment.....	5
5.1 Reasons, benefits and constraints.....	5
5.2 Status	6
6 GridFTP Usage Number Collection	7
6.1 Standard Logfiles	7
6.2 Implemented Database Approach.....	8
7 New File Transfer Technologies.....	9
7.1 Methodology.....	9
7.2 Quantitative Assessment	10
7.2.1 ARC	10
7.2.2 BBCP.....	10
7.2.3 GlobusOnline.....	11
7.3 Qualitative Assessment	12
7.4 Conclusion	13
8 Gsatellite.....	13
8.1 Reasons, Benefits and Constraints	13
8.2 Evaluation of the Service	13
8.2.1 Setup of test environments	14
8.2.2 Environment independent tests	14
8.2.3 Environment dependent tests.....	15
8.3 Conclusion	17
9 File System Technologies	17

9.1	Use-case and Purpose of the Evaluation.....	17
9.2	Ceph.....	17
9.3	Test Environment.....	18
9.4	Testing Methology.....	18
9.5	Summary of Results.....	19
9.5.1	Read, write and delete tests.....	19
9.5.2	Stream tests.....	19
9.5.3	IOzone tests.....	19
9.5.4	Server load.....	19
9.6	Conclusion and future Work.....	20
10	iRODS.....	20
10.1	PRACE Data Management Services.....	20
10.2	iRODS user documentation and best practices guide.....	20
10.3	iRODS Testbed setup.....	20
10.4	PRACE/EUDAT collaboration.....	21
10.4.1	PRACE evaluation of the EUDAT implementation of iRODS.....	21
10.4.2	Evaluation of the PRACE iRODS implementation.....	21
10.5	10Gbit/s performance tests.....	22
10.6	Feature evaluation.....	22
11	Collaboration with other projects.....	22
12	Conclusion.....	23
13	Annex.....	24
13.1	Methodology to assess new file transfer technologies.....	24
13.1.1	Definitions.....	24
13.1.2	Hardware Requirements.....	25
13.1.3	TCP Buffer Size.....	25
13.1.4	MTU and Jumbo Ethernet Frames.....	25
13.1.5	Disk performance.....	26
13.1.6	Network capacity.....	26
13.1.7	Requirements summary.....	26
13.1.8	Methodology.....	26
13.1.9	Production Conditions.....	27
13.1.10	Data sets.....	27
13.1.11	Workload.....	27
13.1.12	Parallel Streams.....	27
13.1.13	Qualitative Factors.....	28
13.1.14	Test cases.....	28
13.1.15	Data sheet (template).....	29
13.2	Detailed Ceph Testing methology and Results.....	31
13.2.1	Read and write test by uncompressing the Linux kernel tree.....	31
13.2.2	Unbuffered stream tests with dd.....	32
13.2.3	IOzone tests.....	32
13.2.4	Server side load.....	35
13.3	iRODS user documentation and best practices guide.....	37
13.3.1	Getting an iRODS account.....	37
13.3.2	Accessible data servers.....	37
13.3.3	Initial iRODS setup.....	37
13.3.4	Configuring the connection to a data server.....	37
13.3.5	Accessing your data, data storage and retrieval.....	38
13.3.6	Storing data to data server.....	39

13.3.7	<i>Retrieving data from a data server</i>	40
13.3.8	<i>Managing archive files.....</i>	40
13.3.9	<i>Sharing data.....</i>	43
13.3.10	<i>Synchronizing data.....</i>	45
13.3.11	<i>Adding metadata and searching.....</i>	45
13.3.12	<i>Accessing a EUDAT storage.....</i>	46
13.3.13	<i>Data servers policy and accessibility.....</i>	47
13.3.14	<i>Data allocations.....</i>	47
13.3.15	<i>Data organization into iRODS collections.....</i>	48
13.3.16	<i>Data management rules</i>	48
13.3.17	<i>Data transfer tips</i>	48
13.3.18	<i>File names.....</i>	48

List of Figures

Figure 1: DMOS homepage showing an overview of the latest maintenance and downtime announcements.....	5
Figure 2: Page for creating a record for a new maintenance or downtime announcement in DMOS.....	7
Figure 3: Results for ARC evaluation inside Hungary.....	10
Figure 4: Test results for BBCP between CEA and CINES. Constant performance measures over all test cases.....	11
Figure 5: Test results of GlobusOnline between CINECA and EPCC. GlobusOnline shows poor behaviour when using 4 parallel streams.....	11
Figure 6: Memory allocation check used to monitor memory allocation.....	15
Figure 7: Memory check script output.....	15
Figure 8: Example tgftp benchmark result diagram (local transfer speeds at CINECA).....	16
Figure 9: Narrow and Tight Links.....	24
Figure 10: NIIF cluster CPU load.....	32
Figure 11: IOzone Read test results.....	33
Figure 12: IOzone Write test results.....	33
Figure 13: Random read tests with IOzone.....	34
Figure 14: Random write tests with IOzone.....	34
Figure 15: NIIF cluster Network I/O.....	35
Figure 16: SURFsara cluster Network I/O.....	36
Figure 17: SURFsara cluster CPU Load.....	36

List of Tables

Table 1: Description of the keys in a logfile.....	8
Table 2: DB layout showing the netlogger key, the DB key and the variable type used.....	8
Table 3: Involved partners in the New File Transfer Technologies task.....	9
Table 4: Qualitative assessment ranked from 1 (very bad) to 5 (very good) and with individual comment.....	12
Table 5: Setup of <i>Gsatellite</i> test environments.....	14
Table 6: iRODS testbed characteristics.....	21
Table 7: Measures Definition.....	24
Table 8: Requirements List.....	26
Table 9: Each test case includes at least 18 runs.....	29
Table 10: Data sheet template filled in with sample data.....	30

References and Applicable Documents

- [1] <https://bscw.zam.kfa-juelich.de/bscw/bscw.cgi/840521>
- [2] <http://netlogger.lbl.gov/>
- [3] http://wiki.nordugrid.org/index.php/ARC_middleware
- [4] <http://www.slac.stanford.edu/~abh/bbcp/>
- [5] <https://www.globusonline.org/>
- [6] <http://www.prace-project.eu/IMG/pdf/d6.3.1.pdf>
- [7] <http://fasterdata.es.net>
- [8] <http://fasterdata.es.net/host-tuning/linux/expert/>
- [9] <http://fasterdata.es.net/host-tuning/linux>
- [10] <http://www.psc.edu/index.php/networking/641-tcp-tune>

List of Acronyms and Abbreviations

AAA	Authorization, Authentication, Accounting
AMD	Advanced Micro Devices
BSC	Barcelona Supercomputing Center (Spain)
CINECA	Consorzio Interuniversitario, the largest Italian computing centre (Italy)
DEISA	Distributed European Infrastructure for Supercomputing Applications. EU project by leading national HPC centres.
DMOS	Distributed Maintenance Information Organisation System
EPCC	Edinburg Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
FTP	File Transfer Protocol
GB	Giga (= $2^{30} \sim 10^9$) 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, managed by DANTE. GÉANT2 is the follow-up as of 2004.
GridFTP	Certificate based File Transfer Protocol
HPC	High Performance Computing
HLRS	Höchstleistungsrechenzentrum Stuttgart (Germany)
HBP	The Human Brain Project
ICHEC	Irish Centre for High-End Computing
IDRIS	Institut du Développement et des Ressources en Informatique Scientifique (represented in PRACE by GENCI, France)
Inca	Service monitoring tool used in PRACE
IPsec	Internet Protocol Security
iRODS	integrated Rule-Oriented Data-management System, a community- driven, open source, data grid software solution
ISTP	Internal Specific Targeted Project
LRZ	Leibniz Supercomputing Centre (Garching, Germany)
MPI	Message Passing Interface
Mb/s	Mega (= 10^6) bits per second, also Mbit/s
NIIF	Nemzeti Információs Infrastruktúra Fejlesztési Intézet (National Information Infrastructure Development Institute, Hungary)
OpenMP	Open Multi-Processing
SDSC	San Diego Supercomputer Center
SURFsara	The national HPC and e-Science support center of the Netherlands
TB	Terra (= 10^{12}) Bytes (=8bits), also TByte
Tier-0/-1	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
UiO	University in Oslo
UNICORE	Uniform Interface to Computing Resources. Grid software for seamless access to distributed resources.
WAN	Wide Area Network
XSEDE	Extreme Science and Engineering Discovering Environment (NSF- funded US-Project)

Executive Summary

Task 6.3 ‘Technical evolution of the PRACE operational services’ in Work Package 6 ‘Operation of the Distributed Research Infrastructure’ of PRACE-3IP continues the work from PRACE-1IP (WP6, Task 6.3). It also took over the technical evolution of the PRACE operational services from work package 10 ‘Advancing the Operational Infrastructure’ in PRACE-2IP after the end of WP10.

This deliverable describes the work of Task 6.3 in WP6 of the second year.

In the first year Task 6.3 started to evaluate a couple of software already. Some investigations were finished in the first year already. The *gsatellite* and the *DMOS* activity have been continued.

After the handover from PRACE-2IP WP10, WP6.3 continued the work to improve the existing infrastructure with the *Inca* enhancements, the implementation of the *DECI-Portal*, and the collaboration with other technologically oriented projects, like *EUDAT*, *EGI*, and *XSEDE*. Also the continuation of the Data-Services with *iRODS-Data-Management*, *New File Transfer Technologies*, and the evaluation of file system technologies was part of WP 6.3 work in the second year.

1 Introduction

The objectives of Task 6.3 in the second year were:

- Continuation of the technology watch for the extension and completion of existing PRACE services including the continuation of the work of WP10 of PRACE 2IP
- Work on PRACE internal software needs

Chapter 2 describes the work in developing the DECI portal to provide an online DECI peer review process to the users.

The *Inca* tool is essential to monitor the functionality and the status of the PRACE software. The enhancements are described in chapter 3.

The service certification started in PRACE-2IP WP10 already and has been continued in PRACE-3IP Task6.3. The results are described in chapter 4.

In a big research infrastructure –as PRACE is– it is essential to have a formalized way of announcing and managing machine maintenances. Therefore the *DMOS* tool was evaluated. The results are outlined in chapter 5.

To get some statistics about the GridFTP usage and the network traffic the task “GridFTP Number Usage Collection” (see chapter 6) has been introduced.

New in the second year was the task “New File Transfer Technologies”, taken over from WP10 of PRACE-2IP. Chapter 7 describes the work on the performance and usability of several promising file transfer tools like *BBCP*, *Globus online* and *ARC*.

Gsatellite as explained in chapter 8 is a tool for scheduling remote file transfers. This is essential for an optimized use of the (PRACE) network.

For data management and transfer it can be helpful to take advantage of a distributed files system. In the “File System Technologies” task (chapter 9), Task 6.3 concentrated on the evaluation of *Ceph* as such a file system, identified as the currently most promising one.

An essential building block of the PRACE data management, e.g. concerning the collaboration with EUDAT, is the iRODS tool (see chapter 10).

As part of the collaboration between PRACE and XSEDE a joint call to enhance the interoperability between both infrastructures has been launched. The results are described in chapter 11.

2 DECI Portal

During the PRACE All-Hands-Meeting in Paris (September 2012) the decision was taken to provide a DECI Project Proposal Revision portal that is based on the already existing PRACE Tier-0 peer review tool. The actual adaptation work from the Tier-0 tool started on 6th April 2013 in close contact with WP2's DECI program manager. This application was then used for the first time on the DECI 11th call for proposals, open from 6th May to 14th June . A steering committee was set-up for accompanying and driving the development and features.

Members of the Steering Committee are:

- WP2 Chris Johnson, Petri Nikunen (DECI management);
- WP10 Jules Wolfrat, Andreas Schott (technical observers);
- CINES team (implementors).

Development, technical support, and maintenance are provided by the CINES Team:

- Philippe Prat - prat@cines.fr (Project leader);
- Fabien Cadet - cadet@cines.fr (Developer);
- Florent Marceteau - marceteau@cines.fr (Developer).

This application uses PHP with Symfony2 framework and MySQL. The source code license allows usage and possible improvement by and for members of the PRACE community with no commercial use.

2.1 Status

As of May 2014, the tool implements the DECI peer review process with fully on-line proposals submission along with technical and scientific review assessment. It covers the following workflow:

- Electronic submission of HPC project;
- Technical evaluation (TE) where all relevant data from proposal are visible in TE form;
- Scientific Evaluation (SE) where evaluators can get limited access to relevant proposals and TE.

The progress of the peer review process for a given call is mainly tracked in some sort “master spreadsheet“ page containing all relevant data on the status of proposals and reviews.

2.2 Planned Extensions

- Interfacing with the DPMDB tool for project follow-up

- Interfacing with central PRACE LDAP for the integrated authentication of DECI staff users
- Interfacing with the GridSAFE/DART accounting infrastructure for integration with the awarded projects resources and their consumption.

3 Inca

PRACE monitoring infrastructure is based on Inca, an application for user-level monitoring. Inca implements a client-server model, where Inca reporter manager clients, deployed on all PRACE resources, are testing capabilities of PRACE e-Infrastructure and send collected monitoring data to the Inca server for processing, archival and presentation. Inca functionality was extended and adapted over the course of the project to satisfy requirements of the evolving PRACE infrastructure. These changes addressed client and server functionality and operation.

3.1 Description

Inca tests availability, functionality and performance of services using test scripts called reporters. Inca comes with a set of reporters designed to test common services. Many of these reporters were extended to provide required functionality and to integrate with services and utilities available in PRACE. New reporters were implemented in accordance with requirements defined in the PRACE Service Catalogue. Inca reporters developed in PRACE are used to test the following capabilities and services: PRACE Common Production Environment (PCPE) components, Globus GSISSH, Globus GridFTP, LDAP and UNICORE. Inca tests are configured to take into account scope of services, differentiating between PRACE internal, open access, door node and central services. Availability and functionality of GSISSH and GridFTP services are monitored over the PRACE dedicated network and the public Internet. Furthermore GridFTP data transfer performance is measured and monitored over both networks depending on service availability.

3.2 Status

Multiple custom modifications of the Inca web-interface were implemented to improve presentation of monitoring results and achieve integration with PRACE Wiki and TTS systems. These integration efforts allowed Inca to display resource and service maintenance information. Furthermore integration with PRACE TTS supports creation and reference of trouble tickets for problems detected by Inca directly through Inca web-interface.

Inca client and server components were updated to support latest versions of the OpenSSL library and SHA256 X.509 certificates. This was necessary to meet new requirements of the European authentication infrastructure and maintain security of inter-component communication.

Over the course of the project the number of configured test instances increased so that currently Inca executes thousands of tests every day. This high data rate revealed limitations in scalability of Inca server components. To address this Inca server architecture was improved and a new version implementing a refined task scheduling algorithm was deployed in PRACE.

Inca modifications implemented in PRACE were done in coordination with Inca developers from the San Diego Supercomputer Center (SDSC). Feature enhancements and other changes

realized in PRACE are reviewed by SDSC and, if applicable, are carried over to Inca development trunk.

4 Service Certification

The Service Certification activity in Task 6.3 of PRACE-3IP is a continuation of the same activity in Task 10.1 in PRACE-2IP. The goals of Service Certification are to verify deployed services before offering them to users, ensuring that technical requirements (e.g. non-functional requirements) are satisfied and thus improving the quality of deployed services. The Service Certification procedures are complementary to the live monitoring of the infrastructure in the sense that certification tests are only performed when indicated, for instance after deploying new services or after major maintenance changes to a service have been performed.

4.1 Status

Each certification procedure is defined in the PRACE wiki in the form of checklists with steps necessary for verifying each service. For some services the certification procedures are automated using shell scripts or INCA reporters which can be executed through the existing monitoring infrastructure. Certification results are stored internally on the wiki.

Currently, the certification procedures for the following services from the PRACE Service Catalogue are defined in the wiki:

- Uniform access to HPC
- PRACE internal interactive command-line access to HPC systems
- **Data transfer, storage and sharing**
- Authentication
- Authorization
- Accounting
- **GridSAFE Accounting repository**
- Network management
- Monitoring
- **Software Management and Common Production Environment**

For services marked in bold, certification tests are implemented either as scripts or in case of GridSAFE in the form of an Excel sheet.

