034115

PHOSPHORUS

Lambda User Controlled Infrastructure for European Research

Integrated Project

Strategic objective:
Research Networking Testbeds

Deliverable reference number D3.3

Report on initial adaptation of application

Due date of deliverable: 2007-09-30
Actual submission date: 2007-09-30
Document code: Phosphorus-WP3-D.3.3

Start date of project:
October 1, 2006

Duration:
30 Months

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

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

Report on initial adaptation of application

Abstract

This report describes the status of the changes and implementation of the first version of the applications. In the first phase of the project, use-cases have been defined that utilize the new functionality provided by the Phosphorus test-bed. Based on these use-cases, required modifications and extensions of the application have been designed. This report summarises the current status of the implementation of these changes, as well as the status of the application deployment in the test-bed.

Jointly with the first version of extended middleware that is described in Deliverable D3.2 the applications are ready to perform first experiments in the Phosphorus test-bed. The plans for the experiments in the first testing phase scheduled for project month 13 and 14 are also sketched in the report.

Project:	Phosphorus
Deliverable Number:	D.3.3
Date of Issue:	30/09/07
EC Contract No.:	034115
Document Code:	Phosphorus-WP3-D.3.3

Table of Contents

Table of Figures

0 Executive Summary

This report describes the status of the changes and implementation of the first version of the applications. In the first phase of the project, use-cases have been defined that utilize the new functionality provided by the Phosphorus test-bed. Based on these use-cases, required modifications and extensions of the application have been designed. This report summarises the current status of the implementation of these changes, as well as the status of the application deployment in the test-bed.

Jointly with the first version of extended middleware that is described in Deliverable D3.2 the applications are ready to perform first experiments in the Phosphorus test-bed. The plans for the experiments in the first testing phase scheduled for project month 13 and 14 are also sketched in the report. These achievements are in line with the work plan of the project.

The WISDOM use-case consists of virtual screening techniques, computing compounds of large-scale molecular dockings on targets implicated in diseases like malaria. The goal in Phosphorus is to implement the WISDOM workflow in UNICORE in order to improve the reliability in comparison to previous experiments. For that purpose the workflow has been analysed, the software deployed and tested at 4 different local Phosphorus test-beds. UNICORE 6 enabled executions and workflows are planned as first test-bed experiments.

The KoDaVis application provides distributed collaborative visualizations of huge atmospheric data-sets. The software has been enhanced to use UNICORE to manage its resources and provide access to the test-bed functionality. It has been deployed in 3 local Phosphorus test-beds. Experiments to test the functionality and performance of the test-bed as well as a demonstration at SC 2007 are planned for the next months.

TOPS (Technology for Optical Pixel Streaming) is a system that enables remote viewing of large scientific datasets on high resolution display devices (Tiled Panel Displays). These displays typically have 30 to 100 million pixels. TOPS streams these pixels, uncompressed, from the data center to remote displays. For this, high bandwidth networks are required. TOPS will be adapted to be able to utilize user/application controlled lambdas for the visualization service. Datasets viewed with TOPS are typically gigapixel scientific visualizations, from many different fields of science, among which climatology and biology are important application drivers.

Distributed Data Storage System (DDSS) provides two use-cases, one based on GridFTP and one based on Backup-Restore/Archive applications. In both use-cases, the applications have been enhanced to interact directly with the network resource reservation systems. The software has been deployed at 4 sites in different local Phosphorus test-beds and will execute different scenarios including connection reservations in the first test-bed experiments.

1 Introduction

The Phosphorus project will provide solutions for some of the key technical challenges to enable on-demand e2e network services across multiple domains. It will provide applications with a consistent view of all Grid resources, including compute, storage and network resources. These concepts are evaluated and demonstrated in a test-bed. The objective for the applications in the Work Package 3: Middleware and Applications Work is to demonstrate and evaluate the functionality provided by the other work packages.

For that purpose, existing distributed applications with high communication requirements regarding bandwidth and/or QoS are being extended to utilise the capabilities of the Phosphorus test-bed. The general approach is as follows:

- For each application, use-cases have been defined that shall be executed in the test-bed. Based on these use-cases, requirements towards middleware and the other work packages have been developed.
- Required design changes and extensions of the applications and the underlying middleware have been derived from the use-cases.

These initial steps have been documented in the previous Deliverable D3.1 [WP3-USE-CASES]. The current Deliverable D3.3 describes the status of the implementation of the designed changes, the status of the application deployment in the test-bed and the test-bed experiments planned for the first test phase that is scheduled for months 13 and 14.

It should be noted that the related Deliverable D3.2 [WP3-MIDDLEWARE] is describing the status of the middleware implementation and deployment. The applications are accessing the network services offered by Phosphorus either via this middleware (KoDaVis and WISDOM) or directly via the interfaces that have been defined jointly with Work Package 1 and which are documented in Deliverable D1.1 Requirements and specifications of interfaces and architecture for interoperability between NRPS, GMPLS, Middleware [WP1-SPECS] (TOPS and DDSS).

