



034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds



Deliverable reference number D6.5

Application and middleware testing report

Due date of deliverable: 2008-03-31
Actual submission date: 2008-03-31
Document code: <Phosphorus-WP6-D6.5>

Start date of project:
October 1, 2006

Duration:
30 Months

Organisation name of lead contractor for this deliverable:
Forschungszentrum Jülich GmbH

Revision 1.0

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission)	X
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission Services)	



Application and middleware testing report

Authors

Karl Catewicz	FhG
Stephan Springstubbe	FhG
Wolfgang Ziegler	FhG



Application and middleware testing report

Abstract

This document presents deliverable D6.5, the application and middleware testing report. It describes the tests of middleware and applications and the results of these tests conducted within the PHOSPHORUS testbed. The testbed itself is described in D6.1 and D6.2.



Table of Contents

0	Executive Summary	7
1	Introduction	8
2	Distributed Collaborative Visualization (KoDaVis)	9
2.1	Description	9
2.2	Experiences and results	10
2.2.1	PSNC	10
2.2.2	UESSEX	11
2.2.3	VIOLA	11
2.3	Summary	11
3	Distributed Data Storage Systems (DDSS)	12
3.1	Description	12
3.2	Experiences and results	14
3.2.1	Status of implementation and deployment	14
3.2.2	PSNC	16
3.2.3	UESSEX	16
3.2.4	VIOLA	17
3.3	Summary	17
4	WISDOM	18
4.1	Description	18
4.2	Experiences and results	19
4.2.1	PSNC	20
4.2.2	VIOLA	20
4.2.3	UESSEX	21
4.3	Summary	22
5	TOPS	23
5.1	Description	23
5.2	Experiences and results	25



Application and middleware testing report

5.2.1	SARA and FHG	25
5.3	Summary	27
6	INCA	28
6.1	Description	28
6.2	Experiences and results	29
6.2.1	PSNC	29
6.2.2	CESNET	30
6.3	Summary	30
7	Sites running UNICORE 6	31
7.1	Description	31
7.2	Experiences and results	31
7.2.1	PSNC	31
7.2.2	UESSEX	32
7.2.3	VIOLA	32
7.3	Summary	32
8	MetaScheduling Service MSS	33
8.1	Description	33
8.2	Experiences and results	35
8.2.1	PSNC	35
8.2.2	UESSEX	36
8.2.3	VIOLA	36
8.3	Summary	36
9	Summary	37
9.1	Deviations from planned activities	37
10	References	38
11	Acronyms	39



Table of Figures

Table 1.1: Demo applications deployed at each local test bed	8
Figure 2.1: Setup of a KoDaVis experiment, connecting two visualization clients at PSNC over different routes to the KoDaVis Server cluster in Juelich	10
Figure 3.1: DDSS GridFTP application servers and clients	13
Figure 3.2: DDSS Backup/Archive application servers and clients	14
Figure 4.1: WISDOM Overview	18
Figure 4.2: Screenshot after script-based AutoDock run on PHOSPHORUS testbed	21
Figure 5.1: Test of TOPS Application between SARA and FHG	24
Figure 8.1: Integration of the Metascheduling Service and UNICORE	34
Figure 8.2: Integration of the Metascheduling Service, the Network Service Plane and the Control Plane	35

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



0 Executive Summary

This document presents deliverable D6.5, the application and middleware testing report. It describes the tests of middleware and applications and the results of these tests conducted within the PHOSPHORUS testbed. The testbed itself is described in D6.1 and D6.2, the middleware and the applications in D3.2 and D3.3. The tests comprised different scenarios testing different possibilities of the user controlling network resources: using Grid middleware interacting with the Network Service Plane (NSP), direct interaction of applications with the NSP and using a command line interface to the NSP.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



1 Introduction

Several demonstration applications have been deployed in a number of sites of the local testbeds participating in the PHOSPHOROUS project as the PHOSPHOROUS grid testbed. This report summarizes the activities carried out during the deployment and test of the applications. Information about the setup of the testbed can be found in deliverables 6.1 and 6.2, whereas D6.3 reports on the functionality of the testbed in general. The following section reports on the deployment of each of the test applications and the problems encountered in each of the testbeds where these applications were deployed. Table 1.1 gives a brief overview of which application is supposed to run on each local test bed.

Test bed	KoDaVis	DDSS	WISDOM	TOPS	INCA
PSNC	X	X	X		
SARA				X	
UESSEX	X	X	X		X
VIOLA	X	X	X	X	
CESNET					X

Table 1.1: Demo applications deployed at each local test bed

The goal of this deliverable was to report about middleware and application testing on the PHOSPHOROUS testbed with a focus on the results of the testbed experiments. The middleware and the applications are described in D3.2 and D3.3 respectively. The tests comprised different scenarios testing different possibilities of the user controlling network resources:

- using Grid middleware interacting with the Network Service Plane (NSP),
- direct interaction of applications with the NSP, e.g. via a web service interface and,
- using a command line interface to the NSP.

The following sections provide description of the experiments made with the applications followed by sections describing the middleware installation. The individual sections comprise of a brief introduction of the respective components, the experience gained when installing and using it at different sites, open issues with the installation at the individual participating sites and a concluding summary.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



2 Distributed Collaborative Visualization (KoDaVis)

2.1 Description

KoDaVis enables users from various remote sites to collaborate in visualising scientific data sets. It is made up of several components:

- Data server
- Collaboration server
- Visualisation client
- UNICORE 6 Client

For the purpose of the tests conducted with this application, the data server and collaboration server components have been installed at FZJ. The other partners in this application's test – PSNC and UESSEX – each acted as a client in the tests that were run. This setup allowed for three modes of operation for the tests.

1. Test via Internet
2. Test via Phosphorus global testbed without middleware
3. Test via Phosphorus global testbed with middleware

All tests were conducted successfully and showed the global testbed is working and that KoDaVis is an appropriate application for this purpose. The application has also been demonstrated successfully at the first Phosphorus review in Poznan on December 13th, 2007.

The experimental setup comprised of the KoDaVis data and collaboration server, which had been installed on the Cray-XD1 machine in Juelich, and the visualization clients located in Poznan, Essex and Juelich. Firstly, the communication between the servers and the visualisation clients was successfully tested via SSH-tunnel over the Internet. This test was conducted by starting all parts of the application separately by hand, without using the UNICORE middleware. As a next step, the Phosphorus testbed was used to interconnect the KoDaVis data and collaboration servers with the clients in Juelich and in Poznan. Initially, a static link was established between the servers and the clients. Then, the network reservation and provisioning systems (NRPS) Argon, UCLP, DRAC and GMPLS via Thin-NRPS were applied to establish a connection, using the AJAX Demonstrator web interface as outlined in Figure 2.1. For this test, two visualization clients were run in Poznan,

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

one of which was routed via CRC, Canada, and the other one via i2CAT, Spain to Juelich. Due to the huge distance between Canada and Juelich or Poznan, this setup was not feasible for a collaborative visualization, as the long latencies of the synchronization messages sent to and from the collaboration server inhibited a fluent handling of the visualization.

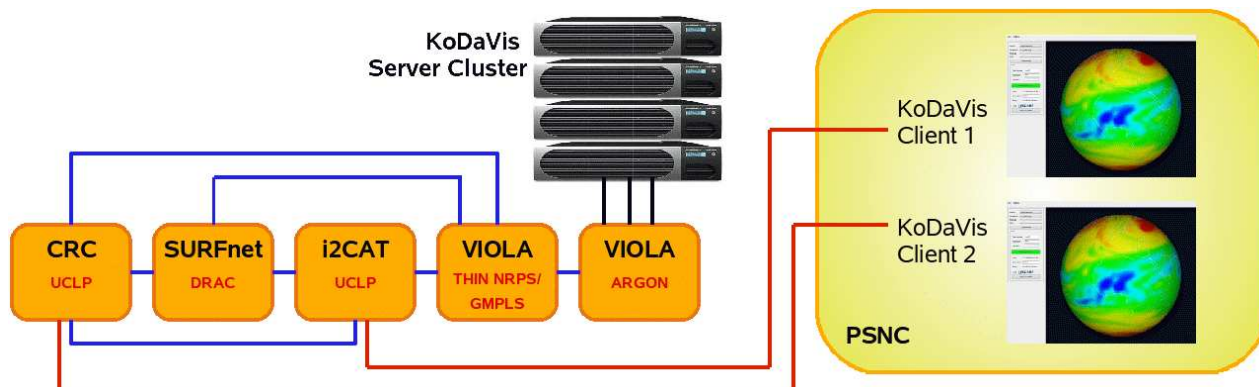


Figure 2.1: Setup of a KoDaVis experiment, connecting two visualization clients at PSNC over different routes to the KoDaVis Server cluster in Juelich

As a next step, the UNICORE client was deployed in order to start both the data and collaboration server in Juelich and the visualisation clients in Poznan and Juelich, and to establish a connection between them. Moreover, the client was enabled to make advanced reservations of bandwidth between the participating sites.