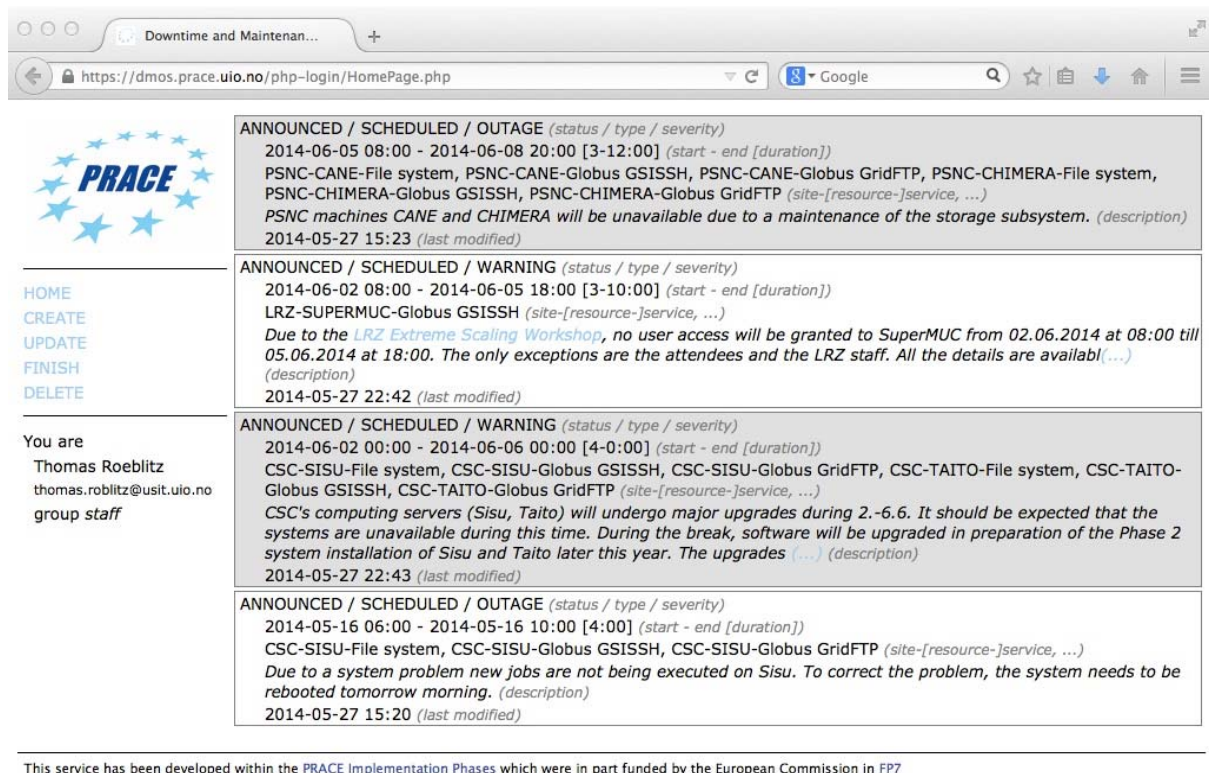
The main challenges of the certification procedure so far were related to the very high heterogeneity of the sites (different compilers, libraries, and versions), making it difficult to create fully automated test scripts. Missing environment variables in the PCPE containing the paths to actual compilers and libraries on each site (e.g. PRACE_C, PRACE_CXX, PRACE_F90) made it necessary to create a configuration map of compiler and library versions for all sites, which then can be used by the test scripts.

4.2 Planned Extensions

The future work will include implementation of automated scripts for other services, most importantly for the PRACE internal interactive command-line access to the HPC systems, uniform access to HPC, covering Authentication and Authorization issues in detail.

5 DMOS development and deployment

The Distributed Maintenance Information Organisation System, *DMOS*, is a service to announce and manage information about sites, resources and services in the PRACE infrastructure. The service is structured in three tiers: a relational database backend, a lightweight control layer exposing a REST API and a web-based frontend. The graphical user interface supports authentication, authorization and a variety of data views that focus on interests and requirements of PRACE stakeholders. PRACE administrators and users may use the frontend to manage and to search for information about downtimes of resources and services in a federated distributed environment. Figure 1 depicts the homepage of the frontend showing an overview of the latest announcements.



The screenshot shows a web browser window with the URL `https://dmos.prace.uio.no/php-login/HomePage.php`. The page features a PRACE logo on the left and a navigation menu with links: HOME, CREATE, UPDATE, FINISH, DELETE. Below the menu, the user's identity is shown as Thomas Roebnitz, group staff. The main content area displays four announcement cards, each with a status (ANNOUNCED, SCHEDULED, OUTAGE, or WARNING), a duration, a list of affected resources, a description, and a last modified timestamp.

<p>PRACE</p> <p>HOME CREATE UPDATE FINISH DELETE</p> <p>You are Thomas Roebnitz thomas.roblitz@usit.uio.no group staff</p>	<p>ANNOUNCED / SCHEDULED / OUTAGE (status / type / severity)</p> <p>2014-06-05 08:00 - 2014-06-08 20:00 [3-12:00] (start - end [duration])</p> <p>PSNC-CANE-File system, PSNC-CANE-Globus GSISSE, PSNC-CANE-Globus GridFTP, PSNC-CHIMERA-File system, PSNC-CHIMERA-Globus GSISSE, PSNC-CHIMERA-Globus GridFTP (site-[resource-]service, ...)</p> <p>PSNC machines CANE and CHIMERA will be unavailable due to a maintenance of the storage subsystem. (description)</p> <p>2014-05-27 15:23 (last modified)</p>
	<p>ANNOUNCED / SCHEDULED / WARNING (status / type / severity)</p> <p>2014-06-02 08:00 - 2014-06-05 18:00 [3-10:00] (start - end [duration])</p> <p>LRZ-SUPERMUC-Globus GSISSE (site-[resource-]service, ...)</p> <p>Due to the LRZ Extreme Scaling Workshop, no user access will be granted to SuperMUC from 02.06.2014 at 08:00 till 05.06.2014 at 18:00. The only exceptions are the attendees and the LRZ staff. All the details are availabl(...)</p> <p>(description)</p> <p>2014-05-27 22:42 (last modified)</p>
	<p>ANNOUNCED / SCHEDULED / WARNING (status / type / severity)</p> <p>2014-06-02 00:00 - 2014-06-06 00:00 [4-0:00] (start - end [duration])</p> <p>CSC-SISU-File system, CSC-SISU-Globus GSISSE, CSC-SISU-Globus GridFTP, CSC-TAITO-File system, CSC-TAITO-Globus GSISSE, CSC-TAITO-Globus GridFTP (site-[resource-]service, ...)</p> <p>CSC's computing servers (Sisu, Taito) will undergo major upgrades during 2.-6.6. It should be expected that the systems are unavailable during this time. During the break, software will be upgraded in preparation of the Phase 2 system installation of Sisu and Taito later this year. The upgrades (...)</p> <p>(description)</p> <p>2014-05-27 22:43 (last modified)</p>
	<p>ANNOUNCED / SCHEDULED / OUTAGE (status / type / severity)</p> <p>2014-05-16 06:00 - 2014-05-16 10:00 [4:00] (start - end [duration])</p> <p>CSC-SISU-File system, CSC-SISU-Globus GSISSE, CSC-SISU-Globus GridFTP (site-[resource-]service, ...)</p> <p>Due to a system problem new jobs are not being executed on Sisu. To correct the problem, the system needs to be rebooted tomorrow morning. (description)</p> <p>2014-05-27 15:20 (last modified)</p>

This service has been developed within the PRACE Implementation Phases which were in part funded by the European Commission in FP7

Figure 1: DMOS homepage showing an overview of the latest maintenance and downtime announcements.

The REST API is currently used by the web-based frontend only, but may be used by other PRACE services, in particular, the INCA monitoring service. Persistent storage of information describing downtime of resources is provided by the database backend.

The *DMOS* development started during DEISA2 to provide an easy to use and flexible solution for management of maintenance information. Finally it will replace the Wiki-based information system by a relational database.

5.1 Reasons, benefits and constraints

Currently the PRACE operations team relies on a dedicated section in the PRACE Wiki for announcement and documentation of service and resource maintenances. Each partner is instructed to publish information about scheduled and unplanned maintenances in this Wiki section. Maintenances can be announced at a site, resource or, in special cases, service level and contain the following details:

- Scheduled start of a maintenance

- Scheduled end of a maintenance
- Time of the actual maintenance ending
- Affected sites and resources
- Maintenance description

The provided information describes the general availability of services and resources and allows PRACE operations team to plan support, deployment and maintenance activities. Furthermore this information helps PRACE Operator on Duty to appropriately react to problems with the e-Infrastructure, for instance by appropriately treating service failures caused by unavailability of the respective resource.

Maintenance information collected in PRACE Wiki has been turned out as not sufficient for a detailed analysis of the real time e-Infrastructure state. For example:

- Creation and update of maintenance announcements can only be performed manually
- Information provided in the maintenance description field is often not suited for end-users
- Service and resource dependencies cannot be specified in the PRACE Wiki

DMOS addresses these limitations and provides added flexibility by supporting standard access interfaces and extended functionality.

During the second year of PRACE-3IP, GOCDB was evaluated whether it could be used as an alternative to *DMOS*. Although GOCDB has been in use for several years within Grid environments (e.g., EGI), it was concluded that it does not provide the additional features required for the benefit of the PRACE infrastructure. Capabilities such as automatic notifications for upcoming downtimes and maintenances are not part of GOCDB, but only provided through additional components which use the information of the GOCDB backend. In addition, by not relying on such an external service, *DMOS* may be tailored more easily and more quickly to fit the specific needs of the PRACE infrastructure.

5.2 Status

DMOS has been developed by the three PRACE partners IDRIS (France), LRZ (Germany) and UiO (Norway). The **DMOS** backend and REST API are hosted at LRZ, while the frontend is hosted at UiO.

The *DMOS* functionality was successfully evaluated against the following regular and usual operational requirements:

- Provide information about scheduled maintenances and unscheduled downtimes
- Store information in a format suitable for manual and automatic processing
- Contain information about sites, resources and services
- Support several granularity levels for maintenances and downtimes (outage / warning for sites, resources and services)
- Offer persistent data storage
- Provide and open interfaces to other tools and services, for instance monitoring, user support, etc.

The evaluation results were presented and discussed with PRACE operations team and necessary adaptors to the database backend, the REST API and the web-based frontend were developed. **Figure 2** shows the page for creating a record for a new event.

The screenshot shows a web browser window with the URL `https://dmos.prace.uio.no/php-staff/StaffCreateMessage.php`. The page features the PRACE logo and a navigation menu with links: HOME, CREATE, UPDATE, FINISH, DELETE. A user profile section identifies the user as Thomas Roebnitz (`thomas.roblitz@usit.uio.no`) in the `group staff`. The main form contains three dropdown menus for Status (set to 'announced'), Severity (set to 'outage'), and Type (set to 'scheduled'). Below these are input fields for 'Begin' and 'End' times. Two text areas are provided for 'Description for staff (classified)' and 'Description for users (public)'. A table titled 'Select services and sites/resources that are affected' lists various sites and resources with checkboxes for selection. The table columns include Site[/Resource], GSISSH, GridFTP, RFT, WS_GRAM, Unicore, LDAP, GPFS, PRACE Net, Public Net, File system, and DPMDB. The rows listed are: BSC/MARENOSTRUM, BSC/MINOTAURO, CASTORC/CY-TERA, CEA/CURIE, CINECA/FERMI, CINECA/PLX, CINES/JADE, CSC/SISU, CSC/TAITO, CYFRONET/ZEUS, EPCC/ARCHER, EPCC/BLUEJOULE, FZJ/JUQUEEN, and HI DS/HEPMT.

Figure 2: Page for creating a record for a new maintenance or downtime announcement in DMOS.

6 GridFTP Usage Number Collection

GridFTP is used as the core service for data transfer in PRACE. GridFTP includes logging of all data transfers. This can provide powerful information, including number of transferred bytes as well as the duration of the transfer. With this information, conclusions on network usage, directions of transfers, etc. can be made.

6.1 Standard Logfiles

Information is logged to human readable logfiles by default. That makes an automated processing difficult. Hence another logging methodology is possible, called netlogger [2], which provides information in key/value pairs. An example line of a transfer then looks as follows:

```
DATE=20080630235211.369377 HOST=somehost.edu PROG=globus-gridftpserver
NL.EVNT=FTP_INFO START=20080630235211.339810 USER=someuser
FILE=/tmp/x BUFFER=87872 BLOCK=262144 NBYTES=262144 VOLUME=/
STREAMS=1 STRIPES=1 DEST=[127.0.0.1] TYPE=RETR CODE=226
```

Table 1 gives a brief description of the keys used by the netlogger.

Netlogger Key	Description
USER	User name
START	Start date of a transfer
DATE	End date of a transfer
HOST	Local host

DEST	Remote host
TYPE	Type of transfer like <i>RETR</i> or <i>STOR</i>
CODE	FTP result code like 226 or 4xx
BLOCK	Local block size
BUFFER	Local buffer size
STREAMS	Number of data streams
STRIPES	Number of stripes
NBYTES	Number of transferred bytes
FILE	File name

Table 1: Description of the keys in a logfile

At first glance, automated analysis of such logfiles seems easy. But when using a splitted GridFTP configuration, where the backend and the frontend reside on different hosts, the information is scattered between them, thus only disjointed information is available. This also holds for multiple backends and parallel streams on one physical backend machine. In such cases the value of the *USER* key always equals to *globus-mapping* and the *DEST* key will have the value *[0.0.0.0]*.

6.2 Implemented Database Approach

At first, the database layout for storing information is presented, on which was agreed on the Barcelona F2F meeting in March 2014. This is presented in Table 2.

Netlogger Key	DB Key	Values
-	pk	Primary Key
-	netlogLineHash	Text
START	transferStart	Date
DATE	transferEnd	Date
HOST	hostLocal	Text
DEST	hostRemote	Text
TYPE	transferType	Text
CODE	resultCode	Text
BLOCK	transferBlocksize	Integer
BUFFER	transferTCPBuffersize	Integer
STREAMS	transferStreams	Integer
STRIPES	transferStripes	Integer
NBYTES	transferNumberOfBytes	Integer

Table 2: DB layout showing the netlogger key, the DB key and the variable type used

Most information from the logfiles can be directly inserted into the according database fields. User as well as file names are not stored due to possible privacy restrictions.

In the case of a non-splitting configuration, information can be transferred to the database in a quite straightforward manner: For each line, a database insert is created holding all the information from the key/value pairs.

If a splitted configuration of the GridFTP infrastructure exists, the idea is to correlate the frontend and the backend logfiles.

First, the data is read line by line from both frontend and backend logfiles separately and is then stored in a distinct DB table.

After that first step, the two tables are correlated: Each transfer is uniquely identified by the *NBYTES* and the *FILE* value, so for each entry in the frontend logs, the corresponding entry in the backend logs is searched and then combined. The combined log entry is then stored in a main table. Since the filename is needed for the correlation, the filename is also stored and thrown away after combining the logs for reasons of anonymity.

The field *netlogLineHash* is built as sha1 hash and used for checking, if an entry already exists. All the scripts include checking and removing of double entries.

7 New File Transfer Technologies

The objective of this subtask is the investigation of new software tools that are able to provide high performance bulk data transfer (comparable to GridFTP). The aim is to achieve a quantitative assessment, where performance numbers are measured, as well as a qualitative assessment to gain better insights to the software reliability.

Table 3 shows the tools evaluated and the sites involved in the testing:

Tool	Involved Partners
ARC	NIIF, UiO (PRACE-2IP only)
BBCP	CEA, CINES, EPCC (PRACE-2IP only)
GlobusOnLine	CINECA, EPCC

Table 3: Involved partners in the New File Transfer Technologies task

The task coordination was conducted by HLRS, taking over the duty from BSC in January 2014. Work was already started in PRACE-2IP under WP10 and then continued in PRACE-3IP.

7.1 Methodology

The methodology was defined by Gabriele Carteni (formerly BSC) and is documented in the Annex (see 13.1). In the following, the approach will be described briefly:

First, requirements were defined to establish comparable conditions for testing. The requirements covered TCP buffer size, MTU size (resp. Jumbo frames), disk performance as well as network capacity.

Then, test workloads were defined based on the following three parameters: number of files (1 or 100), size of each file transferred depending on the total workload (100GB, 500GB, 1000GB), and the number of parallel streams (4, 8, 16). This lead to 18 test runs to be executed for each tool.