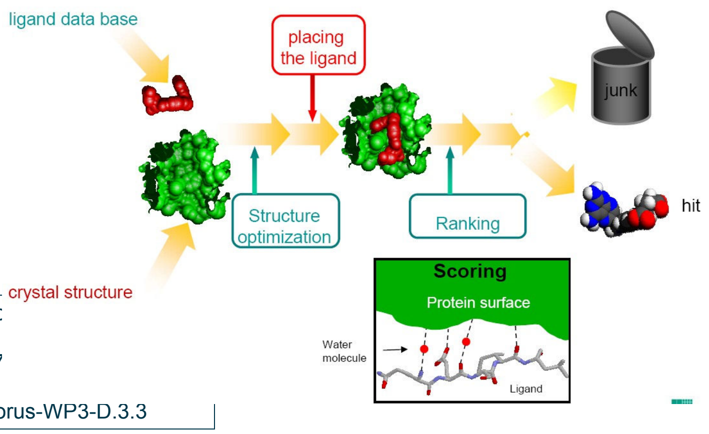
The deployment of applications and middleware in the test-bed has been performed in close cooperation with WP6 and has also been documented in the Deliverables D6.1 Test-bed design [WP6-DESIGN], D6.2 Global test-bed [WP6-GLOBAL].

The rest of the document is organised as follows. In chapter 2, the main chapter of this Deliverable, a section for each application starts with a summary of the planned design changes and extensions of the application to support the use-cases as defined in Deliverable D3.1. It is followed by a description of the current status of the implementation and the deployment in the local test-beds. Each application section ends with a description of the test-bed experiments planned for the next months. Chapter 3 provides conclusions.

2 Application design changes and extensions

2.1 WISDOM

The use case WISDOM consists of virtual screening techniques including two of the most popular protein docking simulation tools, AutoDock and FlexX. Both are in silico docking techniques, where researchers use large computing power, e.g. provided through Grid systems, to compute the probability that potential drugs will dock with a target protein [WISDOM2007]. This allows researchers to rule out the vast majority of potential drugs, so that they can concentrate on the most promising compounds in wet-laboratory tests.



This speeds up the screening process and reduces the cost of new drug development to treat diseases such as malaria. Malaria together with many other tropical and protozoan diseases is one of the most neglected diseases by the developed countries as well as by pharmaceutical industries [WISDOM2006, WISDOM-IEEE].

2.1.1 Planned design changes and extensions

One of our experiences made during the EGEE WISDOM data challenge [WISDOM-EGEE, WISDOM-Grid] was, that both the distribution of input data, and especially the transfer of result data of the millions of docking processes from the participating sites back to the user's site were complex, cumbersome and therefore resulting in significant data losses. As a consequence by this we will concentrate in Phosphorus on two aspects: 1) the test of network and middleware functionalities and 2) the implementation of a perfect workflow for WISDOM. Thus WISDOM will be mainly used to implement and test UNICORE 6-enabled workflow jobs with stage-in, execution (run) and stage-out phases. So, we are concentrating on automation of the WISDOM workflow, which will be executed in three phases:

The WISDOM stage-in phase consists of licensing (FlexX) 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 (**Figure 2.1**).

Based on the experiences in EGEE, we need for the WISDOM execution phase a good job control and allowing job resubmission in case of job failure.

The WISDOM stage-out phase includes the transfer of the local output data (FlexX Mol2 files, AutoDock DLG and Mol2 files) after termination to a specified output server into a directory hierarchy or a database.

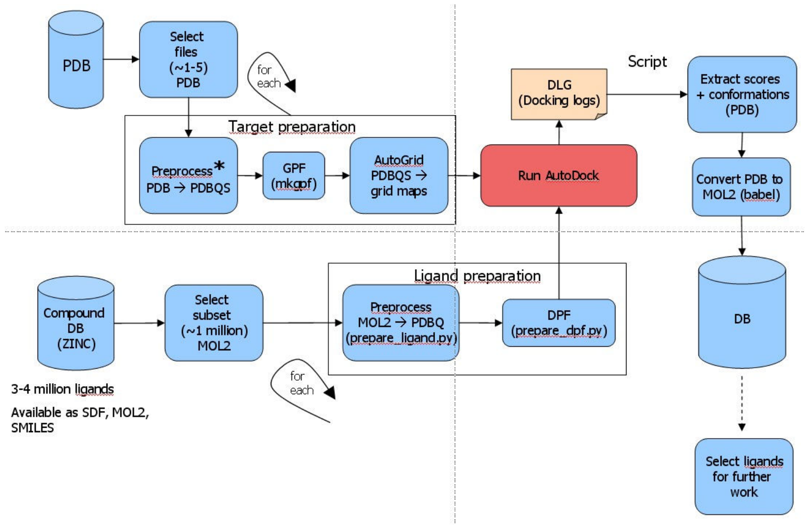


Figure 2.1: Typical workflow of the protein docking simulation tool AutoDock

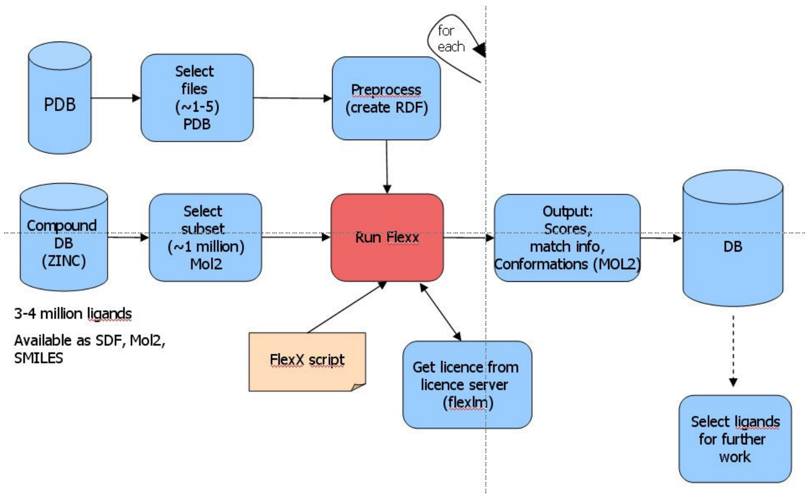


Figure 2.2: Typical workflow of the alternative protein docking simulation tool FlexX