The experiments showed that all components are functional and are correctly interacting with each other, providing the intended functionality and ease of use. The successful integration and deployment of this first version of middleware and application has also been demonstrated at the Phosphorus Review meeting in Poznan in December 2007.

A lesson learned from the testbed experiments is that due to the many distributed components involved, it is difficult for less experienced users to identify the cause of errors that may occur. Moreover, in collaborative mode, it is currently difficult to find out which visualisation client is responsible for a potentially poor performance of the whole system. A suitable performance analysis tool remains to be developed. Furthermore, it could be verified, that the KoDaVis data server, which was only used in serial mode so far, constitutes a bottleneck in the current system. Therefore, the performance might be enhanced by deploying the parallel data server. This will be tested in the continuative performance experiments.

2.2 Experiences and results

2.2.1 PSNC

The visualization client was installed at PSNC. This installation was part of the experimental setup described above. The installation was first used for the demonstration during the 2007 project review.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

2.2.1.1 *Open issues*

None.

2.2.2 **UESSEX**

The visualization client was installed at UESSEX. This installation was part of the experimental setup described above.

2.2.2.1 *Open issues*

None.

2.2.3 **VIOLA**

The visualization client was installed at FZJ within the VIOLA testbed. This installation was part of the experimental setup described above. Also, the collaboration and data servers were installed at this site, as was the UNICORE 6 middleware to perform tests using the full software stack for this application's scenario. The installation was used for the demonstration during the 2007 project review and during the first phase of testbed experiments.

2.2.3.1 *Open issues*

None.

2.3 **Summary**

For the KoDaVis experiments we used a testbed with UNICORE 6 and the MetaScheduling Service (MSS) as middleware. The installation of the KoDaVis application was successfully completed as planned before the first phase of the testbed experiments. It could be shown that the application benefits from the resource co-allocation in terms of compute and network resources. Further, the middleware environment (UNICORE 6 and the MSS) required for the KoDaVis experiments, namely the UNICORE 6 client, the co-allocation service and the interface to the NSP for the bandwidth reservation, was installed and proved in the experiments to have all properties to support the KoDaVis application.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



3 Distributed Data Storage Systems (DDSS)

3.1 Description

Distributed Data Storage Systems (DDSS) are widely used in scientific and commercial applications in order to transport, exchange, share, store, backup/archive and restore data. Two DDSS applications are examined in the Phosphorus project: GridFTP service [GridFTP] and a commercial backup/archive/restore application from IBM Tivoli.

The GridFTP server exchanges data with single (one-to-one setup) or multiple clients (many-to-one setup). The single client can communicate with single (one-to-one setup) or many servers at the same time (one-to-many setup, 'striped' transmission mode). Data transmission between a single client-server pair can be done over one TCP/IP stream or in parallel using up to 128 parallel streams; this allows to effectively exploit the bandwidth of the network link even in the presence of a big transmission delay. The setup used in PHOSPHORUS is mainly the one-to-one scenario, run between the test nodes located in PSNC, FZJ, FHG and UESSEX.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>

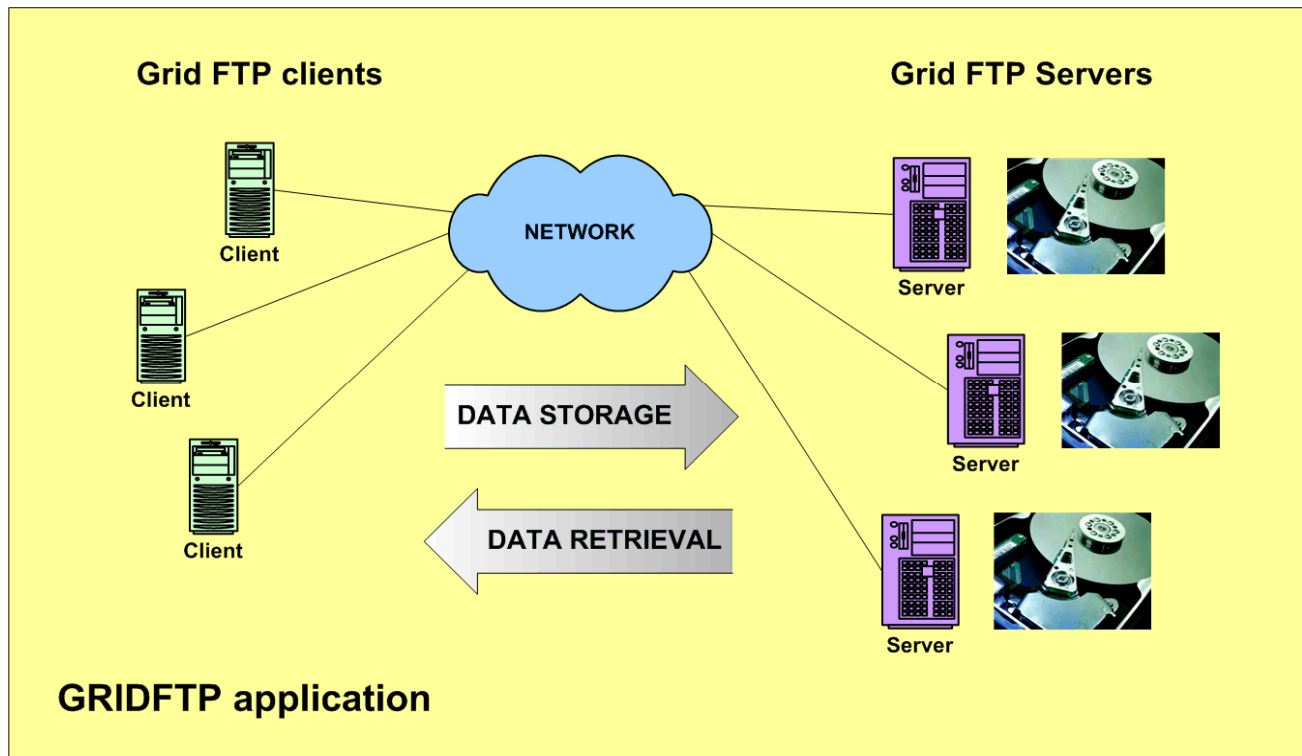


Figure 3.1: DDSS GridFTP application servers and clients

Backup/Archive clients collect the end-user data stored on the client machines and send them through TCP/IP streams to the Backup/Archive server. In case of failure or corruption of the user data, they can be restored from the server using the Backup/Archive client. B/A server itself manages the storage pools that can include disks, tapes and other storage media. User data are stored in these pools, according to a policy defined for a user. The setup that is used in the PHOSPHORUS test-bed contains two B/A servers: in PSNC and in FZJ. B/A clients are located both in PSNC and FZJ.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>