Besides pure performance measures, it has been considered as valuable to take into account also qualitative factors. The following factors have been ranked from 1 (really bad) to 5 (really good) for each tool along with a short comment specifying the motivation of the mark.

- **Reliability:** This indicates the ratio how often transfers failed.
- **Footprint/Intrusiveness:** How much additional treatment/setup is required on the system or for the users to be able to use the tool?
- **Maintenance:** How much maintenance effort is needed for the tool?
- **Fault Tolerance:** How can the tool deal with errors (auto-restart etc.)?
- **Code Maturity:** Is the code well-written, maintained and documented?
- **Community Acceptance:** Is or will the tool be accepted by the PRACE community?

7.2 Quantitative Assessment

7.2.1 ARC

ARC [1] tests were conducted between the two PRACE-connected NIIF HPC sites, SC and SEGED in Hungary. The distance between the two sites is ~170 km and they are connected with a 10 GB/s Ethernet link.

Tests results were obtained as shown in **Figure 3**. The results indicate that ARC suffers from performance problems in case many small files are transferred. Only 2 out of 9 test cases with 100 files show satisfactory performance behaviour, whereas performance with only one large file (100GB, 500GB or 1TB file) is mostly constant at about 550Mb/s.

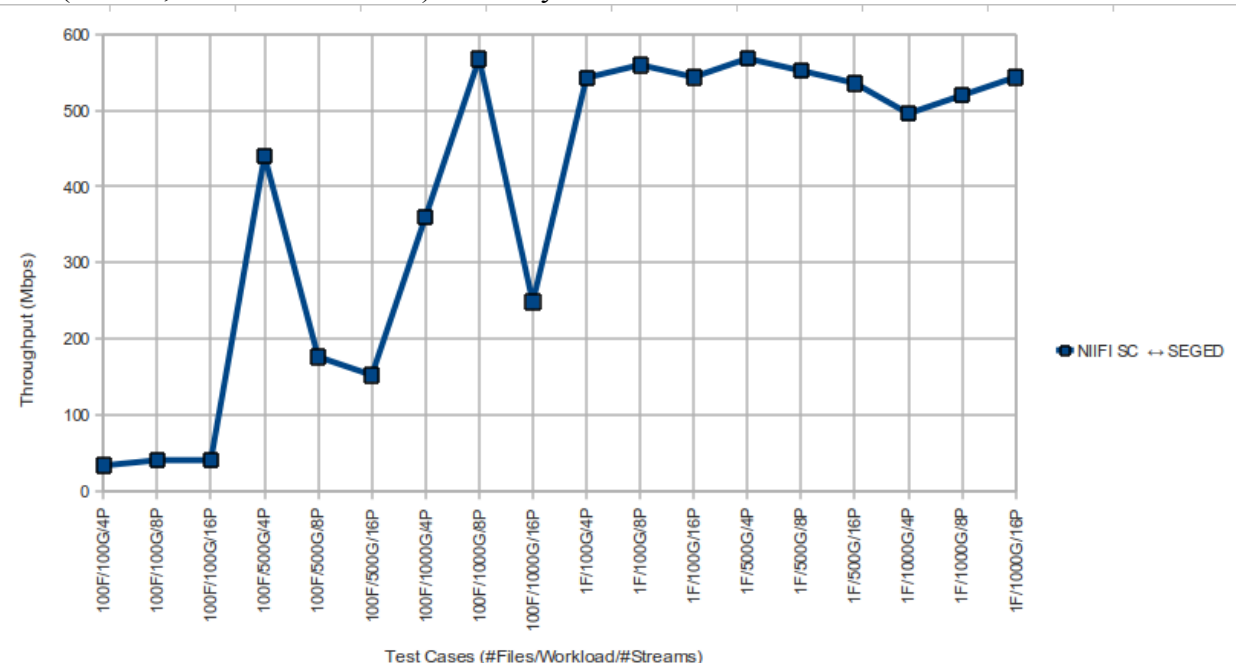


Figure 3: Results for ARC evaluation inside Hungary.

7.2.2 BBCP

The tool BBCP [4] was evaluated between CINES and CEA. The tool shows constant performance measures for all test cases as depicted in Figure 4 .

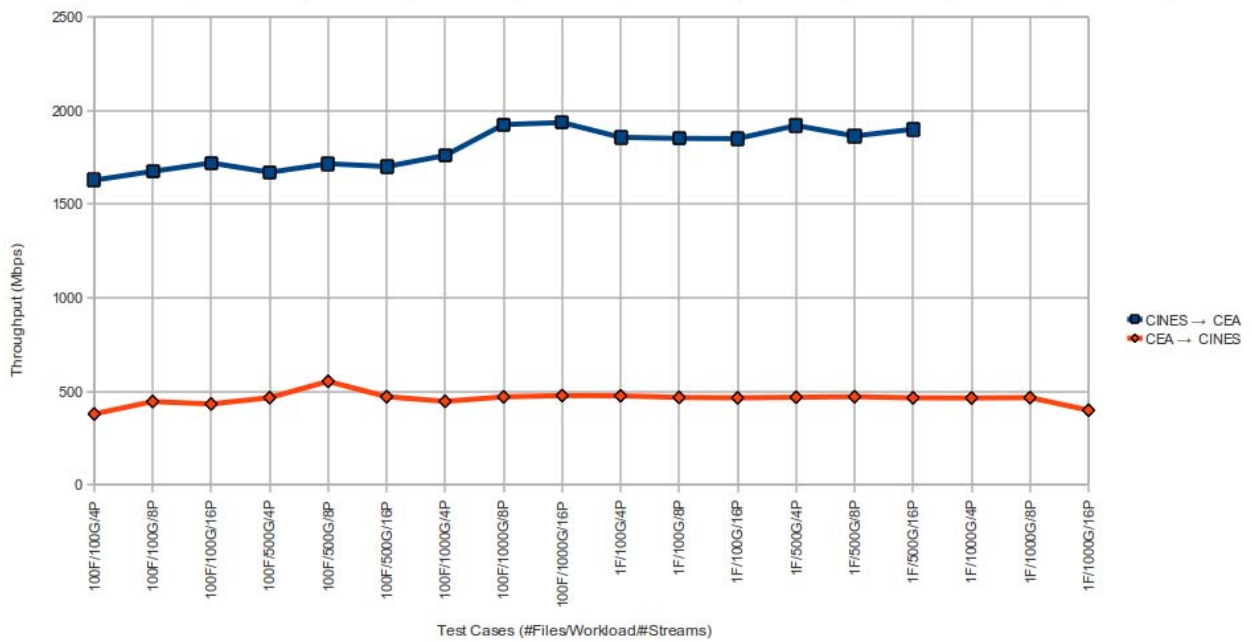


Figure 4: Test results for BBP between CEA and CINES. Constant performance measures over all test cases

7.2.3 GlobusOnline

GlobusOnline [5] was evaluated between EPCC and CINECA, the results are shown in Figure 5. As shown the GlobusOnline performance is poor for all cases with 4 parallel streams. Furthermore, the performance differs significantly depending on the number of streams used.

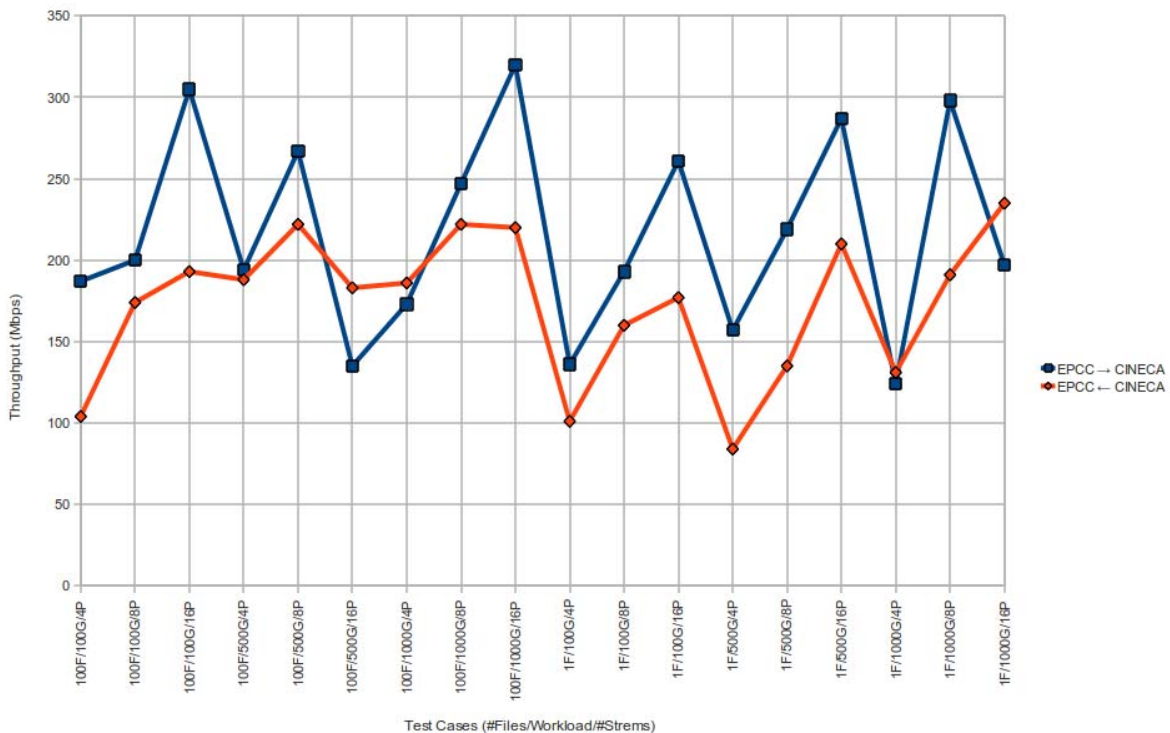


Figure 5: Test results of GlobusOnline between CINECA and EPCC. GlobusOnline shows poor behaviour when using 4 parallel streams

7.3 Qualitative Assessment

	ARC	BBCP	GlobusOnLine
Reliability	4: All files transferred successfully	2: Crashes for transfer of 500GB and fail to transfer 1TB file. Error : “bbcp: Copy process 4497 was killed via signal 9 bbcp: Connection reset by peer writing to stream.”	5: No problem during the test activity.
Footprint Intrusiveness	5: None: It does not require additional permissions for a user. The user is able to create proxy certificate and call the transfer function.	5: Minimal: it does not require administrative rights or system servers, e.g. can be installed by user.	4: Minimal: if GridFTP servers are available, no more administrative rights are required In order to enable GO to move data inside PRACE network a getaway is required, i.e. a simple port forwarding rule.
Maintenance	5: No maintenance required by system administrators.	5: No maintenance required by system administrators.	5: No maintenance required by system administrators.
Fault Tolerance	5: It was trying to handle all network connection failure.	1: The tool doesn't provide failure restart capabilities.	5: The tool provide auto-restart tool in case of network or service problems and is able to renew the user's proxy in case it expires before the end of the transfer.
Code Maturity	5: Very mature: First version released at 2004. We used very fresh release. It was released 1 month ago, it is well maintained and ~10 years old	3: Settled: first version released in 2011	4: Quite mature: even if the software is not that old and it undergoes frequent changes, the reliability is seldom affected.
Community Acceptance	4: Medium: X509 certificate required for a usage.	4: Medium: Number of users is growing as reported by PRACE partners for their local scientific communities.	3: Medium: Some of the PRACE sites don't like the use nor of proxy neither of a closed foreign tool such as GlobusOnline.

Table 4: Qualitative assessment ranked from 1 (very bad) to 5 (very good) and with individual comment

7.4 Conclusion

From the measurement results it can be obtained that ARC is only performing well and showing a constant behaviour for large files, whereas for smaller files the transfer rate is not satisfying. However the testing partners ranked ARC in the qualitative assessment to be reliable, easy to handle for both users and administrators and fault tolerant.

BBCP showed constant performance over all test sets, but it was ranked poor for fault tolerance and reliability.

The performance of GlobusOnline always dropped when only 4 parallel streams were used. Regarding the qualitative assessment only community acceptance was ranked poor due to the open security questions related to GlobusOnline.

Therefore, none of the tested tools can be recommended to replace the current GridFTP/gtransfer data transfer solution deployed in the PRACE environment.

8 Gsatellite

Gsatellite is a client toolkit for scheduling data transfer tasks like batch jobs. It allows users to submit and manage large GridFTP data transfers running non-interactively in the background.

The evaluation process consisted of a planning phase, the setup of test environments including the relevant network connections, and the detailed testing of *gsatellite*, especially concerning the needs of the PRACE community.

The partners in this task are NIIF, HLRS, CINECA, and CSC.

8.1 Reasons, Benefits and Constraints

Currently, PRACE users using GridFTP need to perform data transfers manually and to monitor them from the beginning to the end. This can be a rather expensive task, especially in the case of:

- long transfers of big data without the need of user interaction
- regularly scheduled transfers (e.g. every Friday at 3 pm)
- transfers that have to be done within a specific timeframe (e.g. only after 10 pm and before 6 am)

Gsatellite provides this functionality in an automated way. Additionally, *gsatellite* adds enhanced reliability to data transfers because it retries them automatically in case of temporary errors. Users can log into one or multiple (frontend) machines, submit their data transfer jobs, leave, and let *gsatellite* take care of the remaining work. They do not have to be online during the data transfer but can return at any time and check the status of their jobs. If required, email notifications can also be activated to get information about the job status.

8.2 Evaluation of the Service

The evaluation started in June 2013 and was conducted in four phases

- Planning phase
- Setup of the test environment
- Environment independent evaluation

- Testing of installation process and compatibility with different operating systems and software environments
- Testing of general usage and user experience
- Testing of general stability and resource requirements
- Environment dependent evaluation
 - Detailed testing of the *gsatellite* features (e.g. support for benchmarking, automatic restart for data transfers, etc.)

The planning phase consisted of creating the work plan above, the specification of testing environments and of environment independent and environment dependent tests. Planning was finished in the first year of the project.

8.2.1 Setup of test environments

The following requirements had to be satisfied for test environments:

- The *git* versioning system has to be supported
- A login for all team members must be possible on all test machines
- The team members can write to a shared directory and execute predefined *gsatellite* services, e.g. email notifications
- Each test machine has a connection to both the PRACE internal network and to the public internet
- Each test machine has a connection to the PRACE GridFTP servers

A centralized git repository was used to set up test environments. This allowed for an easy deployment of new revisions to all sites. The test environment has been set up at NIIF, CSC and CINECA. Connections to the PRACE Network were established for all sites to ensure constant and high quality connectivity for the tests.

Site	Location	Server Address
NIIF	Budapest, Hungary	prace-login.sc.niif.hu:2222
CSC	Kajaani, Finland	sisu.csc.fi:{22,2222}
CINECA	Bologna, Italy	{gssh.fermi.cineca.it,gssh-prace.cineca.it}:2222

Table 5: Setup of *Gsatellite* test environments

8.2.2 Environment independent tests

In order to gain a general overview, without detailed testing of the transfer features, standalone reviews were made of the installation process, usage experience and resource management of *gsatellite*. Review documents were created for all tests.

First, the installation process and compatibility with different operating systems and software environments were evaluated. Installation tests were performed on Debian, Fedora, OpenSUSE, CentOS and the OS X operating systems, separately testing global and single user installations.

Documentation of the installation was clear and straightforward. Software requirements are easy to be satisfied, as only the *bash* shell and core Linux utilities are required.

The general usage and user experience review was performed by testing the user interface features and functionality from a usability perspective. The advantage of the UI is its similarity to the (Open)PBS UI and commands, which are widely known amongst HPC users. The built-in help is easily understandable.

Some smaller remarks on error handling and improvement requests for the program output for success and failure were forwarded to the developer of *gsatellite*.