Both workflow graphics show on the left side the preparation of input data (pre-processing) and includes the distribution of the needed data (stage-in phase). FlexX includes in this phase a license check and afterwards the execution of the WISDOM docking applications can be started. On the right side different post-processing steps are shown. Before this can be done, the transfer of output data to a specified output server (stage-out phase) must be performed.

In the EGEE project WISDOM was used to perform a huge docking scenario with 46 million dockings on 26 computing sites (~ 2.000 CPUs) in 15 countries. Due to the size of the Phosphorus testbed it is not intended to perform a similar big docking scenario. However, focusing on a real promising protein database (PDB) subset, a successful virtual screening execution finding real drug candidates can be achieved, using the PHOPHORUS testbed too.

2.1.2 Status of implementation and deployment

WISDOM applications AutoDock and FlexX are installed on the computing sites

- Fraunhofer Sankt Augustin (PACK cluster, i686/386),
- Supercomputing Center Poznan (Itanium cluster, ia64),
- Research Center Juelich (Cray XD-1, X86_64), and
- University of Essex (WISDOM cluster, i686).

Local tests are done 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 availability of UNICORE 6 installations first Grid-enabled WISDOM executions can be started.

2.1.3 Planned test-bed experiments

In following weeks WISDOM will be used to realize and test the implementation of

- UNICORE 6-enabled WISDOM executions on the Phosphorus Grid
- UNICORE 6-enabled WISDOM workflows, tested in different steps
  - Job executions with pre-distributed data sets
  - Job executions with stage-in processing
  - Job executions with stage-in and stage-out processing
- WISDOM protein docking, processing a real promising protein database subset of ZINC (PDB)

2.2 KoDaVis

In many disciplines of science and engineering, researchers have to handle huge amounts of data, originating from experiments or computer simulations. Interactive visualisation is often an important tool to explore this data. KoDaVis is a distributed, collaborative visualisation-system which provides remote access to huge atmospheric simulation data originating from the simulation of the transport of chemical tracers in the troposphere. As the size of a single dataset is up to about 1 TByte, it is not feasible to transfer the whole set to the local visualisation hardware, especially when affordable, PC-based systems are used. Instead the data is stored a central data server or at the supercomputer where the simulation was done. Slices of the data are transferred to the client on-demand. As described in Deliverable D3.1, two application use-cases are supported in KoDaVis: a single scientist, performing interactive visual analysis of remote data (use-case 1) and scientists at two or more different sites, collaboratively exploring the data.

Although, in use-case 2. the visualization clients may be implemented using different visualisation environments, a synchronization service ensures that all share the same view on the data. For reliable fluent visualization sessions, the reservation of bandwidth between the data server and each of the clients is required. The main goal within Phosphorus is to enable KoDaVis to perform such reservations using the UNICORE middleware [UNICORE, U@SOURCEFORGE].

2.2.1 Planned design changes and extensions

In order to support the use-cases sketched above, the pre-existing version of KoDaVis needs significant extensions. This software consists of the following components:

- visualization applications,
- a parallel data-server that distributes fragments of data selected by the clients,
- a collaboration server that synchronizes the clients, and
- an optional control GUI that monitors client activity and interacts with the ARGON system to handle immediate network connection requests.
- All of these parts have to be started separately, that is, the users have to agree on session parameters allowing to connect to each other, have to log into the machines that are involved and start the server and client components manually. Afterwards, they have to establish the desired connections between the components.

The main design change of the KoDaVis software is to enhance it with respect to resource reservation and access control by adapting it to the Grid middleware UNICORE as outlined in **Figure 2.3**. In this figure, components that have to be developed or adapted have been coloured, the green components are already available, the red ones will be implemented in the forthcoming months. A UNICORE client gridbean that is especially developed and tailored for KoDaVis, will offer an interface for resource specification and control of the data- and collaboration-servers. Thus, the user just has to start the UNICORE client and select the required resources, such as bandwidth, start time and duration of the visualization session, data- and collaboration-servers. Finally, he submits the request to the MetaScheduler System (MSS) via UNICORE gateway. The MSS then performs the required negotiations and reservations. At the reserved starting time, all reserved resources are allocated and the visualisation application is started. Moreover, the required exchange of bind information (IP-address) is handled by the middleware. Additional users aiming to join a collaborative session will be able to use the middleware to query existing sessions and connect to them.