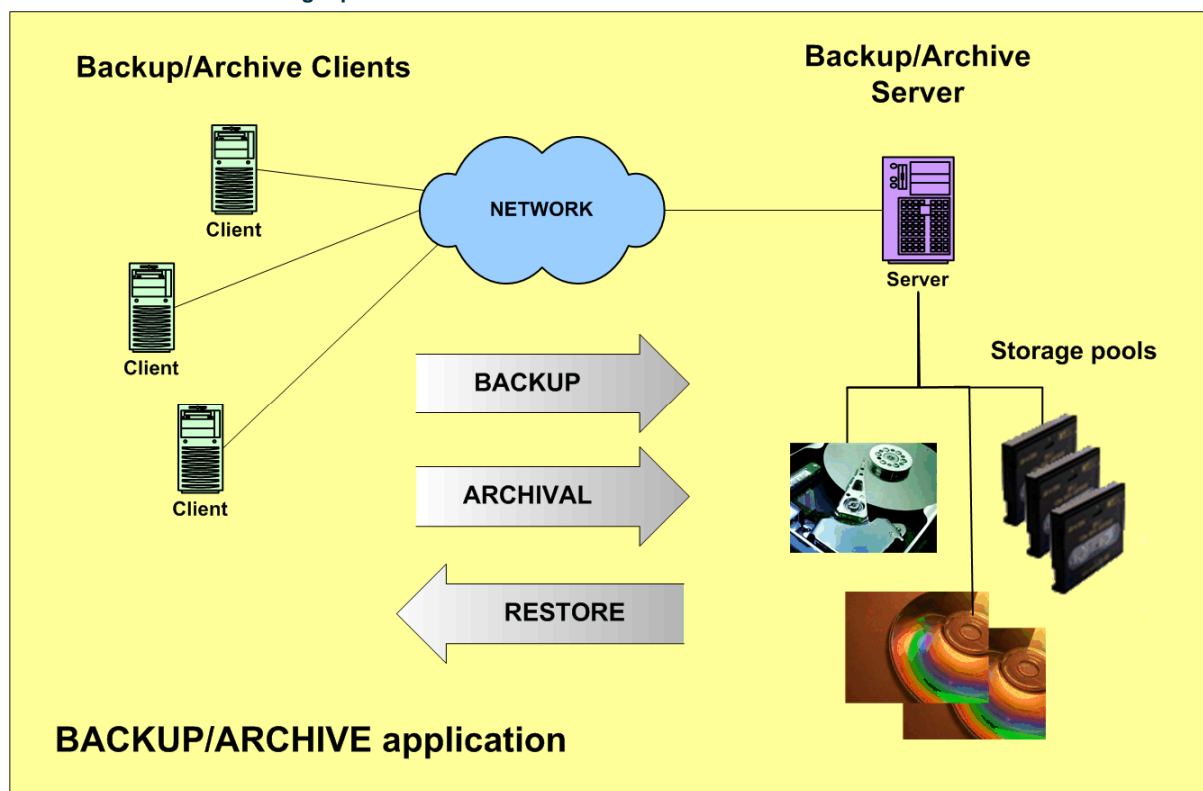


Figure 3.2: DDSS Backup/Archive application servers and clients

DDSS applications can generate various types of workloads. This makes possible evaluating various features of the PHOSPHORUS optical network and its link reservation capabilities. Link bandwidth can be examined by moving huge data volumes over the network links using multiple parallel transmission streams – such workloads are typical for moving large files between computers. Latency-related features of optical links can be examined using these DDSS scenarios in which multiple (hundreds to thousands) of small files are exchanged by DDSS clients and servers; this happens e.g. during making the backup copy of the users' home directories stored in file servers. Specific features of the protocols used in both DDSS applications used in PHOSPHORUS test-bed vary significantly. GridFTP protocol extends the well-known FTP protocol, by adding security, parallelism, commands pipelining and other features [GridFTP2]. Backup/Archive application in turn, uses proprietary vendor protocol, that includes compression, encryption and parallelism.

3.2 Experiences and results

3.2.1 Status of implementation and deployment

Adaptation of the DDSS applications to the PHOSPHORUS required some implementation activities as well as configuration and testing work.

First, the use cases that fit into the idea of testing PHOSPHORUS optical network features were defined. They were described using use case description template provided by the WP3 coordinator.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

Second, the applications were deployed in the local test-beds and the application setup was adapted to the capacities of these test-beds. Three institutions responded to the 'DDSS Call for Participation', which was prepared in the November of 2006 by PSNC. The applications use-cases were tuned up in order to fit to the capabilities of the local test-beds and the connectivity between them. This activity took place between December 2006 and January 2007. It was done by PSNC in collaboration with FZJ, FHG and UESSEX. In the same period, the local test-bed setup started. It has been continued during February till March 2007 period. DDSS test-bed was set-up in PSNC firstly, then in the local test-beds of the rest of the partners. From March 2007 to June 2007, testing of local DDSS was performed in PSNC. In the same time, PSNC and partners have set up the local test-beds and started preliminary, remote testing using Internet links. In this period a lot of configuration fixes and test-bed debugging actions were performed, including opening the firewall passes and tuning another settings of test nodes' operating systems to the application characteristics.

Third, in parallel to test-bed configuration and preliminary test actions, we investigated the possible changes of DDSS GridFTP and DDSS Backup/ Archive applications (February 2007) and implemented them (March 2007 to June 2007). In case of DDSS GridFTP, we introduced the functions which contain the NRPS function calls used to assign the optical network links to the application source code. We have also added timers and instrumentation functions to the GridFTP client code. They allow us to evaluate performance of the GridFTP data transfers done in the optical network. Opposite to DDSS GridFTP, DDSS Backup/Archive application source code was not available for us (as this is a commercial application from IBM Tivoli). Therefore, we decided to wrap the B/A application binary with the binary code prepared by us. The wrapper contains calls of the network resource reservation functions and instrumentation functions (timing, performance measurements). At that stage of the project, the NRPS function calls, added to DDSS applications, were in fact 'dummy' calls – they have not been implementing the actual reservation, as the reservation facility was planned to be available in the later phases of the project.

Four, since July 2007 to August 2007, we developed the tools for automating the execution of the DDSS use-cases and scenarios in the PHOSPHORUS. We also developed the mechanism for collecting the results of the tests. Automation tools run the DDSS tests in the test-bed periodically. Thus, they allow us to acquire a lot of measurements concerning the application and network performance. Results collection mechanism collects and registers the results in the relational database and generates graphs that depict the results of particular rounds of tests (graphs are put to the web-site). This allows us to perform results evaluation and analysis just after the test rounds finish and afterwards.

Five, we have run DDSS tests periodically, using above-mentioned automation tools. In August 2007, we started periodic testing the DDSS transfers between PSNC and FZJ. After completing the configuration of the rest of the test-beds – configuration of FHG and UESSEX test-beds finished between August and October 2007 – we have also performed a periodic runs of DDSS tests between PSNC and FHG as well as between PSNC and UESSEX.

Six, starting from October 2007, we attempted to switch from Internet links to optical network links. In October, the link between PSNC and VIOLA (FZJ, FHG) was configured. Switching to optical links required reconfiguration of network interfaces in test nodes and some changes in the configuration of test automating tools. There were some troubles and bugs in optical network configuration that were partially solved till February 2008. There was also a major failure of one of the test-bed sites that required reconfiguration of DDSS part of the test-bed. We were dealing with these issues during January 2008 and February 2008. March 2008, we worked on establishing the reserved optical link between PSNC and FZJ for DDSS purposes and worked on solving optical connectivity issues between PSNC and UESSEX.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

At the end of the first 18 months' period of the PHOSPHORUS project, we have done following improvements and modifications of the DDSS applications. First, the applications are ready to perform the optical network reservations, as this functionality will be ready. Second, the applications contain the instrumentation functions that allow measuring performance of the applications operation. Third, automation tools allow us to perform multiple rounds of application tests automatically. In that way, we are able to acquire multiple measurements captured for various application use-cases and various network links configurations (they can be changed in a meantime). Four, thanks to collecting the test results in the relational database, we are able to analyse the results any time, having access to both recent and historic performance data.

All these improvements and extensions of DDSS applications enable using these applications as effective benchmarks for evaluating the optical network links reservation features developed in PHOSPHORUS. From the other hand, thanks to extensions, these applications can benefit from NRPS features developed in the project.

3.2.2 PSNC

PSNC hosts both GridFTP servers and clients. This application is run on four HPC cluster nodes. These nodes act also as Backup/Archive clients. A dedicated storage server runs Backup/Archive server software. PSNC runs applications tests periodically i.e. every two hours. Each round of tests includes data transfers with various parameters: number of transmission threads, size of transmission data block, number and size of file to be transmitted (numerous small files versus some large files). Test results are automatically put to the website in a numeric and visual form (Gnuplot graphs).

Some Backup/Archive server licensing issues occurred in PSNC, therefore Backup/Archive testing started with some delay. However currently, full assumed functionality is available in the PSNC testbed.

3.2.2.1 Open issues

None.

3.2.3 UESSEX

GridFTP clients and servers have been successfully installed in UESSEX and have been successfully tested. The installation is fully operational.

There were some issues related to test environment reconfiguration in Essex, but they were solved and the full assumed functionality is available.

3.2.3.1 Open issues

None.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

3.2.4 VIOLA

VIOLA sites: FZJ and Fraunhofer run both GridFTP clients and servers. Additionally, FZJ runs Backup/Archive application server.

There were some issues in one of the VIOLA sites: FZJ's cluster experienced a major failure, preventing it from working. Therefore, the FZJ testbed was moved to another machine. Currently the testbed functionality is restored.

3.2.4.1 Open issues

None.