The general stability and resource requirement evaluation has been performed by issuing an increasing number of jobs at the same time and checking their status and memory usage. This functionality was tested by using data transfer jobs that copied 500 GB of data from source to destination.

```
#!/bin/bash
function
echo 0 $(cat /proc/${1}/smaps | grep Private | awk '{print $2}' | sed 's#^##+' | bc
)
single_mem(){
}
total_mem=0
count=0
for i in `ps auxx | grep gsatellite | grep -v grep | awk '{print $2}'`
do
do
#echo
#cat
#echo 0 $(cat /proc/${i}/smaps | grep Private | awk '{print $2}' | sed 's#^##+' | bc
this_mem=`single_mem
total_mem=`echo "${total_mem} + ${this_mem}" | bc`
let count+=1
done
done

echo "The total memory used by ${1} processes (and ${count} subprocesses) is
${total_mem} kB!"
true
```

Figure 6: Memory allocation check used to monitor memory allocation

No problems were detected with the stability; the queuing mechanisms handled the load well. And the resource consumption was very low, even with lots of jobs. The memory allocation is not increasing when increasing the number of jobs as seen on Figure 7.

```
The total memory used by 0 processes (and 2 subprocesses) is 1728 kB!
The total memory used by 1 processes (and 7 subprocesses) is 5952 kB!
The total memory used by 2 processes (and 7 subprocesses) is 5952 kB!
The total memory used by 3 processes (and 7 subprocesses) is 6016 kB!
The total memory used by 4 processes (and 7 subprocesses) is 6016 kB!
```

Figure 7: Memory check script output

8.2.3 Environment dependent tests

During environment dependent evaluation, benchmarking tests were created to evaluate, how *gsatellite* can support benchmarking tasks and to test other features like the *services facility* (which includes notifications and the execution of arbitrary scripts on specific events like *job start* or *job termination*).

The *tgftp* benchmarking test was created to evaluate

- The notification of the user when the benchmark has finished

- The automatic sending of a tarball with job output via email (reduced test).
- The automatic processing, plotting and sending of benchmark results via email plus sending of a tarball with job output via email (full test).

Tests were run by two different partners and both the reduced and full tests were successful. The latter sent Box-and-Whisker diagram plots of the measured values. This evaluation showed the potential of *gsatellite* for benchmarking tests.

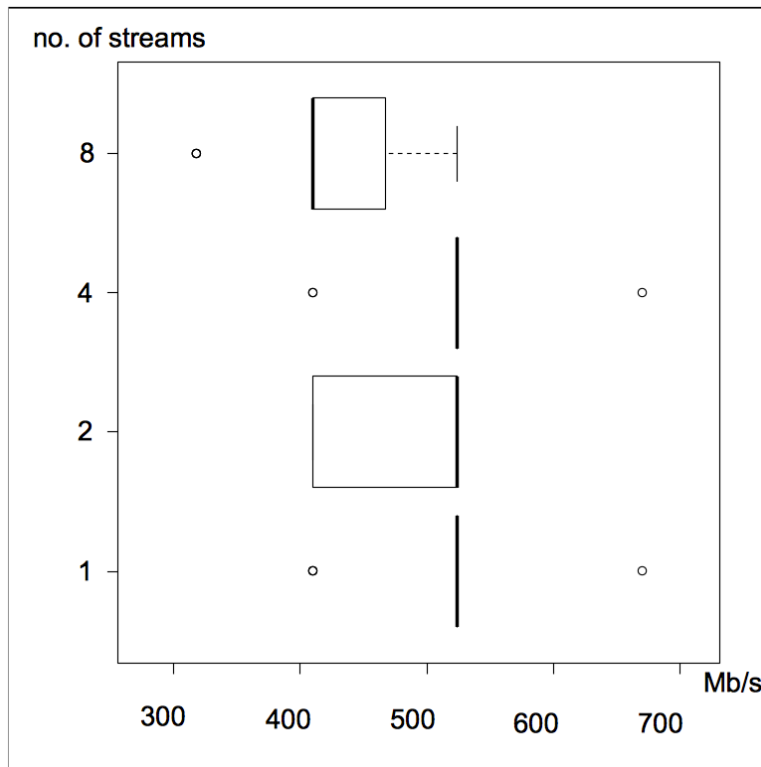


Figure 8: Example tgftp benchmark result diagram (local transfer speeds at CINECA)

gsatellite is basically a job scheduler, but together with tools like tgftp (a tool for GridFTP benchmarking) and gtransfer it can serve benchmarking and data transfer tasks well. A *gsatellite* job is just an arbitrary script that is scheduled and executed non-interactively in the background.

To evaluate how *gsatellite* could support data management tasks, several *gtransfer* [6] (a tool for GridFTP data transfers) jobs were created that evaluated the automatic restarting of data transfer jobs by *gsatellite* and that should determine if inter-network data transfers could also be managed by *gsatellite*.

The auto-restart tests show the automatic restart of *gsatellite* that have exited due to temporary errors during the file transfer. It does not restart *gtransfer* jobs that exit due to persistent errors like wrong usage, etc.

Temporary errors were triggered by killing the GridFTP data transfer process(es) at the destination node two times, because *globus-url-copy* (the underlying data transfer tool used by *gtransfer*) was configured to automatically retry failed transfers at least once. Usage errors were generated by providing wrong arguments to *gtransfer*.

Both tests yielded the expected results and this clearly indicates that *gsatellite* can be useful when transfer conditions are not ideal, providing automatic retries for transfer tasks that fail due to temporary errors.

The inter-network tests were configured to transfer data through a third, temporary location between source and destination. This is useful, when there are network constraints for the source or the destination hosts, which do not allow direct transfers between the two nodes or limit the possible bandwidth. In PRACE, there are nodes with data transfer services, which can only be reached from within the PRACE network and there are other ones, which have no such limitations.

A *dpath* file can be used by *gtransfer* to make sure the transfers are going inter-network. The tests were successful and showed that *gsatellite* can also manage inter-network data transfers. This is a very advantageous real usecase for PRACE, and supports users of systems with network constraints clearly.

8.3 Conclusion

WP6.3 members concluded that *gsatellite* will bring an additional benefit to the PRACE infrastructure and propose it as a PRACE production service.

The evaluation showed that the data job scheduling functionality and the possibility to automate GridFTP jobs will be an essential advantage for the user when transferring data in the PRACE infrastructure.

9 File System Technologies

In this chapter the basic framework for the evaluation of the selected distributed file systems is described. From possible use-cases the technical requirements are derived. Then the methodology for testing, the measurement metrics, the results and finally some conclusions are presented.

In the last deliverable the Ceph filesystem has been identified as one of the best candidates for future file system technologies. Here now Ceph has been evaluated over the PRACE network and its capabilities over the WAN have been tested.

9.1 Use-case and Purpose of the Evaluation

A common use-case for a distributed file system would be the sharing user specific personal and configuration data between HPC systems. This would allow users to compute on HPC systems of different sites during or in consecutive DECI calls more easily, since data would be accessible from more than one HPC site. Such functionality could also reduce the need for user initiated data transfers, leading to several copies of the same data in different locations.

Another possible use-case could be to give the users a common home directory – same on all sites – with some preconfigured scripts and configuration files to provide them a very similar environment on every PRACE system. This facilitates again the old DEISA philosophy, where users could maintain just one home directory shared on the HPC systems.

9.2 Ceph

Ceph is a storage platform designed to present a scalable object, block, and file storage from a distributed cluster without a single point of failure. The *Ceph* server cluster consists of three main components:

- OSD: The OSD Daemon stores data and handles data distribution
- Monitors: The Monitor maintains maps of the cluster state

- MDS: The Metadata Server stores metadata on behalf of the *Ceph* Filesystem

Ceph stores client data as objects within storage pools using the so-called CRUSH algorithm that enables the cluster to scale, rebalance, and recover dynamically. Clients connect to the monitor nodes and read or write data from or to the OSDs. In case of monitor or OSD errors the clients automatically do a failover to the remaining cluster parts.

9.3 Test Environment

In order to test the distributed nature of *Ceph*, real clusters have to be setup on both, client and server side.

Two test environments were deployed to test *Ceph* capabilities on the WAN, one at SURFsara, and the other one at NIIF.

The distributed *Ceph* server is deployed at SURFsara on a production cluster. It consists of eight Dell R720 XD with 9 data disks of 3 TB each and 2 SAS disks for data journaling that are configured as follows:

- 3 monitor nodes, with 9 OSD daemons each
- 4 OSD nodes, with 9 OSD daemons each
- Each OSD daemons has his journal on a SAS disk

The client cluster is deployed by NIIF on four virtual machines. Ansible-based configuration automation scripts were developed for easier deployment and tracked in a git repository. *Ceph* version 0.72 and CentOS 6.5 are used in all test environments. The server and client cluster were connected through the PRACE network and monitored with Ganglia hosted on an Apache webserver.

9.4 Testing Methology

In this phase the goal was to test and evaluate the *Ceph* block storage sub-system over the WAN. The following common steps were done:

- Determination of the optimal hardware configuration for the server and client cluster
- Installation and configuration of CentOS with Ansible scripts
- Connecting the clusters to the PRACE network and testing the connectivity
- Installation of the *Ceph* server cluster and the clients
- Sharing *Ceph* server administrator key and configuration
- Creating block devices on the clients
- Mapping and mounting block devices on the clients
- Running standard filesystem tests

The following file system tests have been performed in order to model the real life usage scenarios as closely as possible:

- Read/Write test by uncompressing the linux kernel tree
- Deletion of these uncompressed files
- Unbuffered stream test with dd
- IOzone tests

9.5 Summary of Results

For detailed test methodology descriptions and result graphs, see 13.1.

9.5.1 *Read, write and delete tests*

In order to model and test file creation performance the duration of uncompressing and then deleting the current Linux kernel tree have been measured. The performance of the remote *Ceph* cluster has been compared with the local disk. In the tests the remote *Ceph* storage was in average 2 times slower than the local one.

The creation and deletion tests were looped and ran for a day on each client simultaneously to stress-test the cluster. The wait load was high and the overall write performance was around 35MB/s which is considered as reasonably good since the client cluster was connected with a single 1Gbit/s Ethernet-connection.

9.5.2 *Stream tests*

Stream write tests were done with not buffered dd write. First, the remote and local performance has been compared. Remote stream tests were in average 2 times slower than the local ones. Large file tests were done by writing a 90GB files on each client simultaneously.

9.5.3 *IOzone tests*

Finally, IOzone test were performed to cover various block and file sizes and to get an overall picture of the block storage performance.

Replica size (the desired number of copies of an object) on the block storage subsystem was set to 3. This implies that the overall server network input is 3 times the overall output of the clients. The overall server network output is 2 times the overall output of the clients. The optimal block size is peaked at around 4MB. Write operations have a slow down at the same block size.

The bottleneck for the network traffic is the client cluster uplink which provides around 100-120MB/s. A sustained 35MB/s write with high wait load on the clients has been measured, which is most probably caused by the virtualization layer.

9.5.4 *Server load*

On the server side the overall CPU load was low and the network load was at least twice the client output.

Inspecting SURFsara cluster load graphs, the following conclusions can be made:

- *Ceph* equally divides the workload over the nodes.
- CPU load is barely influenced. The peak usage was just 4%.
- Cluster network load shows ~70 MB/s with a peak to ~150 MB/s.
- The per node graphs indicate ~12 MB/s per node. In total, this amounts to roughly 1 Gbit/s, equalling the NIIF - SURFsara connection maximum.
- Network load was the same for each node (even for monitors)
- Memory cache graphs show that all nodes cache data.

9.6 Conclusion and future Work

Ceph is a mature open source storage cluster alternative that is well usable on large scale production systems. The block storage capabilities of *Ceph* have been tested. The results indicate that the block storage subsystem is scalable and stable even when used over WAN and can be recommended as good alternative to commercial iSCSI systems. The server side load was low. In order to best facilitate *Ceph*'s fault-tolerant nature, the server cluster is ideally equipped with a high bandwidth and low latency internal replica network.

Besides the object store, *Ceph* has a POSIX compliant file system layer which is in experimental state. In case of successful tests of the POSIX layer this could open the way to use *Ceph* as a central distributed file storage in PRACE.

10 iRODS

This section describes the iRODS deployment in PRACE. The focus is on the technical work covering

- the PRACE and EUDAT collaboration
- the 10Gbit/s performance tests
- the evaluation on PRACE required features.

10.1 PRACE Data Management Services

HPC projects require both a computational and a data infrastructure that have to be managed in a combined way. The computation always produces output data, but also relies on input data eventually distributed on external resources.

The WP6.3 PRACE data management training team provides training and support on data management tools like iRODS so that users can fully benefit from the PRACE infrastructure and its data management environment when running their applications. This extends the well-established PRACE application support, which covers compute related application enabling and enhancement, with data support functionalities.

10.2 iRODS user documentation and best practices guide

To support the deployment of an optional iRODS service, PRACE WP6.3 provides iRODS user documentation and a best practices guide, see annex 13.3. This document describes the standard *icommands* usage for storing, retrieving and managing data with standard iRODS storage resources.

10.3 iRODS Testbed setup

An iRODS testbed was set up at CINES, CINECA, IDRIS, and NIIF, which has been used for the

- performance tests between PRACE and EUDAT (see section 10.4)
- 10Gbit/s performance tests within PRACE
- evaluation of specific features required by PRACE

The testbed characteristics are described in the following table:

Site	1 Gbe Internet Address	10 Gbe PRACE Address	Port	Zone	Version	iRODS Resources	Outgoing IP	Remark
IDRIS	N/A	turing2-d.idris.fr	1247	IDRIS	iRODS 3.3	demoResc(default) idrisData		iRODS/GSI GT 5.2.4 gridFTP (DSI 1.6.1) on port 1249
CINECA	N/A	prace07.fermi.cineca.it	1247	CinecaPRACE	iRODS 3.3	PRACEresc(default)	130.186.26.1	GridFTP (DSI) on port 2812
CINES	service4.cines.fr	jade-prace.cines.fr	1247	CINES	iRODS 3.3	cinesData(default)	196.63.184.4	GridFTP (DSI) on 2813
NIIF	irods01.niif.hu (193.224.66.219)	N/A	1248	NIIF	iRODS 3.3	niifData(default)		GridFTP (DSI) on 2811

Table 6: iRODS testbed characteristics

10.4 PRACE/EUDAT collaboration

The aim of this activity was to interoperate with the already available PRACE and EUDAT iRODS solutions. First the two solutions had to be analyzed.

10.4.1 PRACE evaluation of the EUDAT implementation of iRODS

When investigating in an iRODS data management solution for PRACE the iRODS team first checked the already available EUDAT iRODS concept. This is based on the iRODS Data Storage Interface (iRODS-DSI) as an extension to the GridFTP server to interact with iRODS. With this extension a GridFTP server can access an iRODS resource and provide it to any GridFTP client like an embedded file system. Thereby standard GridFTP client options, like `globus-url-copy`, can be used to transfer the data.

The EUDAT iRODS evaluation showed that this GridFTP/DSI module for iRODS provides a reasonable way to transfer data into/from the EUDAT infrastructure but does not provide all the features needed for a complete data management, which is essential for such a complex and heterogeneous infrastructure like PRACE.

The main constraints of the EUDAT DSI solution for PRACE are:

- For authentication the configuration files of both iRODS and GridFTP have to be fully identical at each site
- The EUDAT solution does not support file transfer to a remote iRODS server in a federated infrastructure
- The standard iRODS path cannot be used to transfer files to remote iRODS installations.

Therefore the WP6.3 iRODS team decided to implement a more flexible solution

10.4.2 Evaluation of the PRACE iRODS implementation

The goal of this evaluation was to demonstrate the usage of the standard iRODS commands between PRACE and EUDAT. It was also considered to be able to evaluate the iRODS protocol performances between both PRACE and EUDAT infrastructures. A standard iRODS installation using GSI (PRACE standard authentication mechanism) was planned to be used.

Such a solution would provide a simple way for PRACE users to store data into EUDAT storages. A similar mechanism could also be used by PRACE users to eventually store data into PRACE or communities' medium or long term storages, thus offering a homogeneous way for users to manage distributed storages in an easily extensible way, benefitting from the full iRODS capabilities.

Actually, this solution could not be tested with EUDAT, as the participating EUDAT sites indicated that the currently implemented iRODS clients cannot support these techniques, and so they were unable to provide an EUDAT testbed for such an evaluation.

But the PRACE internal testing proved the method as feasible, thus offering all benefits of iRODS data management. Although currently not applicable with EUDAT, this solution remains interesting and could become useful for other collaborations.

10.5 10Gbit/s performance tests

Several performance tests have been run inside the dedicated PRACE infrastructure covering various file sizes, data sets and number of threads. Three different tools (native iRODS, iRODS with the DSI module, and standard GridFTP) and also memory to memory transfers have been compared.