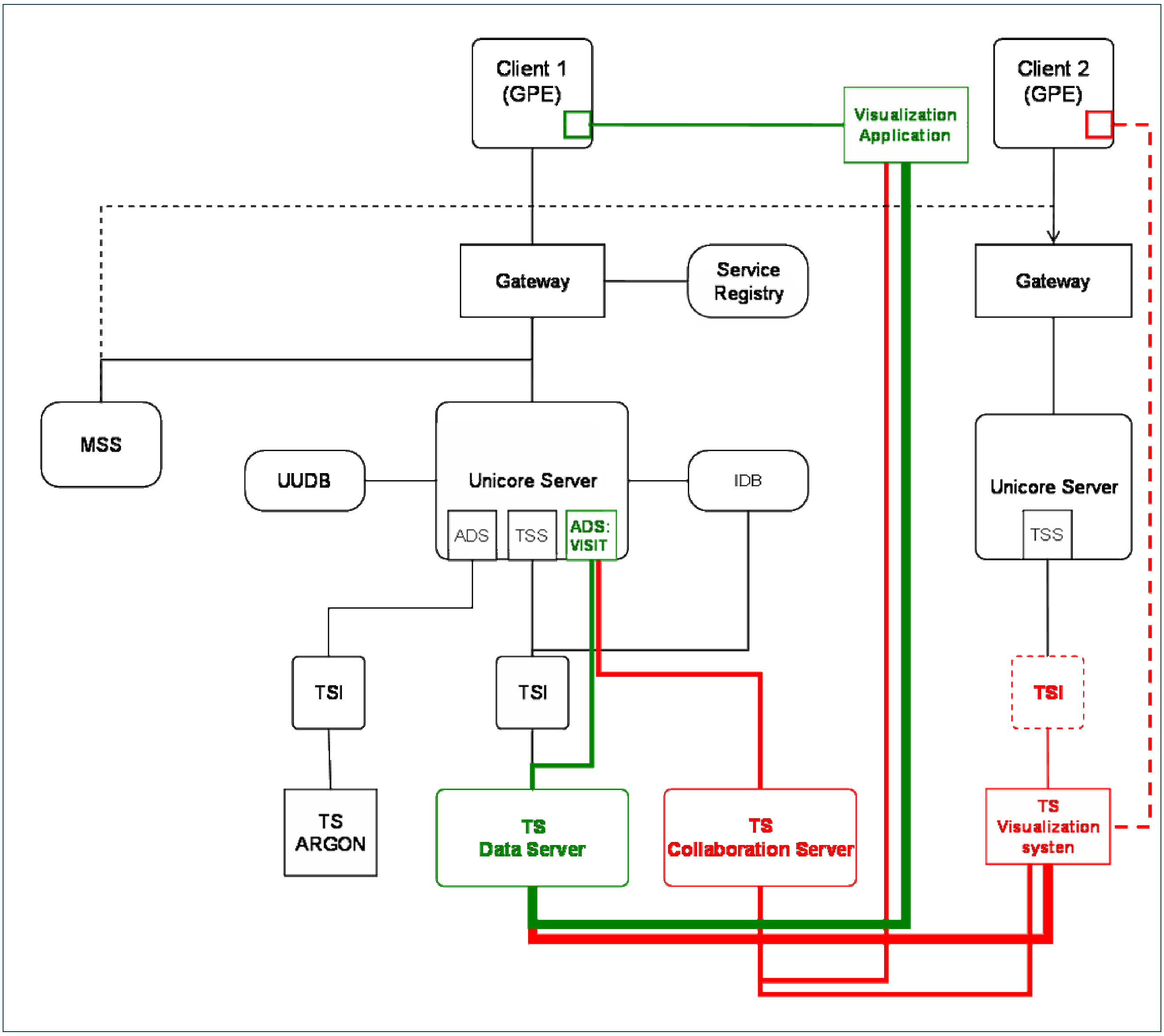


Figure 2.3: Design of the planned extensions of the KoDaVis software, green components have been developed up to now, red components will be implemented in the next months.

2.2.2 Status of implementation and deployment

A decision that was taken at an early stage of the project was to use the new Web-Service based version 6 of the UNICORE middleware (see Deliverable D.3.1). It allowed us to modify the Data and Collaboration servers such that they act as Web Services [WS-RF], thus supporting OGF and OASIS standards [OGF, OASIS]. It should be mentioned that only startup and management of these servers are based on Web Services. For the data-transfers that require high throughput and low latency, the normal transport mechanisms of the underlying communication library VISIT are used.

In the implementation of the design changes defined in Deliverable D3.1 and illustrated in **Figure 2.3** we have started with those components that are required to support use-case 1. The design changes described above are illustrated in Figure 2.3. The following components, drawn in green in the figure, have already been implemented:

- A VISIT/KoDaVis gridbean for the UNICORE GPE application client has been developed. This gridbean offers a seamless and intuitive interface for specification of the above mentioned resources and control of the data-server. Furthermore, it is possible to administer users participating in the particular visualization session and deploy certain privileges among them.
- The VTK/Qt based visualisation client has been extended to receive the parameters required to establish the communication to the server components from the UNICORE client. Figure 2.4 shows a screenshot of this client.
- An additional Web Service Resource Framework (WSRF) based VISIT/KoDaVis Grid service (ADS) for the UNICORE server has been developed. This service starts the data-server Web-Service. It consists of the actual data-server and a wrapper providing the Web-Service interfaces to the outside world. Furthermore, it manages all exchange of status information and commands between the UNICORE clients and the data and collaboration servers.
- The data- and collaboration-servers have been extended to support a common XML-based control protocol. This protocol is used to communicate with the Web-Service wrappers and allow them to be managed from the UNIOCRE client.