3.3 Summary

For the DDSS experiments we used a testbed with Globus Toolkit 4 [Globus] as middleware. The installation of the DDSS application was successfully completed as planned before the first phase of the testbed experiments. During the next phase of testbed experiments DDSS will use the bandwidth reservation capabilities of the testbed.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



4 WISDOM

4.1 Description

The WISDOM use-case consists of the virtual screening techniques AutoDock and FlexX, computing compounds of large-scale molecular dockings on targets implicated in diseases like malaria.

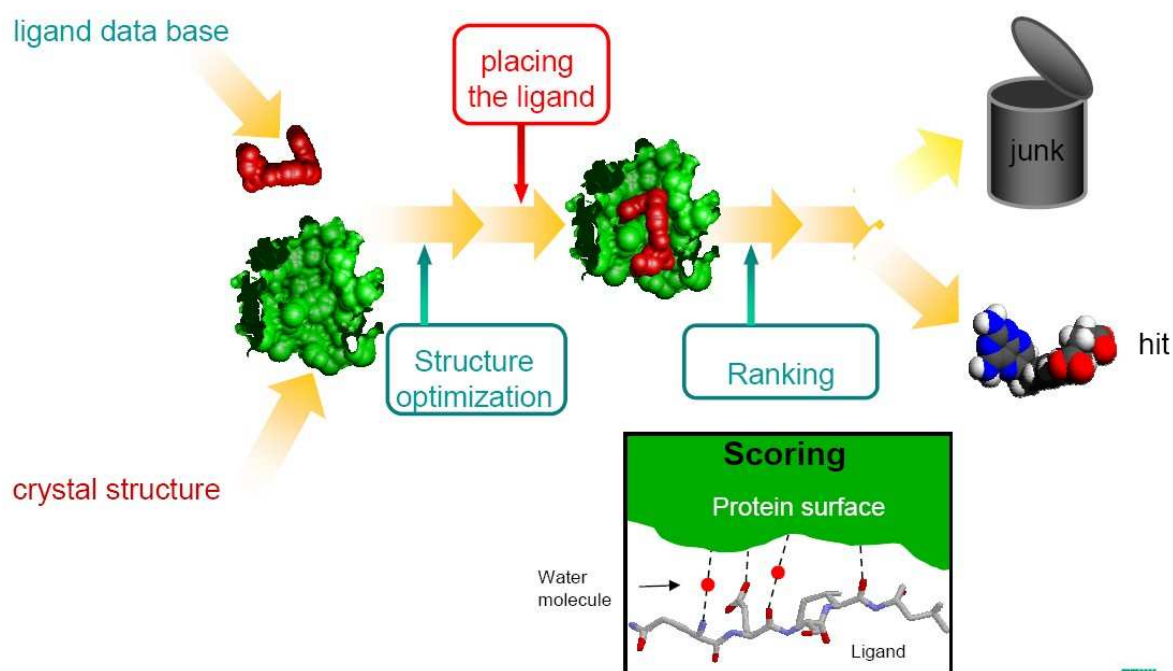


Figure 4.1: WISDOM Overview

- AutoDock is a suite of automated docking tools. It is designed to predict how small molecules, such as substrates or drug candidates, bind to a receptor of a known 3D structure.
- FlexX is an extremely fast, robust, and highly configurable (FlexX-able) computer program for predicting protein-ligand interactions.

More details about the PHOSPHORUS use case WISDOM and its applications are described in deliverable D3.3.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



4.2 Experiences and results

The work on the PHOSPHORUS use case WISDOM started with a call for participation. Finally, four sites (FHG, FZJ, PSNC, UESSEX) offered to provide compute resources and support to execute the WISDOM use case and its applications AutoDock and FlexX.

The next step of preparations was to inspect the first EGEE Data Challenge and their WISDOM workflows and identify necessary changes for the PHOSPHORUS environment. As a result of this analysis different WISDOM workflow phases were identified, which must be enabled based on MSS and UNICORE 6: Stage-in, execution and stage-out phases for each workflow. So, we are concentrating on automation of the WISDOM workflow.

- The WISDOM stage-in phase consists of licensing (for FlexX only) and distribution of input data from a specified input server. In the case of FlexX, these are RDF and Mol2 files. AutoDock uses Grid maps and DPF files for execution (see below).
- The WISDOM execution phase needs a good job control / monitoring and in cases of errors job resubmission functionality.
- The WISDOM stage-out phase includes the transfer of the local output data (FlexX Mol2 files, AutoDock DLG and Mol2 files) after termination from each site to a specified output server into a directory hierarchy or a database.
- Additionally there are pre- and postprocessing actions, e.g. input data format conversion, output data analysis, output data filtering.
- For this first test phase of WISDOM a test data suite from BioSolveIT was selected. So, in the beginning local installations and tests are done on all four PHOSPHORUS sites (see above) to test the docking software modules. For these runs the BioSolveit flexx-200 Testdata suite was used for both the applications AutoDock and FlexX. The BioSolveit flexx-200 Testdata suite is a subset from the PDB, where each ligand was separated from the protein-ligand-complex and hand-fixed concerning protonation, aromaticity, delocalisation, and formal charges. After these first modifications both WISDOM applications use the same input data set with their specific data formats. Additionally start scripts and data paths were modified to realise a good basis for UNICORE 6 executions. Finally code installations and test data sets were available on all participating PHOSPHORUS sites.

On all sites the file system is mounted to all computing nodes and therefore is identically on all nodes. To avoid writing to the same output files, the naming of the output files uses the hostname of the executing node. Thus output-data files are produced based on the schedule file in a node-specific way. This naming convention is also needed to transfer all results back to one single node, the so called output server node.

In this second phase deliverable D3.4 was produced and requirements for automatic WISDOM workflow executions were described. Additionally, application parameters were specified, which are needed for execution and data distribution, and which would improve user friendliness too (e.g. changeable standard parameter settings). Most of these settings will be realised as parameters of the script for WISDOM executions.

These different scripts for executions and data distributions are tested on the PHOSPHORUS Grid testbed, and will be needed for the automatic execution of WISDOM workflows with MSS and UNICORE 6. Thus further work and tests are needed in the following project phase to enable WISDOM workflows to be executed by MSS and UNICORE 6.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

Currently, the workflow engine of UNICORE 6 is only available in a prototype status. Thus the execution of both codes is actually done script-based. These scripts can be executed based on reservations of the available local resource management system. Different scripts are produced for the execution of AutoDock and FlexX jobs, which use the schedule file of the RMS for the execution. The jobs are running by 'ssh calls' on the reserved computing nodes. This need of alignment between node reservations and script files will be changed soon with a better UNICORE integration.

Further post processing functionalities should be supported after additional tests in the next project phase.

4.2.1 PSNC

AutoDock was installed without any problems on SC Poznan.

Unfortunately no FlexX installation was possible for the platform of SC Poznan (Itanium cluster), because this hardware is up to now not supported by code owner BioSolveIT.

First script-based GRID-WISDOM executions with all other sites were performed.

4.2.1.1 Open issues

Further tests with UNICORE 6, the application client and especially the UNICORE 6 built-in workflow support will be performed in the second phase of the testbed experiments.

4.2.2 VIOLA

AutoDock was installed without any problems on FHG (PACK cluster), and FZJ (Cray-XD1). After the Cray-XD1 system at FZJ was becoming unavailable due to an unrecoverable hardware failure, a new installation on the replacement system (Juggle) will be used for further work.

FlexX was installed on FHG (PACK cluster), and FZJ (Cray-XD1). Special port configurations are done on all sites to realize the contact to the Fraunhofer FlexX License Server system from all sites. At FZJ the replacement system (Juggle PC Cluster) will be used for further work.

First script-based GRID-WISDOM executions with all other sites were performed.

4.2.2.1 Open issues

Further tests on FZJ replacement system Juggle are needed. Further tests with UNICORE 6, the application client and especially the UNICORE 6 built-in workflow support will be performed in the second phase of the testbed experiments.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

4.2.3 UESSEX

AutoDock and FelxX were installed without any problems on UESSEX (Wisdom node).

First script-based GRID-WISDOM executions with all other sites were performed.