Since there could no time slots be reserved for the performance tests they had to be undertaken in the fully operational environment.

Due to not fully optimized network configurations not all tests have been running with the expected reliability and symmetric performance. Nevertheless, the test results showed that for large files ($\geq 1\text{GB}$) the iRODS put command *iput* benefits from the large number of threads (16) leading to a performance comparable to a GridFTP transfer. But even for smaller files, the tools seem to give rather similar results.

10.6 Feature evaluation

With version 3.3 iRODS introduced a couple of additional security configuration options, like improvements for the PAM/LDAP authentication. Now the lifetime of the PAM-derived iRODS password can be set. In addition, there is a configuration option, `PAM_AUTH_NO_EXTEND`, which disallows the extensions of the password. This functionality has been positively tested within PRACE.

Also new in version 3.3 are the so-called Workflow Structures Objects. It is now possible to chain workflows together by embedding one in another. The proper use of these new structures showed to be quite complex and therefore should be only implemented by iRODS administrators in a coordinated way between the collaborating PRACE sites.

11 Collaboration with other projects

An essential part of the Task 6.3 work is the collaboration with other projects as EUDAT, EGI, The Human Brain Project (HBP) and XSEDE.

PRACE and EUDAT are working on a joint call to be launched in September 2014 to grant PRACE users replication of their computational results onto EUDAT resources. In combination with this activity, the two projects are defining a more structured collaboration plan to be included in future proposals (next PRACE and EUDAT funding phase).

PRACE and HBP are preparing an agreement for the exploitation of the PRACE network to transfer data between HPC centers involved into both projects (CINECA, FZJ, BSC, CSCS).

PRACE and XSEDE provide peer reviewed access to high-end HPC resources and services both in Europe and the US. There have been a couple of collaborations, e.g. the common HPC Summer School, between both infrastructures in the last years already. In September 2013 PRACE and XSEDE decided to cooperate also by provisioning interoperable services. Therefore the XSEDE Senior Management Team and the PRACE Board of Directors decided

to publish a common call to research teams who require interoperable facilities between PRACE and XSEDE.

The call was published end of 2013. As a result eight proposals submitted. The technical review of the proposals was done by the PRACE WP6.3 and the XSEDE operation teams. At the end three proposals were selected that will get support up to six month to enable their interoperable applications. The selected proposals are:

- Smart Data Analytics for Earth Sciences across XSEDE and PRACE
- Interoperable High Throughput Binding Affinity Calculator for Personalised Medicine
- UNICORE Use Case Integration for XSEDE and PRACE

The activity started in April 2014 and it is an ongoing work and will be reported in the final WP6 deliverable D6.1.3 in M31.

12 Conclusion

The WP6.3 team worked on the evaluation of new technologies that are helpful for using the PRACE infrastructure. The *DECI portal* was introduced successfully for the DECI 11th call.

Inca, the central PRACE monitoring tool was enhanced permanently to insert new services.

WP 6.3 made progress in the *service certification* and developed a formalized way to proof the quality of a service.

The *DMOS* tool to announce maintenances and information about sites, resources and services is ready to use and will be prepared for production.

As a result from the last deliverable [6] the *Ceph* file system has been evaluated successfully for the use in distributed environments. If PRACE should introduce a storage cluster *Ceph* will be an adequate candidate. Depended on the POSIX layer tests *Ceph* also could be used as a future central distributed file storage in PRACE.

The main focus of the WP6.3 work was the PRACE data activity. Now it is possible to collect *GridFTP usage numbers*.

Three data transfer tools, *ARC*, *GlobusOnline*, and *BBCP* have been evaluated concerning the usability and performance in PRACE. The evaluation result is that none of these tools will be proposed as a PRACE production service since no advantages are expected for the users.

In contrast the *gsatellite* has been evaluated positively. WP6.3 is convinced that the user will benefit from scheduling data transfers and from automating GridFTP jobs. Therefore *gsatellite* will be proposed as a production service as well.

One important result of the *iRODS* activity was the development of the new usage and best practices guide (see 13.3).

The latest WP 6.3 activity was the collaboration with XSEDE. Three proposals were selected to enhance the interoperability of PRACE and XSEDE.

13 Annex

13.1 Methodology to assess new file transfer technologies

This document aims to define a common methodology for evaluating file transfer technologies that are new for PRACE, i.e. not yet officially supported. There are no specific technologies specified in this document since the methodology has designed to be independent from a specific software solution.

The main reference for this document has been a similar work being carried out by the Energy Sciences Network (ESnet) operated by LLNL and funded by the US DoE [7].

13.1.1 Definitions

The following table fixes some important definitions related to a file transfer activity that will be considered.

Measure	Definition (unit)
Capacity	Link Speed (Gbps)
Narrow Link	Link with the lowest capacity along a path [see Figure 1]
Capacity of the end-to-end path	Capacity of the Narrow Link
Utilized Bandwidth	Current Traffic Load
Available Bandwidth	= (Capacity) – (Utilized Bandwidth)
Tight Link	Link with the least available bandwidth in a path [see Figure 1]
Bandwidth Delay Product (BDP)	The number of bytes in flight to fill the entire path. BDP = (Capacity) * (RTT)

Table 7: Measures Definition

Figure 9 provides an example for determining narrow and tight links of a network path.

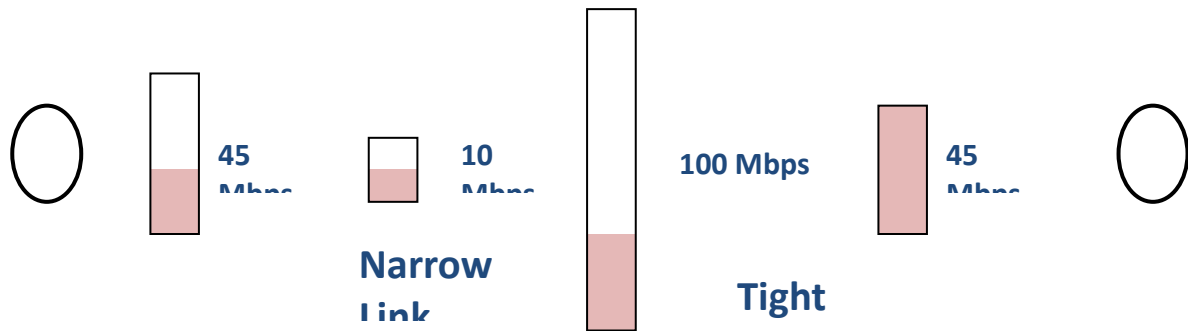


Figure 9: Narrow and Tight Links

Following the formula stated in Table 1, the BDP for a network with 1Gbps of capacity and 50ms of RTT is:

$$\text{BDP} = 1000\text{Mbps} * 0,05\text{s} = 50\text{Mb} (6,25\text{MBytes})$$

13.1.2 Hardware Requirements

It is assumed that different persons will be involved in the evaluation of different file transfer tools by using different network paths with unpredictable network conditions.

Defining hardware requirements for the tests is a solution in order to mitigate as much as possible the effect of different conditions. These requirements have been identified and described in the following sections.

13.1.3 TCP Buffer Size

A host system with a GNU/Linux operating system supporting TCP buffer auto-tuning must be used. Auto-tuning technique allows receiver buffer size (and TCP window size) to be dynamically updated for each connection maximizing the action of a congestion algorithm, which is recommended to be “cubic” or “htcp” [8].

Recent versions of Linux (version 2.6.17 and later) support auto-tuning with a default maximum value for the TCP buffer size of 4MByte (4194304 bytes)¹:

- memory reserved for TCP receiver buffers

```
user@sender_host:~# sysctl net.ipv4.tcp_rmem
net.ipv4.tcp_rmem = 4096 87380 4194304
```
- memory reserved for TCP sender buffers

```
user@sender_host:~# sysctl net.ipv4.tcp_wmem
net.ipv4.tcp_wmem = 4096 16384 4194304
```

It is suggested to increment the maximum value for both sender and receiver buffers, depending from the network card and the BDP measured. [9] and [10] help to check whether the maximum TCP buffer size is coherent with the measured BDP. As example, for a host equipped with 10G NIC and RTT delay below 100ms, is preferable to set a value bigger than 4MB (16MB or 32MB).

¹ To check if the auto-tuning is active, the file “/proc/sys/net/ipv4/tcp_moderate_rcvbuf” must be present and with value equal to 1.

13.1.4 MTU and Jumbo Ethernet Frames

Ethernet's maximum frame size of 1500 bytes is not optimized for Gigabit Ethernet network cards and can actually inhibit the ability of applications to take full advantage of a high network capacity.

This limitation can be overcome by changing the MTU to a value of 9000 allowing Ethernet frames with a payload of 9000 bytes. Assuming `eth0` as the name of the network interface, the MTU can be changed with the following command:

```
user@sender_host:~# ifconfig eth0 mtu 9000
```

Permanent changes take effect by modifying network configuration files, dependently from the specific Linux distribution installed².

13.1.5 Disk performance

Before to run any test, it is absolutely required to check performance of the disks subsystem involved. I/O benchmarks like "hdparm", "bonnie++" and "iozone" could be used to test performance of I/O operations on the disk.

13.1.6 Network capacity

Tests will be executed over both public Internet and private PRACE network.

For public Internet the only requirement is that the user end-point is plugged to a network with the following minimum requirements³:

- RTT below 70ms
- 0% of packet lost
- Jitter not above 1ms

For hosts connected to the internal PRACE network, no minimum requirements are set.

13.1.7 Requirements summary

Requirement	Description
TCP Buffer sizing	TCP buffer auto-tuning supported. Maximum Buffer Size adjusted with the BDP.
MTU and Jumbo Frames	Network cards with MTU=9000
Disk performance	I/O performance better than Network performance
Network Capacity for Public Internet	- RTT < 70ms - Packet Loss = 0% - Jitter <= 1ms

Table 8: Requirements List

² <http://www.cyberciti.biz/faq/centos-rhel-redhat-fedora-debian-linux-mtu-size/>

³ User-side requirements can be checked with online free tools like <http://pingtest.net/>

13.1.8 *Methodology*

The proposed methodology must be able to:

- Produce assessments in a consistent manner across different sites and different network paths;
- Consider production conditions and any network turbulence which might occur;
- Assess performance for different types of workloads and different numbers of parallel streams;
- Gather and record results of the evaluation for each technology by using a well-defined template;
- Create a straight forward way to qualify and compare results;
- Provide well defined test-cases;

In addition to a quantitative assessment, also factors like reliability, footprint or intrusiveness, maintenance, code maturity, support, should be considered and qualitatively evaluated.

Tests must be executed on both PRACE network and public Internet.

13.1.9 *Production Conditions*

Before running a test, a report on the network status must be taken. This implies to define at least the Bandwidth Delay Product (BDP), which is calculated multiplying the capacity of the network path (or the narrow link, if any) and the Round-Trip delay Time (RTT):

$$\text{BDP} = (\text{Capacity}) * (\text{RTT})$$

This gives a measure of the network congestion and the ability to compare different file transfer tools under similar values for the BDP.

13.1.10 *Data sets*

Transferring a large number of small files is significantly different from transferring few large files in terms of performance. Also the directory depth or tree affects performance significantly.

In general, a user should be able to optimize the dataset that has to be transferred, e.g. by using archiving, compression and remote synchronization techniques.

Two dataset are defined to take into account these case studies.

- **Dataset A (Many Small files):**
 - Number of files: ≥ 100
 - Size of each file: $\geq 1\text{GB}$
 - Directory tree: ≥ 1 level
- **Dataset B (Few Large files)**
 - Number of files: ≤ 10
 - Size of each file: $\geq 100\text{GB}$
 - Directory tree: = 1 level

13.1.11 *Workload*

There is not a specific study and/or survey figuring out the average amount of data transferred across PRACE sites. Independently from this lack of understanding, it is recommended to test

different size of workloads and to study how tools scale. Taking into account the storage availability for this test, three workloads are considered:

- **Workload A: 100GB**
- **Workload B: 500GB**
- **Workload C: 1000GB (1TB)**

13.1.12 *Parallel Streams*

Only tools that support data transfer parallelism can be considered.

Choosing the number of parallel streams is not a simple task because performance could decrease with high number of streams. It mainly depends from the memory availability at the end points.

Several studies show that 4 to 8 streams are usually sufficient. 16 streams only in case of bad performance found with 4 and 8. Above 16 is basically wasting resources.

So it is recommended to run test with 3 different numbers of streams:

- **Parallel Streams Configuration A: 4**
- **Parallel Streams Configuration B: 8**
- **Parallel Streams Configuration C: 16**

13.1.13 *Qualitative Factors*

It has been considered as valuable to take into account also qualitative factors that are not strictly related to performance of a specific file transfer tool.

Factors like reliability are important for providing a complete feedback whether deciding to include a specific file transfer tool into data services for PRACE.

Evaluation could be provided by using a ranking from 1 (really bad) to 5 (really good) along with a short comment specifying the motivation of the mark.

Recommended factors to be considered are:

- **Reliability**
- **Footprint (Intrusiveness)**
- **Maintenance**
- **Fault Tolerance**
- **Code Maturity**
- **Community Acceptance**

13.1.14 *Test cases*

Fixed a medium, which could be Internet or the private PRACE network, and taking into account of the methodology above mentioned, there will be **18 runs** to execute for each specific tool. The following table shows an example for two specific dataset types (100 files for Dataset A against 1 file for Dataset B).

#Run	DataSet	Workload	Parallel Streams
1	A (100 files of 1GB)	A (100GB)	A (4)
2	A (100 files of 1GB)	A (100GB)	B (8)

3	A (100 files of 1GB)	A (100GB)	C (16)
4	A (100 files of 5GB)	B (500GB)	A (4)
5	A (100 files of 5GB)	B (500GB)	B (8)
6	A (100 files of 5GB)	B (500GB)	C (16)
7	A (100 files of 10GB)	C (1000GB)	A (4)
8	A (100 files of 10GB)	C (1000GB)	B (8)
9	A (100 files of 10GB)	C (1000GB)	C (16)
10	B (1 file of 100GB)	A (100GB)	A (4)
11	B (1 file of 100GB)	A (100GB)	B (8)
12	B (1 file of 100GB)	A (100GB)	C (16)
13	B (1 file of 500GB)	B (500GB)	A (4)
14	B (1 file of 500GB)	B (500GB)	B (8)
15	B (1 file of 500GB)	B (500GB)	C (16)
16	B (1 file of 1TB)	C (1000GB)	A (4)
17	B (1 file of 1TB)	C (1000GB)	B (8)
18	B (1 file of 1TB)	C (1000GB)	C (16)

Table 9: Each test case includes at least 18 runs

13.1.15 Data sheet (template)

Results must be collected by data sheets based on a predefined layout. A data sheet will include quantitative data as well as information about the test bed used. It acts as a data base from which structured information can be further elaborated, e.g. performance with a fixed dataset type and different workloads and parallel streams, performance with a fixed workload and different dataset type and parallel streams, etc...

Information can be presented in table and/or graphic format (recommended).

General Information				
Tool	Site A	Site B	Bidirectional test	
<i>BBCP</i>	<i>CINES</i>	<i>CEA</i>	<i>NO</i>	
Network Status				
Network	Capacity	RTT	BDP	
<i>Public Internet</i>	<i>200Mbps</i>	<i>50ms</i>	<i>1250 KByte</i>	
Hosts / End-Point Status				
Max TCP Buffer Size (Site A)		Max TCP Buffer Size (Site B)		
net.ipv4.tcp_rmem	net.ipv4.tcp_wmem	net.ipv4.tcp_rmem	net.ipv4.tcp_wmem	
<i>4194304</i>	<i>4194304</i>	<i>4194304</i>	<i>4194304</i>	
Quantitative Assessment				
Run#ID	Dataset Type	Workload	Parallel Streams	Throughput (Mbps)

1	A (100 files)	100GB	4	184.75
2	A (100 files)	100GB	8	192.25
3	A (100 files)	100GB	16	193.10
4	A (100 files)	500GB	4	144.07
5	A (100 files)	500GB	8	121.89
6	A (100 files)	500GB	16	166.27
7	A (100 files)	1000GB	4	184.75
8	A (100 files)	1000GB	8	192.25
9	A (100 files)	1000GB	16	193.10
10	B (1 file)	100GB	4	144.07
11	B (1 file)	100GB	8	121.89
12	B (1 file)	100GB	16	166.27
13	B (1 file)	500GB	4	184.75
14	B (1 file)	500GB	8	192.25
15	B (1 file)	500GB	16	193.10
16	B (1 file)	1000GB	4	144.07
17	B (1 file)	1000GB	8	121.89
18	B (1 file)	1000GB	16	166.27
Qualitative Assessment				
Factor	Rank (1 – 5)	Comment		
Reliability	4	No crashes reported during the tests. None reported on the web.		
Footprint Intrusiveness	5	Minimal. It doesn't require administrative rights or system servers, e.g. can be installed by user.		
Maintenance	5	No maintenance required by system administrators.		
Fault Tolerance	1	Bad, the tool doesn't provide failure restart capabilities.		
Code Maturity	3	Medium, first version released in 2011		
Community Acceptance	4	Medium. Number of users is growing as reported by PRACE partners for their local scientific communities.		