The next step is the implementation of the components needed to support also use-case 2, namely the interface between the UNICORE client and server components and the collaboration server. The connection to the MetaScheduling Service will be implemented as soon as its integration into UNICORE 6 is completed (see Deliverable D3.2).

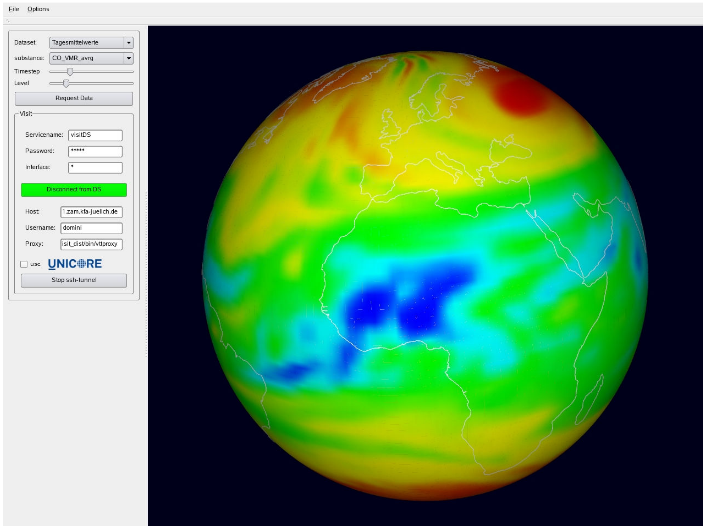


Figure 2.4: Screenshot of the KoDaVis visualisation client.

2.2.3 Planned test-bed experiments

The current version of the software has been deployed in the test-bed as planned:

- The data and collaboration servers are installed on the Cray X-D1 at FZJ,
- The visualization clients are installed on workstations at FZJ, PSNC and UoEssex.

Initial functionality tests have already been performed using the internet. Due to firewall restrictions for the connection of the partner sites to the internet, these tests have used ssh-tunnels for the data-transfers.

In the global test-bed, we plan to test the following functionalities:

- Connectivity of the various software components over the test-bed connections. For connection reservation, the Web-tools provided by other work packages will be used.
- Performance of the application over the test-bed connections. It will be tested if the reserved bandwidth is actually available and can be harnessed by the application.
- The overall stability and usability of the components developed so far. It will be tested if the software is running stable and with reliable performance under the conditions found in the test-bed according to stability, latency and bandwidth.

Besides this, KoDaVis will be demonstrated on the PSNC booth at SC 2007 in Reno. This test will also include connection reservation between FZJ and PSNC over the global Phosphorus test-bed.

2.3 TOPS

As stated in the OptIPuter [TOPS-1] principle, the bandwidth of modern networks exceeds the bandwidth of the internal bus of a traditional computer. If we ignore latency, it is possible to geographically distribute the computer so that disks are in Tokyo, graphics card in Amsterdam and display in New York.

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 refering 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.

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). TOPS streams these pixels, uncompressed, from the data center to remote displays. Network allocation still is a manual process, with many emails between network engineers and application developers to run a distributed application. Part of the planned work in Phosphorus is to automate this tasks by using the Phosphorus network reservation services developed by WP1 and WP2.

For TOPS the data and the rendering machine are located in Amsterdam, the result will be streamed over the network to Sankt Augustin an 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 graphis cards and a gigabit network interface.

2.3.1 Planned design changes and extensions

In discussions with SURFnet (WG-1) we understand that the interface to the NRPS has been designed but is not fully implemented yet. SARA and SURFnet are setting up a DRAC service in both SURFnet6 and in the Phosphorus testbed. As a first step we will implement in TOPS a web services client that will interface to the

DRAC web service, in order to reserve a lightpath between the streaming application server and the clients at the Tiled Panel.

Currently two versions of TOPS are implemented, one for static images and one for synchronized streaming of ultra high resolution video to the Tiled Panel. Integration of the two versions is foreseen. Based on experience during numerous demonstration sessions and the usage of TOPS in the Dutch Research environment, an improvement and extension of the user interface is desirable, but to a large extend outside the scope of the work defined in Annex I (Description of Work) the use-case document [WP3-USE-CASES]. Currently an inventory is being made, what part can be included in Phosphorus. Other resources may become available to implement these improvements as resources in Phosphorus for this work are limited.

2.3.2 Status of implementation and deployment

At IAIS we have set up a test machine within the viola testbed. SARA installed the TOPs software and did first tests using the available internet. Several parametes on the test machine could be modified to get the system runing in a fast way. Indepth testing was not possible as part of the network is out of control of the project. For the application i.e. the MTU parameter of the network device has to be set to jumbo frames, but these packets got lost, packets of 1500 bytes came through.

With this first test also issues on security policies came up very soon. Following FhG security rules the display machine which is accessable form the outside network must not contain any internal data. The cluster machines that are within the internal FhG IAIS network and which contain internal data could not be used for a remote display concept. The projectors can not be connected to the network directly, the signal for the projectors is produced by a graphics workstation with access to internal data.

Therefore we have set up a cluster of four machines that will be used for the further tests. These are two HP xw8000 and two HP xw600 with Quadro FX3000 and Gigabit network interface. The machines are connected to a gigabit switch which is connected to the Phosphorus network. Test will start soon via a 1GB connection between SARA and FhG IAIS, the network setup is described in **Figure 2.5**.