4.2.3.1 Open issues

Further tests with UNICORE 6, the application client and especially the UNICORE 6 built-in workflow support will be performed in the second phase of the testbed experiments.

```
viola03@packcs-e0:~/ng/ScriptExecution
viola03@packcs-e0:~/Docking/ScriptExecution> SE_AutoDock.csh PnodesListAll
AUTODOCK HOST: pack02-e0
AUTODOCK ACTUAL TIMDOCK ENTRY: 13P_1LYX
SUBMIT: AUTODOCK_HOST pack02-e0
AUTODOCK EXECUTION: 13P_1LYX

-----

/home/viola03/Docking/AutoDock/i86Linux2/autodock4: Successful Completion on "pack02-e0"

-----

AUTODOCK ACTUAL TIMDOCK ENTRY: 2PG_1LYX
SUBMIT: AUTODOCK_HOST pack11-e0
AUTODOCK EXECUTION: 2PG_1LYX

-----

/home/viola03/Docking/AutoDock/i86Linux2/autodock4: Successful Completion on "pack11-e0"

-----

AUTODOCK HOST: node01.phosphorus.man.poznan.pl
AUTODOCK ACTUAL TIMDOCK ENTRY: 3PG_1LYX
SUBMIT: AUTODOCK_HOST node01.phosphorus.man.poznan.pl
AUTODOCK EXECUTION: 3PG_1LYX

-----

/home/viola03/Docking/AutoDock/ia64Linux2/autodock4: Successful Completion on "node01.phosphorus.man.pozne"

-----

AUTODOCK HOST: wisdom1.essex.ac.uk
AUTODOCK ACTUAL TIMDOCK ENTRY: 3PP_1LYX
SUBMIT: AUTODOCK_HOST wisdom1.essex.ac.uk
AUTODOCK EXECUTION: 3PP_1LYX

-----

/home/viola03/Docking/AutoDock/i86Linux2/autodock4: Successful Completion on "wisdom1"

-----

viola03@packcs-e0:~/Docking/ScriptExecution> █
```

Figure 4.2: Screenshot after script-based AutoDock run on PHOSPHORUS testbed

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



4.3 Summary

For the WISDOM experiments we used a testbed with UNICORE 6 and the MetaScheduling Service (MSS) as middleware. The installation of the WISDOM application was successfully completed as planned before the first phase of the testbed experiments. It could be shown that the application benefits from the resource for compute resources and network resources. Bandwidth reservation was used for the stage-out phase of the WISDOM workflow when the results of the simulation are transferred back to the site of the user. Further, the middleware environment (UNICORE 6 and the MSS) required for the WISDOM experiments, namely the distribution of jobs within the testbed and the interface to the NSP for the bandwidth reservation, was installed and proved in the experiments to have all properties to support the WISDOM application. For the first phase of testbed experiments a script-based workflow execution was implemented and used. For the second phase the built-in support for workflows of UNICORE 6 will be used.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



5 TOPS

5.1 Description

TOPS (Technology for Optical Pixel Streaming) [TOPS-2] [TOPS-3] enables remote viewing of large scientific datasets (2D or 3D) on high resolution display devices (Tiled Panel Displays). Uncompressed pixels are streamed by TOPS from data center to remote displays. Allocating network resources requires manual intervention from engineers and developers.

The TOPS application has been chosen as application because it was ideally suited for testing and demonstration of a real-time, high bandwidth application within the Phosphorus testbed. It is a point-to-point image streaming application for ultra-high resolution images (image sizes up to 100,000x100,000 pixels), intended to be used by researchers working on very large datasets at computing centers like SARA. Because these large datasets are impossible to mirror locally, methods have been developed to browse the data remotely using image generation techniques. The resulting images are still stored locally at the computing center, and only the part that is being viewed is transferred to the researcher's site. At his site, the researcher has access to a large resolution tiled panel that is capable of displaying this image stream.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

Test of TOPS Application between SARA and FhG (IAIS.VE) via Phosphorus testbed

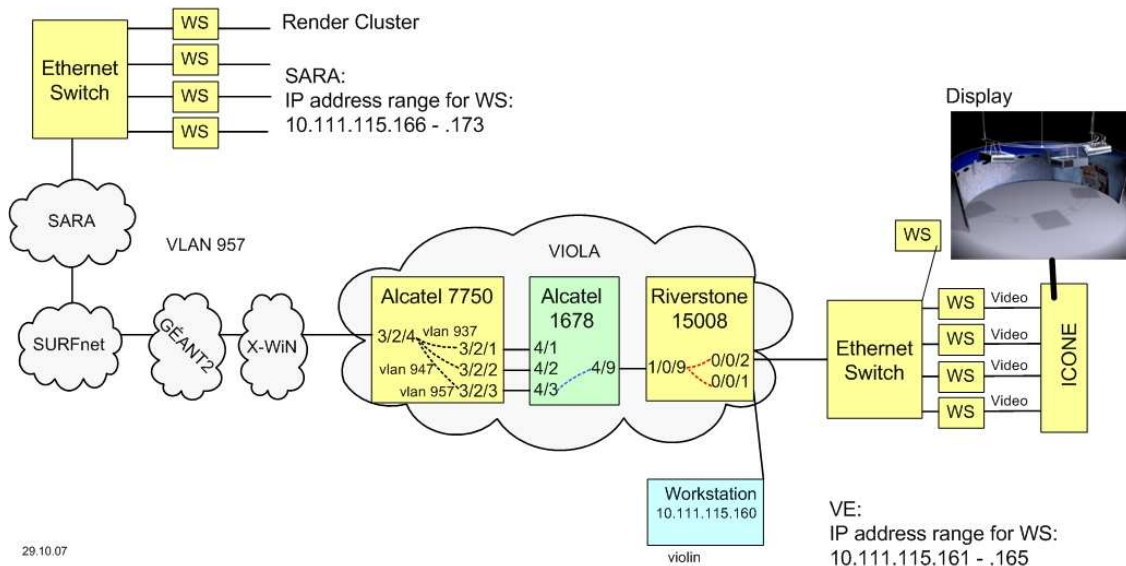


Figure 5.1: Test of TOPS Application between SARA and FHG

In contrast to distributed applications in grid computing the TOPS application is running under interactive control of the user. Besides computing and networking facilities the application renders pictures that are viewed on a remote, large size display by a user.

The TOPS application can take care of all the necessary steps to transport and display the image data. Adaptations were necessary to support various modes of visualization (synchronization, stereo display, animation & video), as well as to support the Phosphorus test bed infrastructure. Finally, steps were undertaken to support a network provisioning scheme, so as to be able to demonstrate all the Phosphorus accomplishments within one application.

For TOPS the data and the rendering machine are located in Amsterdam, the result will be streamed over the network to Sankt Augustin and be viewed on the i-CONE™ display of FHG IAIS. At the display site the scientist can interact (navigate through) with the application at the rendering site. The i-CONE™ is a cylindrical 270-degree projection display system with high-resolution and evenly curved projection surfaces. The i-CONE™ has a visitor capacity of approximately 20 people. The display consists of four projectors with a resolution of 1600x1460 pixel a vertical refresh rate of at 105 Hz each. The projectors support active stereo mode which means that for each eye half of the refresh rate is available. The input for the projectors is provided by a cluster of four workstations equipped with NVIDIA Quadro FX graphics cards and a gigabit network interface.

The SARA and FHG test machines were moved from the VIOLA test bed and connected through the Phosphorus infrastructure.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



5.2 Experiences and results

For TOPS the data and the rendering machine are located in Amsterdam, the result will be streamed over the network to Sankt Augustin and be viewed on the i-CONE™ display of FHG IAIS.

5.2.1 SARA and FHG

SARA has been studying the separation of graphics card and display in this concept. It must be noted that when we mention an OptIPuter graphics card or an OptIPuter display, that we are referring to a large-scale infrastructure. Thus an OptIPuter display is actually a cluster of displays, or a tiled panel display. And an OptIPuter graphics card is actually an aggregate of high performance workstations, each equipped with fast 3D hardware and dual processors.

SARA's approach to driving tiled displays with remote render clusters is a lightweight approach. Our key insight into this problem is based on the short lifespan of a picture element in a video stream. At 50 frames per second, a picture element, or pixel, is only visible for 20 ms. If such a pixel is lost during communication, it will remain unnoticed, because apart from spatial coherence, video streams also exhibit temporal coherence. Our approach therefore uses unreliable network transport (UDP) as opposed to a reliable (TCP) transport. This will give a valuable benefit, as high throughput is easily realized when acknowledgements from remote sites with large latencies are no longer required.

In the first month of the projects, the hardware and software requirements for TOPS were documented. A call for participation was written and published, aiming to gather more researchers and/or participating sites for testing and demonstrations. Also, an overview of other image streaming tiled panel environments was made, to be able to compare them with TOPS. At the SC|06 supercomputing conference (Nov. 2006, Tampa, Florida), these were presented alongside, together with a general presentation of the Phosphorus projects, its goals and aspirations. Finally, input was supplied to deliverable D3.1, defining necessary application interface changes.