Table 10: Data sheet template filled in with sample data

13.2 Detailed Ceph Testing methodology and Results

The following file system tests are used to model real life usage scenarios as closely as possible:

- Read write test by uncompressing the linux kernel tree
- Delete the uncompressed files
- Unbuffered stream test with dd
- IOzone tests

Block storage images are created with the following scheme:

```

rbd -c ceph.conf --keyring ceph.client.admin.keyring --id admin create
IMAGE --size SIZE
rbd -c ceph.conf --keyring ceph.client.admin.keyring --id admin map
IMAGE
mkfs.ext4 -m0 -v /dev/rbd/rbd/IMAGE
mkdir /mnt/IMAGE
mount /dev/rbd/rbd/IMAGE /mnt/IMAGE

```

13.2.1 Read and write test by uncompressing the Linux kernel tree

The process produced the following results:

```

ceph# time tar -Jxf linux-3.15-rc3.tar.xz
real    1m54.501s
user    0m13.921s
sys     0m7.291s

local# time tar -Jxf linux-3.15-rc3.tar.xz
real    0m57.919s
user    0m14.395s
sys     0m6.959s

```

The creation and deletion tests were looped and ran for a day on each client simultaneously to stress test the cluster (see graph time period from 07 to 08). The wait load was high and the overall write performance was around 35MB/s which is considered as reasonably good result since the client cluster was connected with a single 1Gbit/s Ethernet-connection. The large wait and system load is due to the cloud and network overhead.

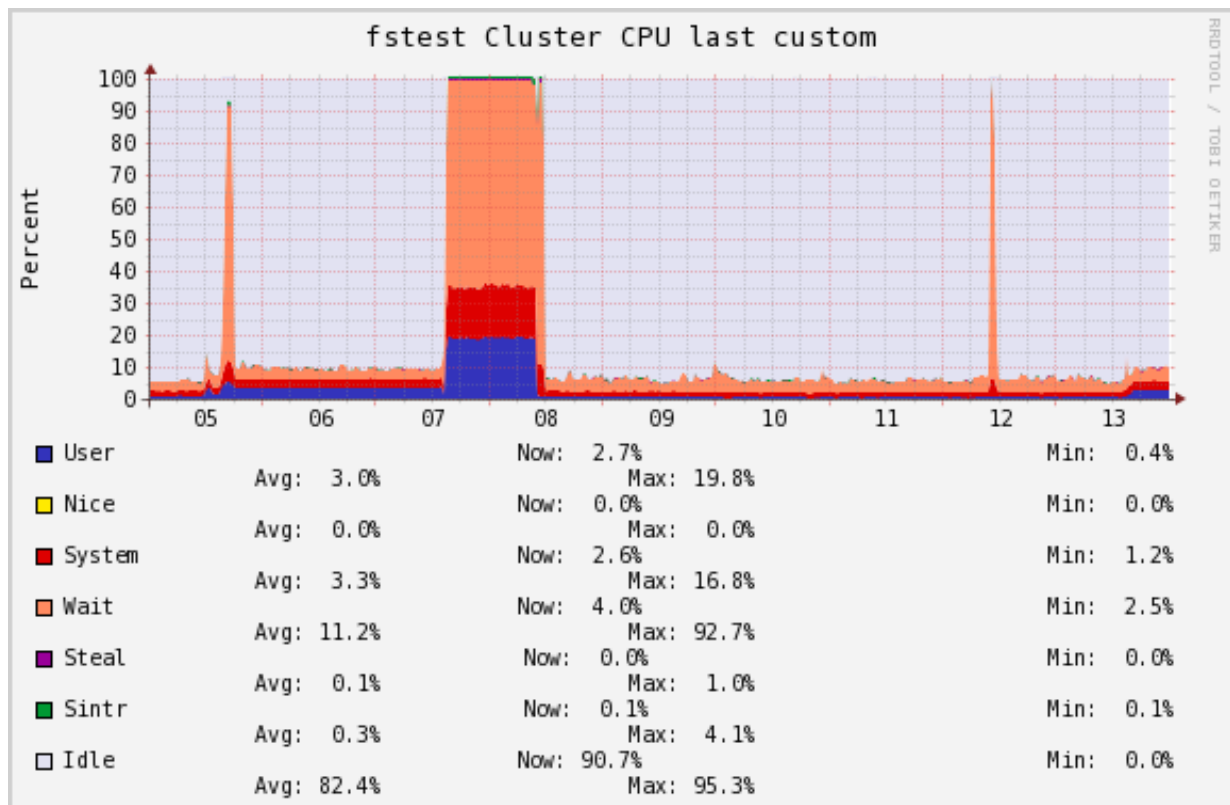


Figure 10: NIF cluster CPU load

13.2.2 Unbuffered stream tests with dd

Stream write tests were done with unbuffered dd write. The first peak in the Figure above shows the dd test running. Every test was repeated 10 times to cancel out noise related errors. Here we show typical run examples with numbers around the average. First, the remote and local performance has been compared:

```
ceph# dd if=/dev/zero of=speetest bs=1M count=1000 conv=fdatasync
1048576000 bytes (1.0 GB) copied, 181.776 s, 5.8 MB/s

local# dd if=/dev/zero of=speetest bs=1M count=1000 conv=fdatasync
1048576000 bytes (1.0 GB) copied, 70.9067 s, 14.8 MB/s
```

Stream tests were in average 2 times slower than the local one. Large file test were done by writing a 90GB files on each client simultaneously. The average speed was around 15 MB/s. Here we show one typical test result:

```
# dd if=/dev/zero of=speetest bs=1M count=90000 conv=fdatasync
94371840000 bytes (94 GB) copied, 6160.74 s, 15.3 MB/s
```

13.2.3 IOzone tests

Finally, IOzone test were done to cover various block and file sizes and to get an overall picture of the block storage performance. The graphs show the IOZone tests run on a single node. Due to the cache effect the performance can be larger than the network speed. However, at large file sizes, where the IO buffer of the system is full, the speed drops under 30MB/s.

The second peak on the CPU load picture shows the IOzone test runs.

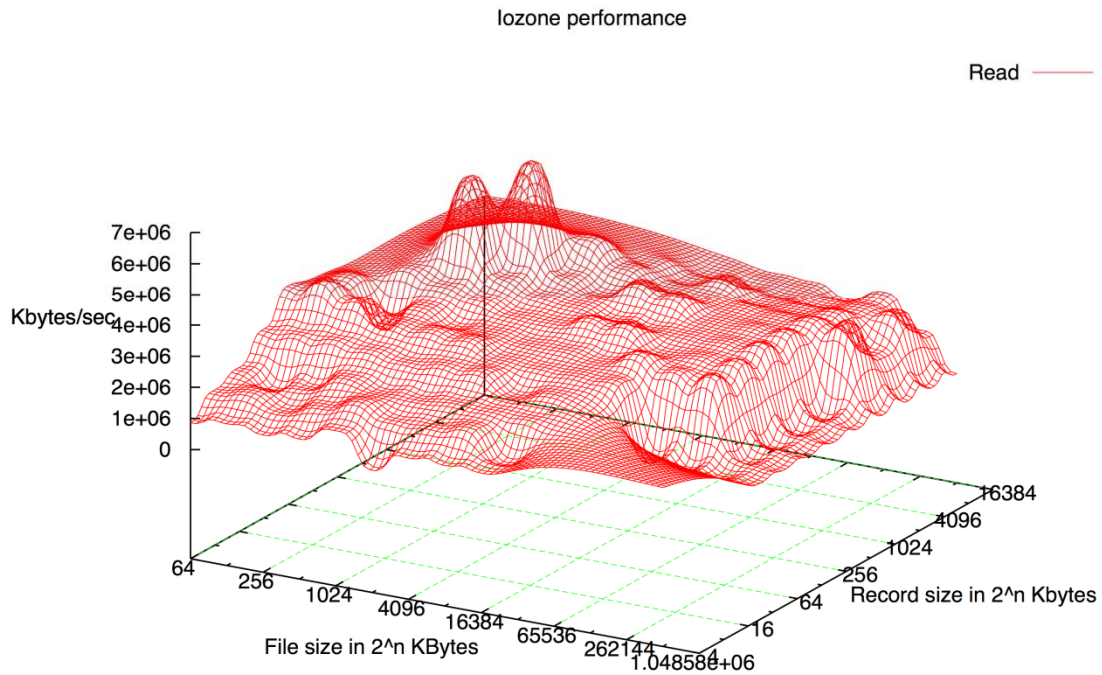


Figure 11: IOzone Read test results

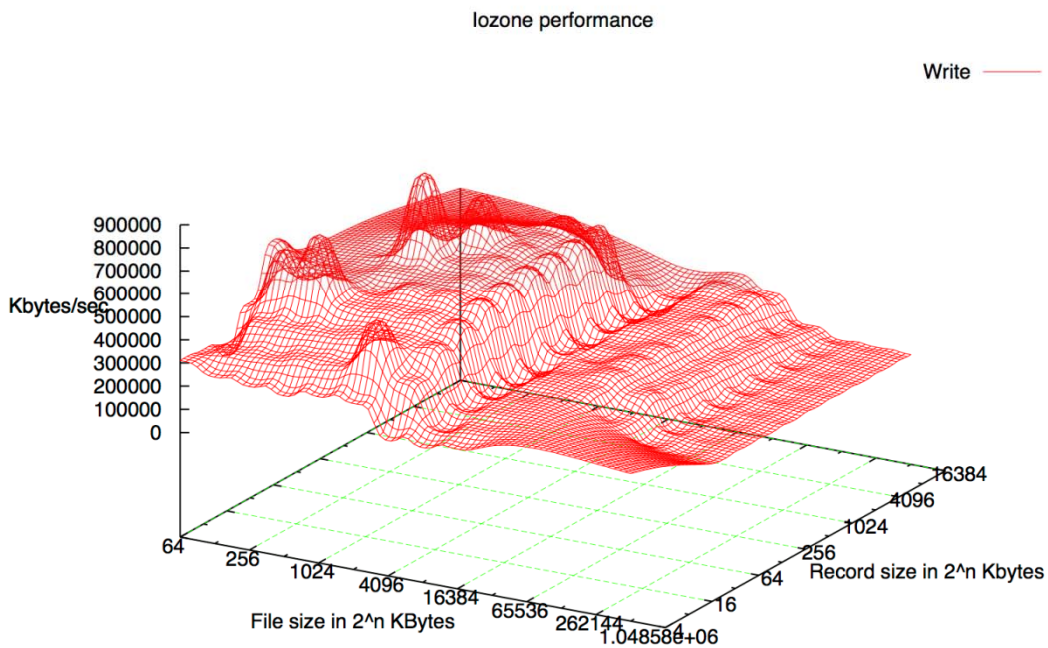


Figure 12: IOzone Write test results

Replica size on the block storage (rbd) subsystem is set to 3, which implied that the overall server network input is 3 times the overall output of the clients and the overall server network output is 2 times the overall output of the clients. The optimal block size is peaked at around 4K. Write operations have a slow down at the same block size as well.

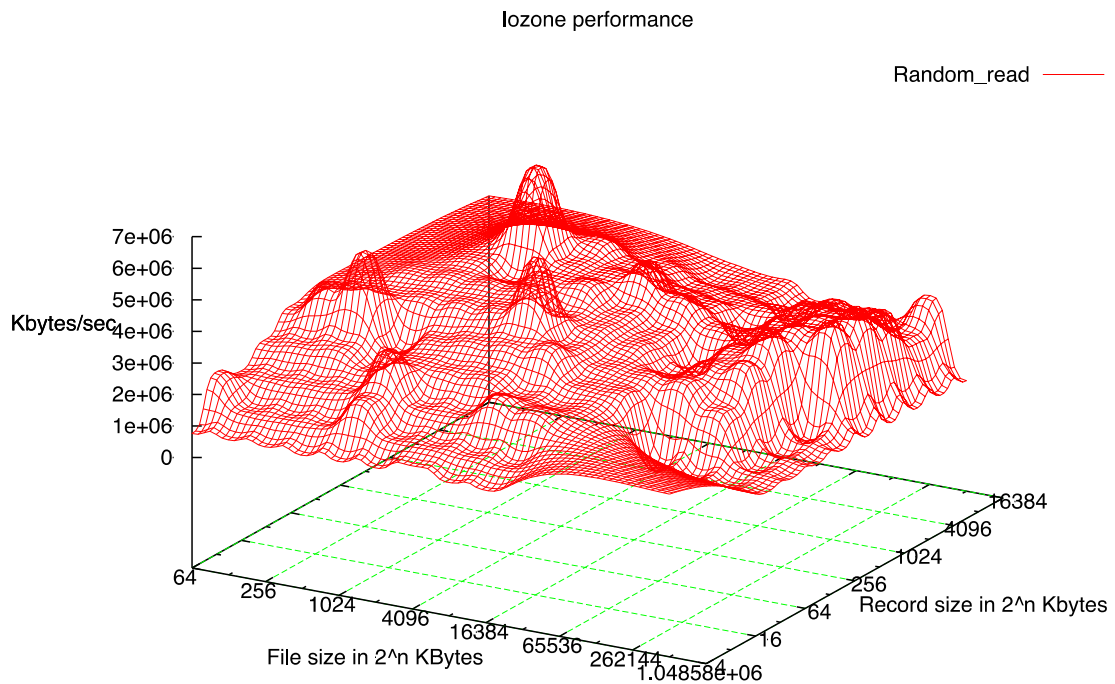


Figure 13: Random read tests with IOzone

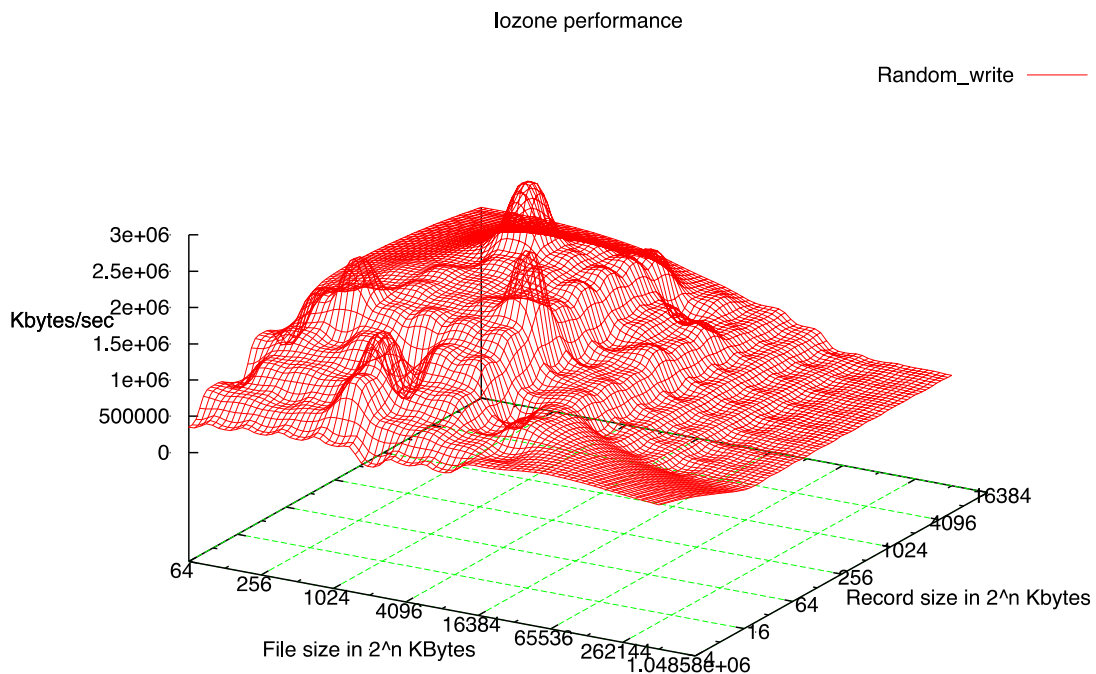


Figure 14: Random write tests with IOzone

The bottleneck for the network traffic is the client cluster uplink which provides around 100-120MB/s. A sustained 35MB/s write speed with high wait and system load on the clients has been measured. In the Figure below the Network IO performance of the client cluster is shown. The first peak is the dd test, the large plateau is the file stress test and last peak is the

IOzone tests. The theoretical maximal speed of the 1Gbit/s Ethernet-connection is around 125MB/s and the test reached 20% of it.