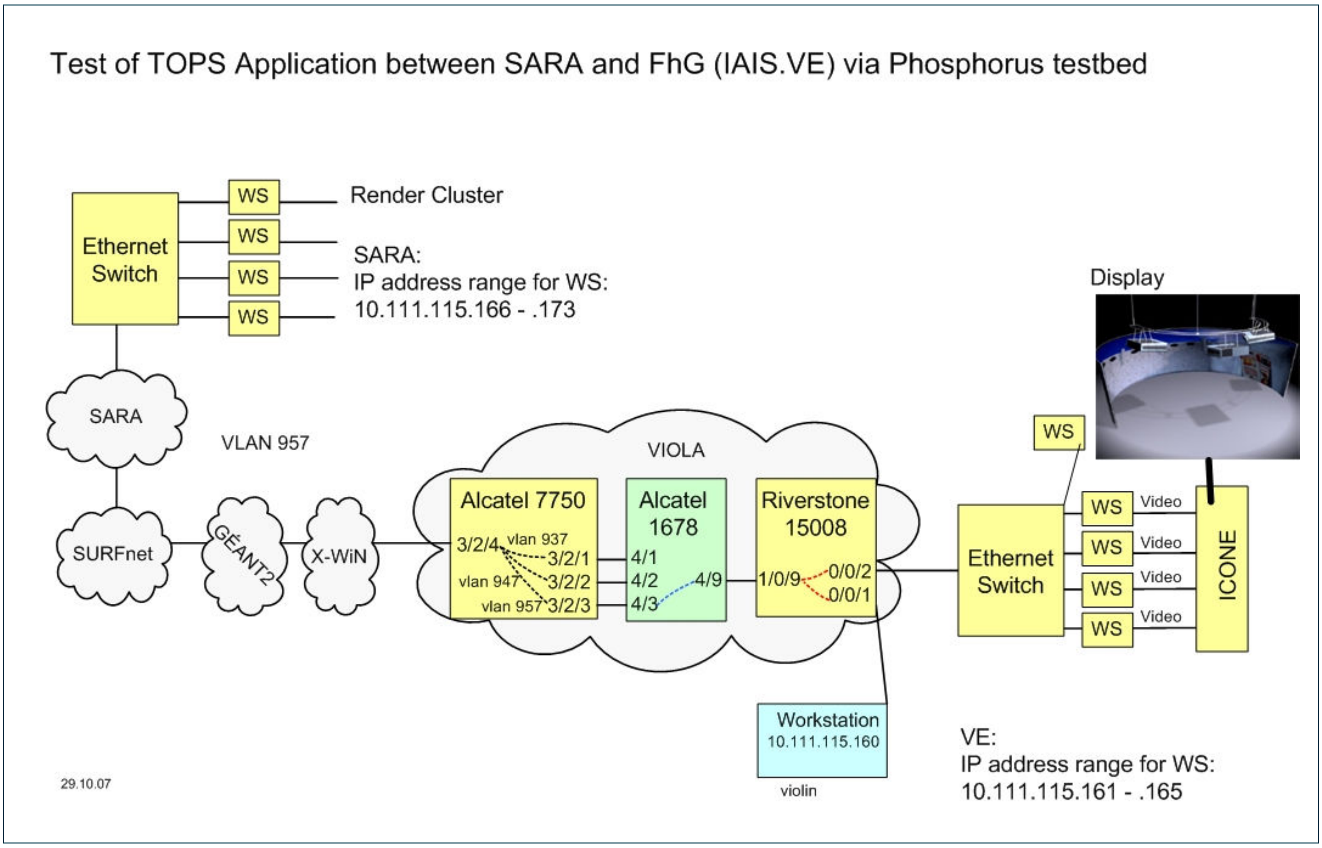


Figure 2.5: Test of TOPS Application between SARA and FhG

2.3.3 Planned test-bed experiments

Until now the tests between SARA and FhG have been executed over the routed internet. In the coming weeks we will start using a lightpath of the Phosphorus network, between SARA and FhG, The next step is implementing and testing a simple web services interface in TOPS to control DRAC in the Phosphorus testbed.

In addition to this, TOPS will be demonstrated by SARA at the Dutch Research Consortium booth at SC2007 in Reno.

2.4 DDSS

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] used for large data transfers between Grid sites and backup/archive application used across Grid sites. GridFTP is the high performance, secure, reliable client-server data transfer protocol optimized for high-bandwidth wide-area IP networks. Single GridFTP client can collaborate with one or many servers at the same time, transmitting the data in multiple parallel streams (stripes). Typically, GridFTP is used by the Grid end-users in order to transport the data to/from remote computing systems (e.g. input/output data in Grid systems). It is also used by higher-level Grid systems and applications as lower-layer data services for moving the data between system nodes.