In the following months, test machines at SARA and FHG have been set up for the TOPS application and documentation was supplied for partners and researchers to learn more about the application and its possibilities and limitations. As distribution of a graphics visualization application is not state of the art, decisions at FHG had to be made to allow external access to the FHG-IAIS display facilities. Due to internal (i.e. firewall) policies the test machines have been set up and put outside the IAIS institute network. As for using the display facilities a human operator (viewing the pictures) must be on site, it has been decided that the assignment of the test machine cluster to the display (i.e. switch video sources between machines and projectors, switch on/off projectors) is done by the user, using the available local procedures. It is not intended to make it an automated process as the situation always is different for different sites. For the application it is assumed that a high resolution picture stream is available at the network interface of the machine acting as a video source for a display.

Content was generated in different formats. Cross-Atlantic tests were performed between SARA and CalIT2 (California, USA), yielding valuable performance data on light paths over long distances. Network tests were set up and performed with the i-CONE display facility at FHG. All these activities resulted in a first design document on necessary adaptations of the TOPS software.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

In March 2007, connectivity with the VIOLA test bed was established, remote access was provided and TOPS was installed at the FHG i-CONE display facility. Parts of TOPS were rewritten as per the design document. Some storage hardware was upgraded in the SARA infrastructure to accommodate the large datasets that were planned in the Phosphorus test bed. A DRAC test bed was set up together with SURFnet in the Netherlands, and tests were initiated in this local test bed.

In the next few months, further tests between SARA en FHG were performed. Networking issues with firewalls, access control lists and MTU settings were found and tackled. The test bed machines have been tuned for optimal performance for image streaming using TOPS. The display facility at FHG was supplied with the necessary hardware to steer the TOPS application.

After the summer break, tests were performed with high resolution (4k) video streams. A multicast demonstration of 4k video streaming was given at the GLIF meeting in Prague, as well as at the eChallenges 2007 conference in The Hague. The SARA and FHG test machines were moved from the VIOLA test bed and connected through the Phosphorus infrastructure. A dedicated 1 Gbps light path was used to connect the SARA rendering facility with the i-CONE display facility at FHG, and various test sessions were held. A number of networking issues (MTU mismatch) had to be resolved, and further tests between SARA and FHG were performed. A number of issues with the application were identified, resulting in a second design document for TOPS. At the SC|07 supercomputing conference, the TOPS application use case was demonstrated in co-operation with a researcher that is currently using the TOPS software and infrastructure at his own lab.

From January 2008, the requirements and recommendations as set forth in the second design document have been implemented in a final version of the TOPS application. Mostly, the changes are necessary to allow sites to use the software without rewriting configurations or implementing code changes, and make use of a grid infrastructure in line with the Phosphorus project. During this process, it has become clear that a direct connection to the Phosphorus network provisioning system(s) would be invaluable to the use case, and it has been decided to extend the development of the User Interface of TOPS beyond the 18 month boundary in order to implement an interface to these systems. To that end, contact has been made with SURFnet and Nortel for access to DRAC resources.

The final version of TOPS has the following improvements compared to the original 2006 version:

- resource allocation through central scheduling and configuration user interface
- complete separation of render and display code
- implementation as daemon invoking the actual display or render code
- parameterization of:
 - display size & bits per pixel
 - display configuration
 - MTU size
 - IP configuration
 - data set location
 - user interface location
- hard coded values removed and put in configuration files.

5.2.1.1 Open issues

Better support for other file types, integration of bandwidth reservation for the streaming.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



5.3 Summary

This application did use a dedicated fibre between the VIOLA testbed and SUFNET and does not use any of the middleware available in the testbed for the first phase of the testbed experiments. During the second phase an interface to the NSP for the bandwidth reservation for the period of the streaming will be used.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



6 INCA

6.1 Description

INCA is an intelligent storage network that provides data-transfers with high performance. Functionalities of INCA are divided in two layers. The first layer is called **Data Management Layer** and its purpose is to infuse in the system totally distributed and automated management functionalities. In more detail we test and evaluate four major features that each storage network must have:

1. The first one is scalability. In order to achieve this goal we have implemented a distributed hash table (DHT) that is used for the distributed metadata maintenance and the distributed location of them.
2. The second is the balanced use of storage network resources in terms of storage space and network bandwidth. In order to achieve this goal any data chunk is hashed and according to the output of a uniform hash function is stored in the proper storage node.
3. The third is the fault-tolerance of a storage network. A stable routing algorithm used for the data and metadata location has been developed and has been evaluated.

As for the second layer it is called **Storage protocol layer**. It's purpose is to provide:

- Point-to-point buffer transfer in order to store chunks in the chosen (by the upper layer) node;
- The glue between Data management layer and the storage control layer.

In the following the status of implementation and deployment is described. As for the first layer (**Data Management Layer**) the following functionality has been implemented and evaluated:

1. Node insertion and removal
2. Distributed routing table and DHT zone maintenance
3. Application layer routing mechanism
4. Metadata insertion and removal
5. Distributed metadata base maintenance
6. Metadata insertion and data placement

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

7. Metadata location and data retrieval

Additionally for the second layer (**Storage protocol layer**) we have evaluated:

1. Point-to-point bandwidth utilization
2. ACK strategies during high rate transmissions.
3. Traffic Congestion.
4. Rate Adaptation.
5. Implementation of an Authentication Checksum in order to secure both the non-corruption of packets and the non-spoofing.

Positive effects on **Data management layer** were observed through implementation. For the maintenance of a DHT and the routing tables special attention has to be paid in order to keep the zones and the routing tables consistent. Temporary inconsistency takes place while nodes are added or removed. System has to handle this situation.

During the routing process, performance of the system is affected by the implementation of routing mechanism. Also stability of the system is affected. A routing algorithm was implemented, with purpose to infuse these two parameters into INCA.

Performance of hash function used in the DHTs towards the balanced metadata placement does well. As for the balanced data placement small data chunks have to be used due to the variant file size of the files that inserted in the system.

Special attention has to be paid on replication, because it is the only way towards consist and data safe fault tolerant system.

Aspects regarding **Storage protocol layer**. Achieved performance of 50% below on another testbed indicates some issues on the testbed. Those issues are partially linked to the kind of access and hardware. This brings in evidence some key points:

- The need of jumbo frames in order to exploit the gigabit link (ok, this is a well known issue but it seems to not be taken in account in other systems);
- A good tuning of nodes is needed;

In absence of that some development has to be done to deal with kind of issues and to try to counter that in **storage protocol layer** (precisely **HITP**). Such kind of development is vertical to ACK- and rate adaption strategies. It is focused in lowering bandwidth usage caused by improper node setup.

6.2 Experiences and results

6.2.1 PSNC

The INCA application has been deployed at PSNC. Several tests were run on the link PSNC-CESNET following these two use cases:

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

- Scientific applications
- Video streaming applications

6.2.1.1 *Open issues*

See below (6.2.2.1).

6.2.2 **CESNET**

The INCA application has been deployed via CESNET. Several tests were run on the link PSNC-CESNET following these two use cases:

- Scientific applications
- Video streaming applications

6.2.2.1 *Open issues*

These tests by far didn't perform as well as on another testbed, mainly because of the lack of jumbo frames on the peer at PSNC. The exact cause of the poor performance remains to be examined. Additionally, firewall issues were encountered during the tests. The overall performance using the PHOSPHORUS testbed was 50% below the expected. Solutions to overcome the remaining issues are evaluated. During the second phase more sites should be used for experiments.

6.3 **Summary**

The installation of the INCA middleware was successfully completed as planned before the first phase of the testbed experiments. It could be shown that the application benefits from the optical testbed. However, there are some performance issues that will be addressed and resolved during the second phase of experiments. Moreover, the INCA will be deployed at more sites of the testbed.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



7 Sites running UNICORE 6

7.1 Description

UNICORE [unicore] has been chosen as the middleware to be linked with both the WISDOM and KoDaVis applications. UNICORE supports a plugin architecture at the client side that allows for application specific plugins to be included. The suitability of the plugin concept has been evaluated for both applications. The work on middleware design started in March 2007.

The first component of the UNICORE middleware to be adapted to the KoDaVis use-case was the client. An application specific plugin was written that can be added to any UNICORE client via the plugin architecture described above. In the following, a specific service for the UNICORE server side was developed, which allowed the creation and management of KoDaVis data server instances as well as collaborative sessions. This service was then coupled with the graphical KoDaVis client through the UNICORE client.