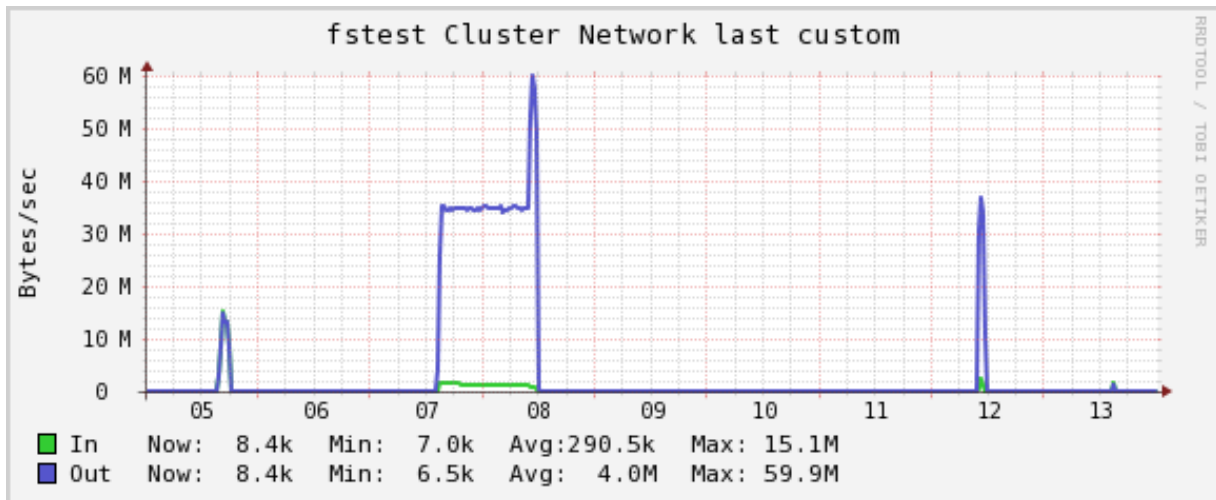


Figure 15: NIIF cluster Network I/O

13.2.4 Server side load

On the server side (octopus cluster) the overall CPU load was low and the network load was at least twice the client output (Wednesday to Thursday on the chart).

Inspecting SURFsara cluster load graphs, the following conclusions can be made:

- Ceph equally divides the workload over the nodes.
- Cpu load is barely influenced. The peak usage was just 4%.
- Cluster network load shows ~70 MB/s with a peak to ~150 MB/s.
- Looking at the per node graphs, it shows ~12 MB/s per node. In total, this amounts to roughly 1 Gbps, the NIIF - SURFsara connection maximum.
- Network load was the same for each node (even for monitors)
- Memory cached graphs show that all nodes cache data.

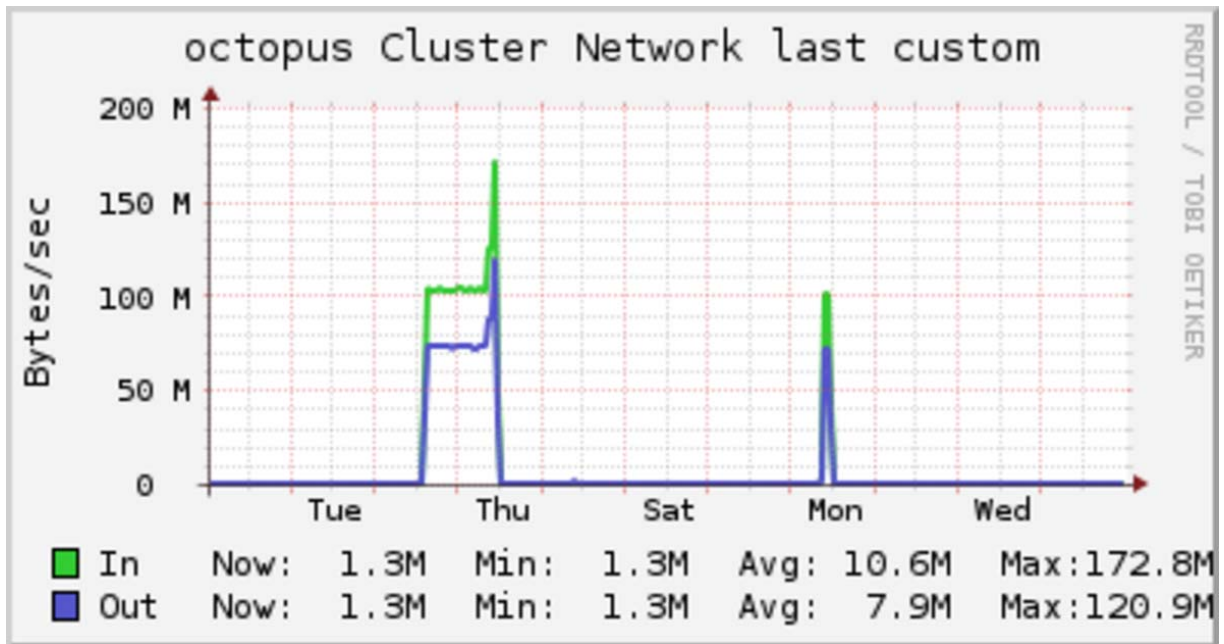


Figure 16: SURFsara cluster Network I/O

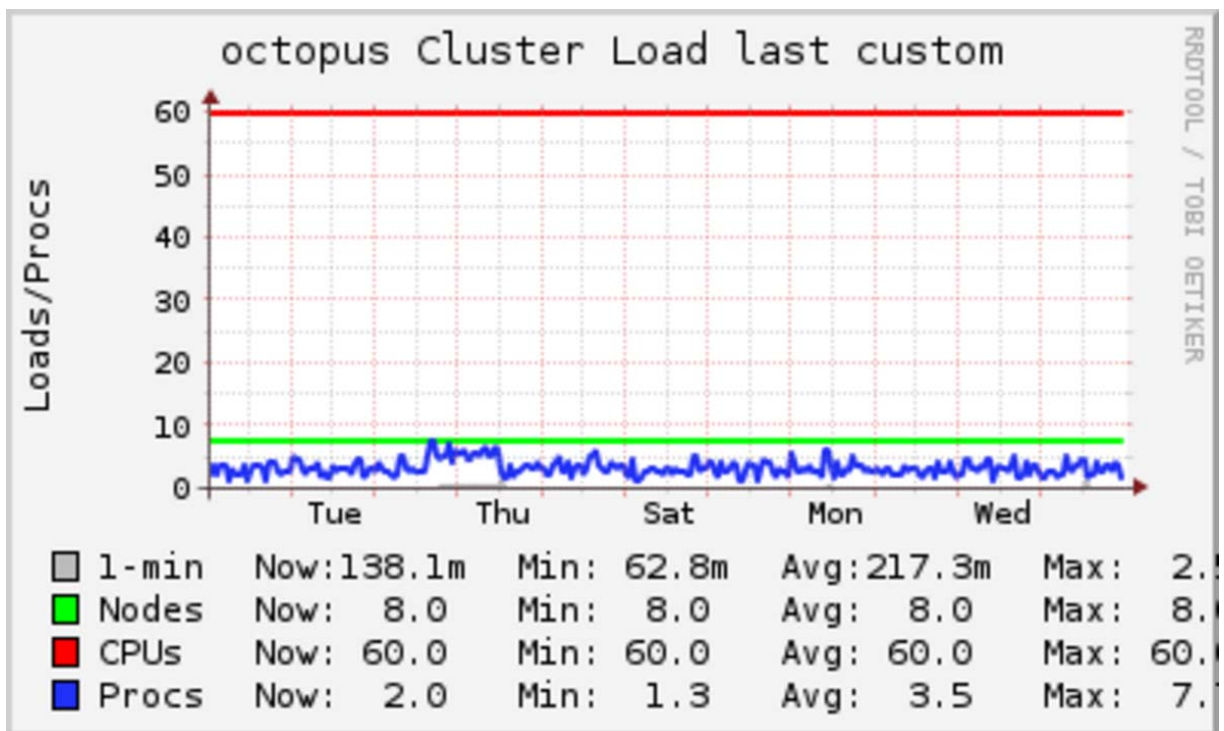


Figure 17: SURFsara cluster CPU Load

13.3 iRODS user documentation and best practices guide

This section provides a best practices guide for the iRODS 3.3 usage in PRACE.

13.3.1 *Getting an iRODS account*

Access to the PRACE data servers is restricted. If you plan to use iRODS, you need to contact your PRACE support.

13.3.2 *Accessible data servers*

The PRACE infrastructure provides a set of data servers for medium or long term storage, for sharing and archiving data sets. These data servers have restricted access and depend on your project. Please contact the PRACE support to get the list of the data servers.

There are several ways to access, store and retrieve data since several iRODS clients exist (icommands, iDROP, iRODS web client...). icommands are the most commonly used client, so their usage will be described here. icommands are command-line only (similar to Unix ones) and are specially well adapted to large and bulk file transfers.

13.3.3 *Initial iRODS setup*

The standard iRODS *icommands* are available on the PRACE HPC system and can be started via the “module” command.

Load the iRODS environment with the module command:

```
module load prace; module load globus; module load irods
```

The iRODS deployment in PRACE supports GSI as the standard authentication mechanism. To get a proxy certificate, run the standard globus command:

```
grid-proxy-init
```

once your usercert.pem and userkey.pem files are properly set.

13.3.4 *Configuring the connection to a data server*

Create the .irods subdirectory:

```
mkdir /$HOME/.irods
```

Configure the iRODS client to connect to the appropriate data server and to use GSI authentication. So, create the file /\$HOME/.irods/.irodsEnv similar to:

```
>cat .irodsEnv
```

```
irodsHost turing2-d.idris.fr
```

irodsPort 1247
 irodsUserName pr1f02is
 irodsZone IDRIS
 irodsAuthScheme=GSI

The Variables are as follow:

Variable	
irodsHost	iRODS server name
irodsPort	iRODS server port
irodsUserName	iRODS user account
irodsZone	iRODS zone you connect to
irodsAuthScheme	Authentication method

The ienv command returns the following:

```
> ienv
```

```
NOTICE: Release Version = rods3.3, API Version = d
NOTICE: irodsHost=turing2-d.idris.fr
NOTICE: irodsPort=1247
NOTICE: irodsUserName=pr1f02is
NOTICE: irodsZone=IDRIS
NOTICE: irodsAuthScheme=GSI
NOTICE: environment variable set, irodsAuthScheme=GSI
NOTICE: created irodsHome=/IDRIS/home/pr1f02is
NOTICE: created irodsCwd=/IDRIS/home/pr1f02is
```

13.3.5 Accessing your data, data storage and retrieval

The two basic iRODS commands are:

- `iput` that stores a file to an iRODS server
- `iget` that gets a file from an iRODS server

And the most commonly used icommands are:

icommand	
<code>Iput</code>	store a file into iRODS
<code>Iget</code>	get a file from iRODS
<code>Ipwd</code>	print the iRODS current directory
<code>Icd</code>	change the iRODS current directory
<code>Ils</code>	list iRODS data objects (files) and collections (directories)
<code>Imkdir</code>	make an iRODS directory (collection)
<code>ichmod</code>	Change access permissions to data objects or collections
<code>Irm</code>	Remove data objects or collections
<code>icp</code>	copy a data object or a collection to another one
<code>Imv</code>	move/rename a data object or a collection
<code>Irepl</code>	Replicate data objects.
<code>Ichksum</code>	Checksum one or more data-object or

	collection from iRODS space
Irsync	Synchronize the data between a local copy and the copy stored in iRODS or between two iRODS copies
Ibun	Upload and download structured (e.g. tar) files
igetwild.sh	Get one or more iRODS files using wildcard characters

Environmental icommands:

icommand	
Ienv	Show current iRODS environment
Ilsresc	List resources
Iuserinfo	List users
Imiscsvrinfo	Get basic server information
Ihelp	List the i-commands and optionally an i-command's help text
Ilocate	Search for data-object(s) OR collections (via a script)
Iquota	Show information on iRODS quotas (if any)

13.3.6 Storing data to data server

Users might need to store data files created in his working directory \$WORKDIR after a computational phase. To perform this operation run the following commands:

```
[pr1f02is@turing2: pr1f02is]$ cd $WORKDIR/simul-04
```

List the content of the iRODS data server directory:

```
[pr1f02is@turing2: pr1f02is]$ ls
result.dat
```

Print the current directory in the data server environment:

```
[pr1f02is@turing2: pr1f02is]$ ipwd
/IDRIS/home/pr1f02is
```

Create a new directory in the data server environment. It will be used to store the data file.

```
[pr1f02is@turing2: pr1f02is]$ mkdir eo-simul-2014-0430
```

Change the current data server directory to the new *es-simul-2014-0430* directory:

```
[pr1f02is@turing2: pr1f02is]$ icd eo-simul-2014-0430
```

Run the `iput` command that will ingest the data file in the data server in the given directory:

```
[pr1f02is@turing2: pr1f02is]$ iput result.dat
```

Check that the file has been properly ingested in the data server:

```
[pr1f02is@turing2: pr1f02is]$ ils -l
/IDRIS/home/pr1f02is/eo-simul-2014-0430:
pr1f02is      0 demoResc      8099785472 2014-05-05.14:01 & result.dat
```

13.3.7 Retrieving data from a data server

We will assume here that a user may need to retrieve data files from a data server in order to use it during a computational phase that will be run from his working directory `$WORKDIR`.

To perform this operation run the following commands:

```
[pr1f02is@turing2: pr1f02is]$ cd $WORKDIR/simul-04
```

Print the current directory in the data server environment:

```
[pr1f02is@turing2: pr1f02is]$ ipwd
/IDRIS/home/pr1f02is
```

Change the current data server directory to the `es-simul-2014-0430` directory:

```
[pr1f02is@turing2: pr1f02is]$ icd eo-simul-2014-0430
```

Run the `iget` command that will retrieve the data file from the data server and will write it in the current directory:

```
[pr1f02is@turing2: pr1f02is]$ iget result.dat
```

Check that the file is here:

```
[pr1f02is@turing2: pr1f02is]$ ls -lrt
total 2335360
-rw-r----- 1 pr1f02is pr1f00is 8099785472 Apr 30 14:30 result.dat
```

13.3.8 Managing archive files

icommands include bundle file operations using the `ibun` command. This command allows structured files such as tar files to be uploaded and downloaded to/from iRODS. For example, for unpacking a tar archive file into iRODS, run the following commands:

```
[pr1f02is@turing2: pr1f02is]$ ipwd
/IDRIS/home/pr1f02is
```

```
[pr1f02is@turing2: pr1f02is]$ ls
result result.dat result.tar
```

result.tar is the archive file and *simul* is a new directory into which the tar file will be unpacked (note that the *simul* collection doesn't need to be created in advance)

```
[pr1f02is@turing2: pr1f02is]$ ibun -x result.tar simul
```

Check that the *simul* collection has been created in the data server environment:

```
[pr1f02is@turing2: pr1f02is]$ ils -l simul
/IDRIS/home/pr1f02is/simul:
C- /IDRIS/home/pr1f02is/simul/result
```

Check that the archive file has been properly unpacked into the data server environment:

```
[pr1f02is@turing2: pr1f02is]$ ils -l simul/result
/IDRIS/home/pr1f02is/simul/result:
pr1f02is      0 demoResc      8099785472 2014-05-05.15:19  result.dat
```

The following command can be used to tar/bundle an iRODS collection into a tar file:

```
[pr1f02is@turing2: pr1f02is]$ ibun -cDtar eo-simul-2014-0516.tar eo-simul-2014-0516
```

Note that the '-cDtar' option specifies that the collection is bundled into a tar file.