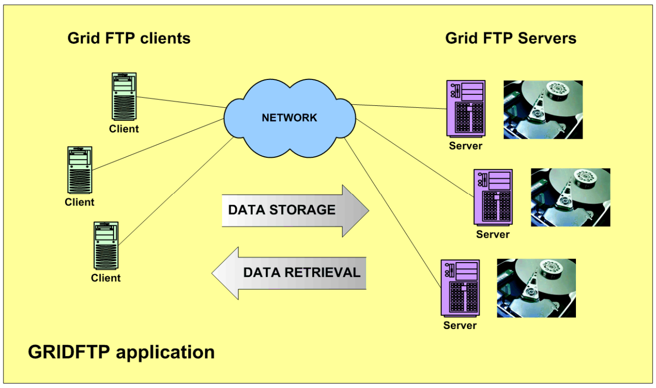


Figure 2.6: DDSS GridFTP application servers and clients

Backup/archive application (B/A) is used for performing automatic backups and/or archive copies of data that are originally stored in the Grid nodes or other end-user nodes. B/A clients collect the backup/archive data and send them using TCP/IP streams to B/A server. B/A server stores user data in disk, tape or other media pools, according to defined policy. Backup/archive copies are performed manually or automatically, according to some typical schemes: full, incremental or cumulative copy. Typical configuration contains single B/A server and multiple B/A clients. Typical users of B/A application are Grid nodes administrators or Grid end-users. In the Phosphorus project, a commercial version of the application is examined [TSMBA].

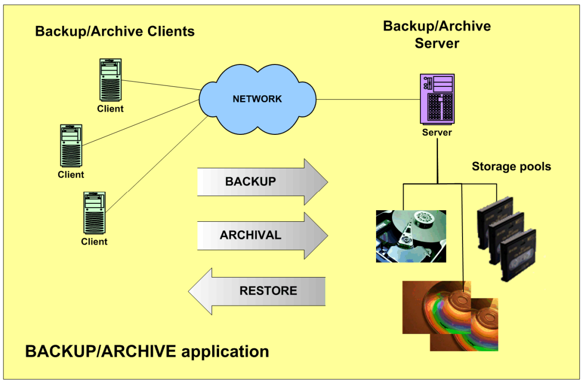


Figure 2.7: DDSS Backup/Archive application servers and clients

2.4.1 Planned design changes and extensions

The main goal of the work that has been performed for purposes of the Phosphorus testbed was to enable both DDSS applications to perform reservations of bandwidth and/or guarantee low latency for one-to-one and/or one-to-many data transfers. The bandwidth and latency guarantees are important in order to make Grid FTP data movements and B/A data copies effective and robust. Both DDSS applications have no native, built-in interface to any resource reservation systems.

Grid FTP application is tied to Globus Toolkit middleware [Globus]. However, the only kind resources that is demanded by the application is the network link. Therefore the network resources will be reserved by the Grid FTP clients using Network Resource Provisioning System (NRPS) application interface (API). The modification of the application include adding functions for interfacing with NRPS to the GridFTP client source code, and adding invocations of reserve/release functions in appropriate points of the client source code. Resource reservation requests are generated by the application each time the Grid FTP data connection is created. No bandwidth and latency reservation is made for a control connection.

DDSS/Backup/Archive application is not tied to any Grid middleware. Similarly to DDSS/GridFTP the only requested kind of resource is a network link. Therefore NRPS is used for resources reservation. The application chosen for the test-bed, IBM Tivoli Storage Manager Backup/Archive [TSMBA], is a closed-source application, therefore no code modifiation was possible. Instead, the application have been wrapped by a Unix shell script, that calls an executable file prepared by us before and after running the actual TSM B/A application. This executable file contain the NRPS interface and sends resource reservation and release requests to NRPS.

The functionalities contained in resource reservation code added to Grid FTP client and in the executable file invoked before and after TSM B/A tests include: advance and on-demand reservation of bandwidth and latency as well as a coordinated link establishment/reservation for many-to-one application use cases. These reservations are essential for a reliable performance of the data transfers of the GridFTP and TSM B/A applications.

In addition to application modifications, PSNC developed a tool for automation of application tests, e.g. automatic, periodic initiation of test use-case, and a tool for results collection and reporting in a visual form. This tool is used during both application tests. The performance data collected using this tool are put on the web site in a form of graphs, and can be used for later analysis of network and reservation mechanisms performance.

2.4.2 Status of implementation and deployment

Two use cases are to be tested in PHOSHORUS testbed: GridFTP data transfers and backup/archive/restore operations. In a former scenario GridFTP server stores or retrieves the data from/to single or multiple clients (many-to-one). The single client can collaborate with many servers at the same time (one-to-many). Data transmission between client-server pair can be done over 1-128 parallel streams (stripes). In the latter scenario, the end-user data are collected on client machines by a client B/A modules and sent through TCP/IP connections to B/A server module. Typical setup contain single B/A server and multiple B/A clients.

The functionality already available in the testbed and verified by preliminary tests include one-to-one Grid FTP scenarios (one client, one server) tested for various number of threads (1-16) and one-to-one backup/archive/restore application scenarios. Test applications are deployed in majority of partners' sites. DDSS Grid FTP is tested in PSNC, FZJ, Fraunhofer and ESSEX. DDSS Backup/Archive application is tested between PSNC and FZJ.

Application modifications are already made according to application design changes presented in the previous subsection. They are implemented by putting additional code to the source code of DDSS/GridFTP application clients and by developing a wrapper for DDSS/Backup/Archive clients.

2.4.3 Planned test-bed experiments

As mentioned in a previous point one-to-one scenarios are already deployed in the partners' sites and their operability is verified. The scenarios to be deployed in following months include more complicated DDSS test cases, including many-to-one and one-to-many use-cases.

Planned test-bed experiments include further testing of one-to-one scenarios for Grid FTP and Backup/Archive applications as well as testing many-to-one and one-to-many setups of these applications.