The meta-scheduler (MSS), which is another of the middleware landscape employed in Phosphorus, had to be integrated with UNICORE. Thus, it would be possible to co-allocate network and Grid resources by means of the MSS and make advance reservations of UNICORE resources through the middleware. For UNICORE, this meant to extend the interface of the target systems to expose the advance reservation capabilities of the underlying batch system. Also, as UNICORE supports a host of batch systems, the Target System Interface (TSI) needed to be adapted for the particular batch system running on the Grid resource. The TSI was first developed for the Cray XD1, which was the Grid resource dedicated to Phosphorus at FZJ. The TSI had to be implemented for the combination of the Torque batch system and Maui scheduler, when the XD1 was replaced with another machine (JUGGLE cluster) due to a hardware failure.

UNICORE was chosen to be the middleware to be integrated with KoDaVis.

7.2 Experiences and results

7.2.1 PSNC

UNICORE 6 environment was installed.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

7.2.1.1 *Open issues*

None.

7.2.2 **UESSEX**

UNICORE 6 environment was installed.

7.2.2.1 *Open issues*

None.

7.2.3 **VIOLA**

UNICORE 6 environment was installed.

7.2.3.1 *Open issues*

Further tests of MSS integration.

The use of UNICORE for multiple applications within the project required all developers to gain further insight into the middleware.

7.3 **Summary**

During the first 18 months the UNICORE middleware made a major transition from UNICORE 5 (which was installed early 2007 after PHOSPHORUS started) and the web service based UNICORE 6, which became available mid 2007. Together with UNICORE 6 new clients with different capabilities have been introduced. Since it offers the best relation between capabilities and complexity of the interface the application client is used in the PHOSPHORUS testbed. The new built-in workflow support of UNICORE 6 will be used during the second phase of the testbed experiments to manage the WISDOM workflows.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



8 MetaScheduling Service MSS

8.1 Description

The initial version of the MetaScheduling Service (MSS) introduced in the PHOSPHORUS project was based on developments of the German VIOLA project.

During the first phase of the middleware adaptations WP3 started gathering requirements from the PHOSPHORUS applications (contributed by partners of WP3) and from the other WPs, especially WP1, WP2, WP4 and WP6. In the same time a use-case document was produced that further contributed to the specification of requirements for the MSS. Based on this work a series of inter-workpackage meetings was organised in Amsterdam and Utrecht, which resulted in the final version of the specification of requirements and a first draft of the interfaces to be implemented towards the NSP, the G2MPLS and the AAI of the network layer.

During the first months of the project the integration of MSS into the UNICORE system was enhanced, making the overall architecture of the PHOSPHORUS middleware more consistent. Figure 8.1 presents the resulting architecture. In contrast to the VIOLA approach where the MSS was using own adapters for the communication with the local resource management systems (RMS), the MSS in this version of the integration is already using the target system interfaces (TSI) of UNICORE to communicate with the local RMS. In order to allow also reservations of resources and the negotiation of available time-slots for the execution of an application the TSIs have been extended. For the communication with the UNICORE systems components the MSS uses the UNICORE protocol language (UPL). To avoid extensive changes in the UNICORE internal communication it was decided to use an adapter that maps the WS-Agreement interface of the MSS to the UPL interface of UNICORE. To that end, we achieved interoperability while the internal protocol of both UNICORE and MSS could remain unchanged. With this new architecture we additionally prepared the ground for the integration with other middleware systems like the Globus Toolkit 4 as shown in Figure 8.1 Moreover, the ARGON NRPS is also connected using the same adapter mechanism: the MSS WS-Agreement protocol is mapped to the ARGON web service interface.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>

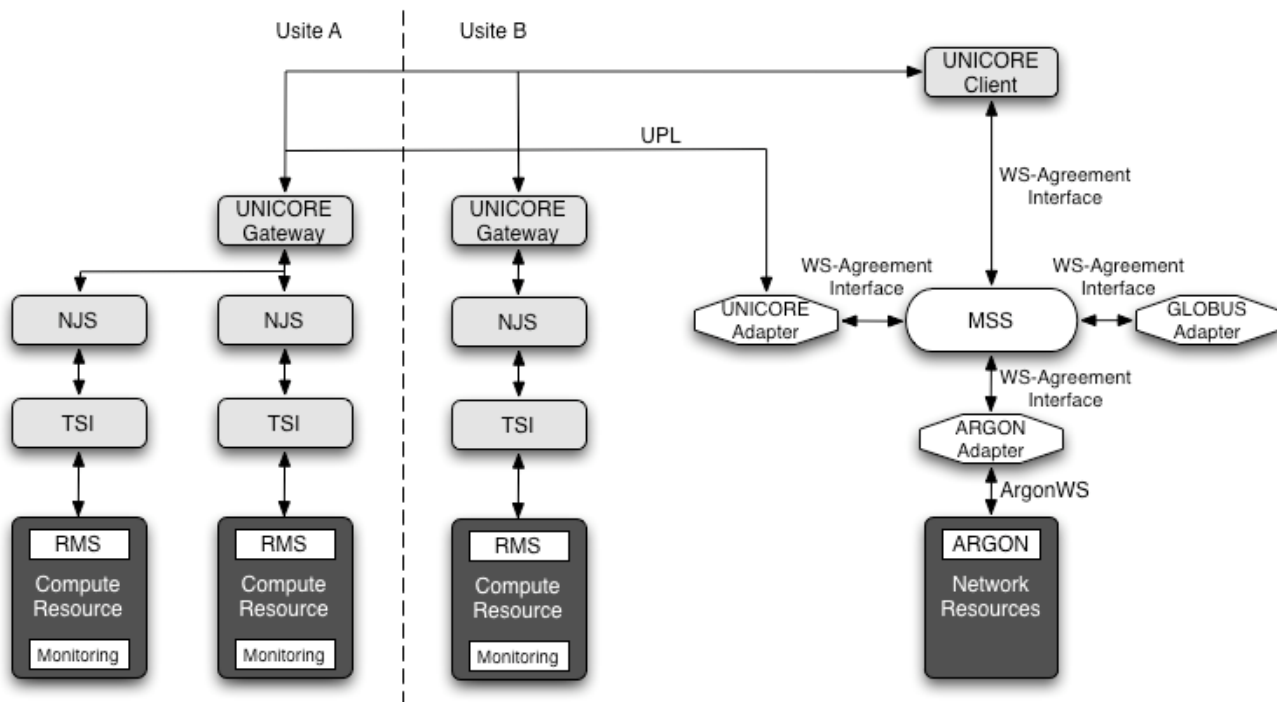


Figure 8.1: Integration of the Metascheduling Service and UNICORE

Parallel to this activity the resource selection mechanism based on semantic matching of application requirements and resource capabilities was outlined and the requirements for an interface of resource selection service (RSS) and the choices for an implementation of the RSS with the MSS were evaluated. It was decided to use the evolving standard of the Open Grid Forum (OGF) for RSS, which is currently under development in the OGF OGSA-RSS working group.

Based on the WP1 specification of the network service plane (NSP) the MSS adapter towards the ARGON NRPS was replaced by an adapter for the NSP. Figure 8.2 shows the current architecture that was used for the demonstrations during the first project review in December 2007 and which is now used as part of the regular PHOSPHORUS testbed infrastructure. As Figure 8.2 depicts the different NRPS available in the PHOSPHORUS network environment are have an interface to the NSP while the request for network resources from the MSS is handled by the NSP and transparently forwarded to the respective NRPS.