The following collection is considered:

```
[pr1f02is@turing2: Perf-pr1f02is]$ ils -l /IDRIS/home/pr1f02is/eo-simul-2014-0516
/IDRIS/home/pr1f02is/eo-simul-2014-0516:
pr1f02is      0 demoResc      8439 2014-05-16.15:26 & result10.dat
pr1f02is      0 demoResc      24553 2014-05-16.15:26 & result11.dat
pr1f02is      0 demoResc      24553 2014-05-16.15:26 & result12.dat
pr1f02is      0 demoResc     1123219 2014-05-16.15:26 & result13.dat
pr1f02is      0 demoResc     130226 2014-05-16.15:26 & result14.dat
pr1f02is      0 demoResc     122496 2014-05-16.15:26 & result15.dat
pr1f02is      0 demoResc     122496 2014-05-16.15:26 & result16.dat
pr1f02is      0 demoResc     100592 2014-05-16.15:26 & result17.dat
pr1f02is      0 demoResc     100376 2014-05-16.15:26 & result18.dat
pr1f02is      0 demoResc     101245 2014-05-16.15:26 & result19.dat
```

```

pr1f02is      0 demoResc      162 2014-05-16.15:26 & result1.dat
pr1f02is      0 demoResc     121528 2014-05-16.15:26 & result20.dat
pr1f02is      0 demoResc     121528 2014-05-16.15:26 & result21.dat
pr1f02is      0 demoResc     82864 2014-05-16.15:26 & result22.dat
pr1f02is      0 demoResc     121598 2014-05-16.15:26 & result23.dat
pr1f02is      0 demoResc     1918823 2014-05-16.15:26 & result24.dat
pr1f02is      0 demoResc      162 2014-05-16.15:26 & result2.dat
pr1f02is      0 demoResc      162 2014-05-16.15:26 & result4.dat
pr1f02is      0 demoResc     3261 2014-05-16.15:26 & result5.dat
pr1f02is      0 demoResc     5917 2014-05-16.15:26 & result6.dat
pr1f02is      0 demoResc     16677 2014-05-16.15:26 & result7.dat
pr1f02is      0 demoResc     39269 2014-05-16.15:26 & result8.dat
pr1f02is      0 demoResc      400 2014-05-16.15:26 & result9.dat
C- /IDRIS/home/pr1f02is/eo-simul-2014-0516/result34.dat

```

Check that the tar file has been created properly following theibun command described above:

```

[pr1f02is@turing2: Perf-pr1f02is]$ ils -l
/IDRIS/home/pr1f02is:
pr1f02is      0 demoResc      0 2014-04-07.16:50 & 1MB_00_R
pr1f02is      0 demoResc     1048576 2014-04-16.16:46 & 1MB_18-f
pr1f02is      0 demoResc     1048576 2014-05-06.10:29 & 1MB_18-f1
pr1f02is      0 demoResc     1048576 2014-03-28.11:06 & 1MB_18-f-2
pr1f02is      0 demoResc     8099788800 2014-05-06.11:11 & eo-simul-2014-
0430.tar
pr1f02is      0 demoResc     4485120 2014-05-16.15:27 & eo-simul-2014-0516.tar
C- /IDRIS/home/pr1f02is/eo-simul-2014-0430
C- /IDRIS/home/pr1f02is/eo-simul-2014-0516
C- /IDRIS/home/pr1f02is/shared-simul
C- /IDRIS/home/pr1f02is/simul

```

Get the tar file:

```

[pr1f02is@turing2: pr1f02is]$ iget eo-simul-2014-0516.tar

```

Check the tar file and extract the information:

```

[pr1f02is@turing2: pr1f02is]$ ls -lrt
total 10249856
-rw-r----- 1 pr1f02is pr1f00is 8099785472 Apr 30 14:30 result.dat
drwxr-x--- 2 pr1f02is pr1f00is 512 May 5 15:13 result
-rw-r----- 1 pr1f02is pr1f00is 8099788800 May 5 15:14 result.tar
-rw-r----- 1 pr1f02is pr1f00is 4485120 May 16 15:27 eo-simul-2014-0516.tar

[pr1f02is@turing2: pr1f02is]$ tar xvf eo-simul-2014-0516.tar

```

```
[pr1f02is@turing2: pr1f02is]$ ls -l eo-simul-2014-0516
total 5376
-rw----- 1 pr1f02is pr1f00is  8439 May 16 15:26 result10.dat
-rw----- 1 pr1f02is pr1f00is 24553 May 16 15:26 result11.dat
-rw----- 1 pr1f02is pr1f00is 24553 May 16 15:26 result12.dat
-rw----- 1 pr1f02is pr1f00is 1123219 May 16 15:26 result13.dat
-rwx----- 1 pr1f02is pr1f00is 130226 May 16 15:26 result14.dat
-rwx----- 1 pr1f02is pr1f00is 122496 May 16 15:26 result15.dat
-rwx----- 1 pr1f02is pr1f00is 122496 May 16 15:26 result16.dat
-rwx----- 1 pr1f02is pr1f00is 100592 May 16 15:26 result17.dat
-rwx----- 1 pr1f02is pr1f00is 100376 May 16 15:26 result18.dat
-rw----- 1 pr1f02is pr1f00is 101245 May 16 15:26 result19.dat
-rw----- 1 pr1f02is pr1f00is   162 May 16 15:26 result1.dat
-rwx----- 1 pr1f02is pr1f00is 121528 May 16 15:26 result20.dat
-rwx----- 1 pr1f02is pr1f00is 121528 May 16 15:26 result21.dat
-rwx----- 1 pr1f02is pr1f00is  82864 May 16 15:26 result22.dat
-rwx----- 1 pr1f02is pr1f00is 121598 May 16 15:26 result23.dat
-rw----- 1 pr1f02is pr1f00is 1918823 May 16 15:26 result24.dat
-rw----- 1 pr1f02is pr1f00is   162 May 16 15:26 result2.dat
drwx----- 2 pr1f02is pr1f00is   512 May 16 15:27 result34.dat
-rw----- 1 pr1f02is pr1f00is   162 May 16 15:26 result4.dat
-rw----- 1 pr1f02is pr1f00is  3261 May 16 15:26 result5.dat
-rw----- 1 pr1f02is pr1f00is  5917 May 16 15:26 result6.dat
-rw----- 1 pr1f02is pr1f00is 16677 May 16 15:26 result7.dat
-rwx----- 1 pr1f02is pr1f00is 39269 May 16 15:26 result8.dat
-rw----- 1 pr1f02is pr1f00is   400 May 16 15:26 result9.dat
```

13.3.9 Sharing data

The icommands allow also to precise the access over your data. You can keep your data private, share them publicly or share them with a limited set of persons by setting permissions and creating shared directories.

Note that read access on directories is set by default.

icommand	
ils -Ar	List contents of iRODS collections and all associated permissions recursively
ichmod own	Grant full ownership permission level for specified user to selected data object (file) or collection
ichmod read	Grant read-only permission level for specified user to selected data object (file) or collection
ichmod write	Grant read and write permission level for specified user to selected data object (file) or collection
ichmod null	Remove permission level for the user to the (data object) file or collection

```
[pr1f02is@turing2: pr1f02is]$ ils -Ar
/IDRIS/home/pr1f02is:
  ACL - pr1f02is#IDRIS:own
  Inheritance - Disabled
1MB_00_R
  ACL - pr1f02is#IDRIS:own
1MB_18-f
  ACL - pr1f02is#IDRIS:own
1MB_18-f-2
  ACL - pr1f02is#IDRIS:own
result.tar
  ACL - pr1f02is#IDRIS:own
C- /IDRIS/home/pr1f02is/eo-simul-2014-0430
/IDRIS/home/pr1f02is/eo-simul-2014-0430:
  ACL - pr1f02is#IDRIS:own
  Inheritance - Disabled
result.dat
  ACL - pr1f02is#IDRIS:own
C- /IDRIS/home/pr1f02is/simul
/IDRIS/home/pr1f02is/simul:
  ACL - pr1f02is#IDRIS:own
  Inheritance - Disabled
C- /IDRIS/home/pr1f02is/simul/result
/IDRIS/home/pr1f02is/simul/result:
  ACL - pr1f02is#IDRIS:own
  Inheritance - Disabled
result.dat
  ACL - pr1f02is#IDRIS:own
```

To allow write data sharing, run the following commands:

```
[pr1f02is@turing2: ~]$ ils -A /IDRIS/home/pr1f02is/shared-simul
/IDRIS/home/pr1f02is/shared-simul:
  ACL - pr1f02is#IDRIS:own
  Inheritance - Disabled
```

Set the write access permissions to the given directory:

```
[pr1f02is@turing2: ~]$ ichmod -r write praceuser shared-simul
```

```
[pr1f02is@turing2: ~]$ ils -A /IDRIS/home/pr1f02is/shared-simul
/IDRIS/home/pr1f02is/shared-simul:
  ACL - pr1f02is#IDRIS:own praceuser#IDRIS:modify object
  Inheritance - Disabled
```

Note that when collections have the inheritance attribute set, new dataObjects and collections added to the collection inherit the access permissions (ACLs) of the collection.

13.3.10 *Synchronizing data*

iRODS offers different icommands to synchronize data. The *irsync* command allows data to be synchronized between the local file system environment and iRODS or within the iRODS environment itself.

To synchronize data between your local environment and the iRODS environment run the following commands:

```
[pr1f02is@turing2: eo-simul-2014-0430]$ ls -lrt
total 5479424
-rw-r----- 1 pr1f02is pr1f00is 7762345984 May  7 10:14 result.dat
```

```
[pr1f02is@turing2: eo-simul-2014-0430]$ ils -l /IDRIS/home/pr1f02is/eo-simul-2014-0430
/IDRIS/home/pr1f02is/eo-simul-2014-0430:
pr1f02is      0 demoResc      8099785472 2014-05-05.14:01 & result.dat
```

Run the synchronization command:

```
[pr1f02is@turing2: ~]$ irsync -r eo-simul-2014-0430 i:/IDRIS/home/pr1f02is/eo-simul-2014-0430
```

The prefix 'i:' is used to distinguish an iRODS collection path from a local file system path.

Check that the synchronization has been performed properly:

```
[pr1f02is@turing2: ~]$ ils -l /IDRIS/home/pr1f02is/eo-simul-2014-0430
/IDRIS/home/pr1f02is/eo-simul-2014-0430:
pr1f02is      0 demoResc      7762345984 2014-05-07.10:17 & result.dat
```

13.3.11 *Adding metadata and searching*

Metadata is information attached to the data. Metadata is represented by an AVU triplet (attribute-value-units) in the system. This triplet consists of an Attribute-Name, Attribute-Value, and an optional Attribute-Units.

icommand	
imeta	add, remove, list, or query user-defined Attribute-Value-Unit triplets metadata

iRODS metadata (*imeta* command) includes user-defined and iRODS system attributes stored in the iCAT database related to a data object, collection, resource, or user etc.

You can use the *imeta* command to add metadata to:

- a. Data object (irods files)
- b. Collections

For example, to add metadata to a collection, run the following command:

```
[pr1f02is@turing2: ~]$ imeta add -C /IDRIS/home/pr1f02is/eo-simul-2014-0430
resolution 00-01-35.45
```

To check that metadata has been attached to the given collection, run:

```
[pr1f02is@turing2: ~]$ imeta ls -C /IDRIS/home/pr1f02is/eo-simul-2014-0430
AVUs defined for collection /IDRIS/home/pr1f02is/eo-simul-2014-0430:
attribute: resolution
value: 00-01-35.45
units:
```

The *imeta* command allows to query, in the following way:

```
[pr1f02is@turing2: ~]$ imeta qu -C resolution = 00-01-35.45
collection: /IDRIS/home/pr1f02is/eo-simul-2014-0430
```

Finally, metadata can be removed:

```
[pr1f02is@turing2: ~]$ imeta rm -C /IDRIS/home/pr1f02is/eo-simul-2014-0430
resolution 00-01-35.45
```

```
[pr1f02is@turing2: ~]$ imeta ls -C /IDRIS/home/pr1f02is/eo-simul-2014-0430
AVUs defined for collection /IDRIS/home/pr1f02is/eo-simul-2014-0430:
None
```

13.3.12 Accessing a EUDAT storage

It is possible to store or retrieve data to/from the EUDAT infrastructure from the PRACE infrastructure.

As EUDAT doesn't authorize users to use the standard iRODS icommands to access their storages from an external infrastructure, gridFTP clients such as *globus-url-copy* have to be used instead. Alternatively, a *gtransfer* client can also be used for an additional EUDAT PID support that is not provided with *globus-url-copy*.

Note: the gridFTP clients usage provides a data transfer service only and not to a full data management service like the icommands.

The following parts are examples, please contact your PRACE support to check which EUDAT data servers you can use.

To store a file into a EUDAT storage:


```
[pr1f02is@turing2:      pr1f02is]$      globus-url-copy      gsiftp://turing2-
d.idris.fr:1249/IDRIS/home/pr1f02is/1MB_00_R      gsiftp://jade-
prace.cines.fr:2813/CINES/home/pr1f02is%23IDRIS/1MB_00_R
```

Check that the file has been transferred properly:

```
[pr1f02is@turing2:      pr1f02is]$      globus-url-copy      -list      gsiftp://jade-
prace.cines.fr:2813/CINES/home/pr1f02is%23IDRIS/
gsiftp://jade-prace.cines.fr:2813/CINES/home/pr1f02is%23IDRIS/
1MB_0
1MB_000
1MB_00_R
1MB_10
1MB_18-f1
500GB-1/
```

To get a file from a EUDAT storage run the following command:

```
[pr1f02is@turing2:      pr1f02is]$      globus-url-copy      gsiftp://jade-
prace.cines.fr:2813/CINES/home/pr1f02is%23IDRIS/1MB_18-f1      gsiftp://turing2-
d.idris.fr:1249/IDRIS/home/pr1f02is/1MB_18-f1
```

Check that the file has been transferred properly:

```
[pr1f02is@turing2:      pr1f02is]$      globus-url-copy      -list      gsiftp://turing2-
d.idris.fr:1249/IDRIS/home/pr1f02is/
gsiftp://turing2-d.idris.fr:1249/IDRIS/home/pr1f02is/
1MB_00_R
1MB_18-f
1MB_18-f1
1MB_18-f-2
result.tar
eo-simul-2014-0430/
simul/
```

13.3.13 *Data servers policy and accessibility*

Data servers are managed under different policy and accessibility rules, so they may be accessible depending on your data management requirements. Please contact the PRACE support to know which data servers you are allowed to use.

13.3.14 *Data allocations*

Data allocations are defined during the project acceptance phase. You have to contact the PRACE support to know your initial data allocation and for any request for additional allocation.

Alternatively, you can check if iRODS quotas has been set by running the following command :

```
[pr1f02is@turing2: ~]$ iquota -a
Resource quotas for users:
None

Global (total) quotas for users:
None

Group quotas on resources:
None

Group global (total) quotas:
None
```

13.3.15 *Data organization into iRODS collections*

iRODS manages collections which are similar to directories. Collections help to organize the files in a logical way, so a hierarchical collection tree has to be created first to reflect the logical data organization. As an example, a collection can represent a data set which gathers data captured during an observation phase that can last several days.

13.3.16 *Data management rules*

You can be requested by the data management administrator to bundle files in a collection into a number of tar files to make it more efficient to store the files on a background HSM.

Indeed, in most of the cases when the number of small files (a few megabytes per file) tends to increase, it is recommended to merge them into larger compressed archive units.

13.3.17 *Data transfer tips*

The icommands can be used to transfer small and bigger files i.e larger than 10 GB. If you need to manage many small files at once, the *iput* bulk option (-b) has be used preferably.

If you are facing reliability issues while transferring your data, try the following command options to ingest data into iRODS (*iput* command) or retrieve data from iRODS (*iget* command):

- T renews the socket connection every 10 minutes
- X <checkpoint-file> saves the progress to a file, so that if you restart a failed attempt it sends only the files that were not sent successfully.
- lfrestart <checkpoint-lf-file> saves the progress to individual file, so that if you restart a failed attempt it sends the portion of that file.

13.3.18 *File names*

Do not use file names with special characters.