After further development of test automation mechanisms, scenarios will be run constantly, periodically or on-demand, and their results will be collected in a visual form on the testbed web site.

3 Conclusions

This document has described the status of the changes and implementation of the first version of the Phosphorus test applications. These applications are based on pre-existing applications and are being extended to support various use-cases that show-case the functionality available in the Phosphorus test-bed.

The progress of the application-related work within Work Package 3 is in line with the work plan:

- The first versions of the applications with enhancements and modifications for the distributed execution in the test-bed have been developed as defined in the use-case document D3.1 [WP3-USE-CASES]. They are ready to access and use the network reservation capabilities of Phosphorus, either directly via the NRPS [WP1-SPECS] or indirectly via Grid middleware [WP3-MIDDLEWARE].
- The first versions of the applications have been deployed in various local test-beds and are ready for tests in the global Phosphorus test-bed in the next months [WP6-DESIGN, WP6-GLOBAL].

Internet-based functionality tests of the applications have shown that the applications suffer from bandwidth limitations and lack of QoS of the Internet. The benefit of the test-bed capabilities will be evaluated and demonstrated in the forthcoming global test-bed experiments that are scheduled for project month 13 and 14.

4 References

[WP1-SPECS]	Phosphorus Deliverable D1.1: Requirements and specifications of interfaces and architecture for interoperability between NRPS, GMPLS, Middleware
[WP3-USE-CASES]	Phosphorus Deliverable D3.1: Use-cases, requirements and design changes and extensions of the applications and middleware
[WP3-MIDDLEWARE]	Phosphorus Deliverable D3.2: Report on middleware extensions and implementation
[WP6-DESIGN]	Phosphorus Deliverable D6.1 Test-bed design
[WP6-GLOBAL]	Phosphorus Deliverable D6.2: Global test-bed
[WISDOM2007]	Antje Wolf, Marc Zimmermann, and Martin Hofmann-Apitius. Alternative to Consensus Scoring-A New Approach Toward the Qualitative Combination of Docking Algorithms J. Chem. Inf. Model.; 2007; 47(3) pp 1036 - 1044; (Article) DOI: 10.1021/ci6004965
[WISDOM2006]	Lyn-Marie Birkholtz, Olivier Bastien, Gordon Wells, Delphine Grando, Fourie Joubert, Vinod Kasam, Marc Zimmermann, Philippe Ortet, Nicolas Jacq, Nadia Saidani, Sylvaine Roy, Martin Hofmann-Apitius, Vincent Breton, Abraham I Louw and Eric Marechal, Integration and mining of malaria molecular, functional and pharmacological data: how far are we from a chemogenomic knowledge space? Malaria Journal 2006, 5:110 (17 Nov 2006)
[WISDOM-EGEE]	Vinod Kasam, Marc Zimmermann, Astrid Maaß, Horst Schwichtenberg, Antje Wolf, Nicolas Jacq, Vincent Breton, Martin Hofmann. Design of New Plasmepsin Inhibitors: A Virtual High Throughput Screening Approach On The EGEE Grid. submitted to Journal of Chemical Information and Modeling (JCIM)
[WISDOM-Grid]	Nicolas Jacq, Jean Salzemann, Yannick Legré, Matthieu Reichstadt, Florence Jacq, Marc Zimmermann, Astrid Maass, Mahendrakar Sridhar, Kasam Vinod-Kusam, Horst Schwichtenberg, Martin Hofmann and Vincent Breton, Demonstration of In Silico Docking at a Large Scale on Grid Infrastructure in: Challenges and Opportunities of HealthGrids; Proceedings of HealthGrid 2006; Part III: Bioinformatics on the Grid [2006]
[WISDOM-IEEE]	Breton, V., Jacq, N., Kasam, V., Hofmann-Apitius, M., Grid Added Value to Address Malaria, IEEE Transactions on Information Technology for Biomedicine, 11 (2007) doi: 10.1109/TITB.2007.895930.
[OASIS]	http://www.oasis-open.org
[OGF]	http://www.ogf.org
[U@SOURCEFORGE]	http://sourceforge.net/projects/unicore/
[UNICORE]	http://www.unicore.eu

[WS-RF]	http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf#announcements
[Globus]	http://www.globus.org
[GridFTP]	http://www.globus.org/grid_software/data/gridftp.php
[TSMBA]	http://www-306.ibm.com/software/tivoli/products/storage-mgr/
[**TOPS-1]**	www.optiputer.net
[TOPS-2]	www.sara.nl/TOPS
[TOPS-3]	Building a 100 Mpixel graphics device for the OptIPuter; Bram Stolk and Paul Wielinga Future Generation Computer Systems, Volume 22, Issue 8, October 2006, Pages 972-975

5 Acronyms

AAA	Authentication, Authorisation, Accounting
DDSS	Distributed Data Storage Systems
e2e	end to end
EGEE	Enabling Grids for E-sciencE (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	Intelligent Network Caching Architecture
KoDaVis	Tool for Distributed Collaborative Visualisation
MSS	MetaScheduling Service
MTU	Maximum Transmission Unit
NREN	National Research and Education Network
NRPS	Network Resource Provisioning System
PoP	Point of Presence
QoS	Quality of Service
SNMP	Simple Network Management Protocol
TOPS	Technology for Optical Pixel-Streaming
TPD	Tiled Panel Display
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 Silicio Docking On Malaria