Figure 8.2 also shows the interface with the control plane and the G2MPLS developed by WP2 in the PHOSPHORUS project. The implementation of these interfaces will be mostly done after month 18. More details about the architecture can be found in D3.5.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

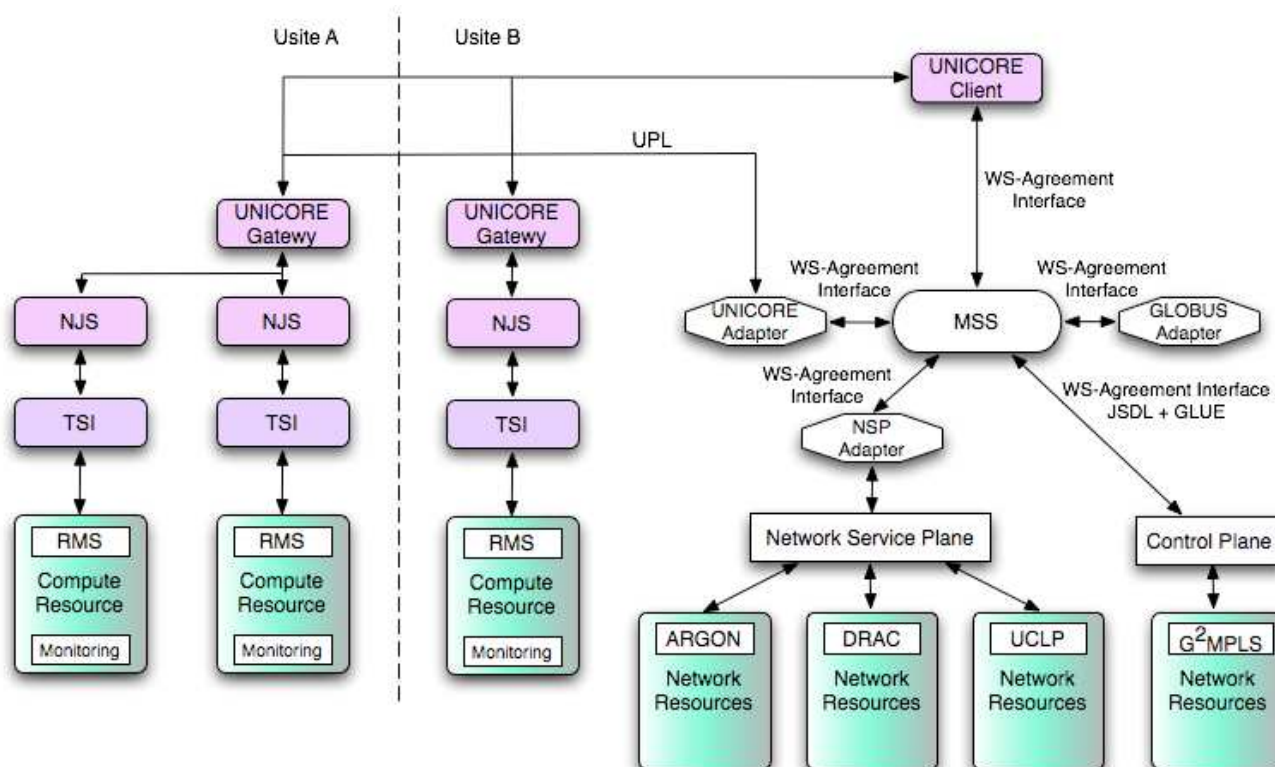


Figure 8.2: Integration of the Metascheduling Service, the Network Service Plane and the Control Plane

8.2 Experiences and results

At month 18 of the project the MSS installation in the testbed is stable and ready to be used in the second phase of testbed experiments, which is scheduled starting month 19. The implementation of the interface to the network control plane to provide middleware service to the network layer will be completed as foreseen month 24 of the project. Since the MSS acts as Grid Scheduler for the entire PHOSPHORUS testbed only one instance needs to run. This MSS instance is running in the VIOLA testbed on one of the FHG resources.

8.2.1 PSNC

The local UNICORE 6 environment is properly working with the MSS installation.

8.2.1.1 Open issues

None.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

8.2.2 UESSEX

The local UNICORE 6 environment is properly working with the MSS installation.

8.2.2.1 Open issues

None.

8.2.3 VIOLA

The local UNICORE 6 environments at FZJ and FHG are properly working with the MSS installation. MSS itself is installed at one of the FHG resources. The final modifications of the MSS concerned the integration in the new web services-based UNICORE 6, which replaced UNICORE 5 as Grid middleware in the testbed. The modifications concerned mainly the new security model of UNICORE 6, which made changes in the MSS interfaces to the UNICORE client and the UNICORE gateway necessary. As a result, the middleware in the PHOSPHORUS testbed is following the WS-Security standard now.

8.2.3.1 Open issues

None.

8.3 Summary

The latest version of the MSS is supporting both UNICORE 6 and the NSP interface is up and running. A initial version has been successfully used during the first phase of the testbed experiments. The last experiments proved that the MSS is now fully compliant to the new security model of UNICORE 6. Current work is addressing the interface to the G2MPLS control plane and the transfer of Grid authentication and authorisation information to the network layer.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



9 Summary

The goal of this deliverable was to report the state and the functionality of the applications and the middleware deployed in the PHOSPHORUS testbed. The middleware UNICORE 6, the MetaScheduling Service MSS and the bandwidth reservation interface to the NSP have been demonstrated together with the KoDaVis application during the first PHOSPHORUS review in Poznan on December 13th, 2007. In addition to KoDaVis running in the testbed, DDSS and INCA were successfully deployed and tested as demonstration applications. An initial deployment of the WISDOM application has been completed and first tests have been executed successfully. Future tasks include the transition of the middleware to UNICORE 6, the integration of the MetaScheduling Service with G-OUNI and G2MPLS, and the optimization of applications in the testbed.

Furthermore, the deployment of the TOPS applications and the full deployment of WISDOM based on the UNICORE 6 built-in workflow support are scheduled for the second phase of the testbed experiments.

9.1 Deviations from planned activities

No deviations from the project plan can be reported.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



10 References

- [D3.2] Phosphorus Deliverable 3.3 – Report on middleware extensions and implementation
<http://www.ist-phosphorus.eu/wiki/images/b/b0/Phosphorus-D3.2.pdf>
- [D3.3] Phosphorus Deliverable 3.3 – Report on initial adaptation of applications
<http://www.ist-phosphorus.eu/wiki/images/6/61/Phosphorus-D3.3.pdf>
- [D3.5] Phosphorus Deliverable 3.5 – Final report on achievements and results of the first 18 months' period
<http://www.ist-phosphorus.eu/wiki/images/2/2d/Phosphorus-D3.5-v0.5.doc>
- [D6.1] Phosphorus Deliverable 6.1 – Design of Testbed
<http://www.ist-phosphorus.eu/wiki/images/c/cf/Phosphorus-deliverable-D6.1.pdf>
- [D6.2] Phosphorus Deliverable 6.2 – Report about the Testbed
<http://www.ist-phosphorus.eu/wiki/images/a/af/Phosphorus-deliverable-D6.2-version3.pdf>
- [D6.4] Phosphorus Deliverable 6.4 – Running demonstration application
http://www.ist-phosphorus.eu/wiki/images/9/99/Phosphorus-deliverable-D6.4_v0.5-3.pdf
- [Globus] <http://www.globus.org>
- [GridFTP] http://www.globus.org/grid_software/data/gridftp.php
- [unicore] <http://www.unicore.eu>

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



11 Acronyms

AAA	Authentication, Authorisation, Accounting
DDSS	Distributed Data Storage Systems
e2e	end to end
EGEE	Enabling Grids for E-scienceE (European Grid Project)
FC	Fibre Channel
FC-SATA	Fibre Channel to SATA technology (mixed technology used in disk matrices: disk matrix have Fibre Channel ports for hosts connectivity, but contains SATA disk drives)
GEANT2	Pan-European Gigabit Research Network
GEANT+	the point-to-point service in GEANT2
GMPLS	Generalized MPLS (MultiProtocol Label Switching)
G2MPLS	Grid-GMPLS (enhancements to GMPLS for Grid support)
GT4	Globus Toolkit Version 4 (Web-Service based)
INCA	
KoDaVis	Tool for Distributed Collaborative Visualisation
MSS	MetaScheduling Service
NREN	National Research and Education Network
NRMS	Network Resource Management System
NRPS	Network Resource Provisioning System
PoP	Point of Presence
Protégé	Ontology Editor and Knowledge Acquisition System
QoS	Quality of Service
SNMP	Simple Network Management Protocol

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Application and middleware testing report

TOPS	Technology for Optical Pixel-Streaming
TPD	Tiled Panel Display
TUAM	Tool for Universal Annotation and Mediation
UNI	User to Network Interface
UNICORE	European Grid Middleware (UNIform Access to COmpute REsources)
VLAN	Virtual LAN (as specified in IEEE 802.1p)
VIOLA	A German project funded by the German Federal Ministry of Education and Research (Vertically Integrated Optical Testbed for Large Applications in DFN)
VPN	Virtual Private Network
WISDOM	Wide In Silico Docking On Malaria

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>



Disclaimer

The Phosphorus project is funded by the European Commission under the FP6 contract no. 034115. This document contains material which is the copyright of Phosphorus contractors and the EC, and may not be reproduced or copied without permission. The information herein does not express the opinion of the EC. The EC is not responsible for any use that might be made of data appearing herein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

Project:	Phosphorus
Deliverable Number:	D6.5
Date of Issue:	31/03/08
EC Contract No.:	034115
Document Code:	<Phosphorus-WP6-D6.5>