EXPReS

JRA 1: Future Arrays of Broadband Radio-Telescope on Internet Computing

Deliverable DJ 1.6
eVLBI – Grid Design Document

Task 2.1.1 Grid – VLBI Collaboration

Document Filename:	grid_vlbi_design.doc
Work package:	JRA1: Future Arrays of Broadband Radio-Telescope on Internet Computing
Partner(s):	PSNC, JIVE
Lead Partner:	PSNC
Document classification:	Public

Abstract: The purpose of this document is to present the design for the integration of VLBI experiments with the Grid environment. The current status of VLBI and eVLBI operations is presented, as well as the analysis of state-of-the-art Grid solutions from the domain of network security, data transportation and resource brokers.

Delivery Slip

	Name	Partner	Date	Signature
From	Marcin Okoń Dominik Stokłosa Tomasz Rajtar Norbert Meyer Damian Kaliszan Jan Węglarz Marcin Lawenda Szymon Trocha Marcin Garstka	PSNC		
Verified by	Huib Jan van Langevelde Ruud Oerlemans	JIVE		
Approved by				

Document Log

Version	Date	Summary of changes	Authors
0.1	19/07/2006	Draft version	Marcin Okoń, Dominik Stokłosa, Marcin Lawenda, Marcin Garstka, Szymon Trocha, Wojbor Bogacki
1.0	10/10/2006	Final version	Marcin Okoń,Dominik Stokłosa Tomasz Rajtar, Norbert Meyer Damian Kaliszan, Jan Węglarz Marcin Lawenda, Szymon Trocha, Marcin Garstka

Table of Contents

List of Figures

1 Introduction

1.1 Purpose

The purpose of this document is to present the design for the integration of VLBI experiments with the Grid environment. The current status of VLBI and eVLBI operations is presented, as well as the analysis of state-of-the-art Grid solutions from the domain of network security, data transportation and resource brokers. A separate section deals with the issues of monitoring the network status, QoS and Grid data routing. The main part of this document presents the proposed architecture for the development of the next-generation eVLBI system, together with the analysis of potential problems and design limitations.

2 Current state of VLBI and eVLBI operations

Networks of radio telescopes can be used to produce detailed radio images of stars and galaxies. The resolution of the images depends on the overall size of the network (the maximum separation between the telescopes) and the sensitivity depends on the total collecting area of all the telescopes involved and, crucially, the bandwidth of the connection between the telescopes. In this technique, called Very Long Baseline Interferometry (VLBI) the signals between all telescope pairs are combined in a data processor. Typically this processor needs to find the correlated response in 2-bit (4 level) sampled signals on every baseline pair with high resolution, and must be able to cope with data rates up to 1 Gbps per telescope.

This functionality has been implemented by constructing a massively parallel, purpose-built ‘supercomputer’ – usually referred to as a Data Processor or Correlator. A decade ago, a 20000 node PC cluster would have been required to match the processing power of the EVN MkIV Data Processor at JIVE.

The typical scenario of VLBI observation is depicted in the following diagram:

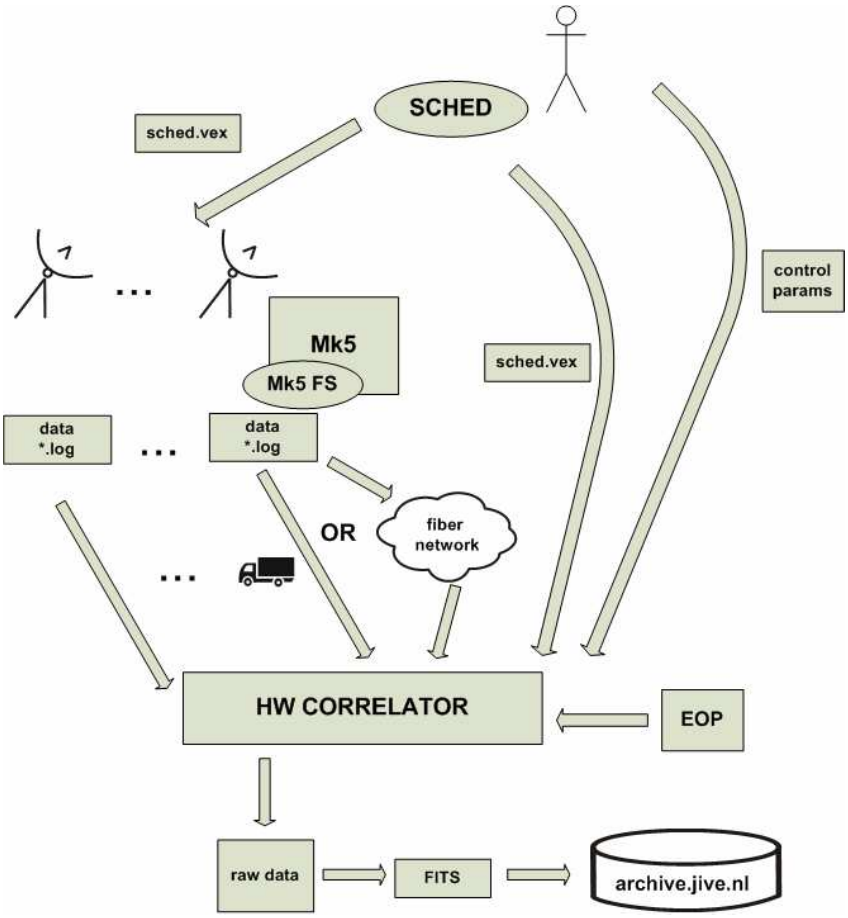


Figure 1. VLBI correlation process

The procedure is as follows:

- The radio astronomer (user) is given a research grant for performing the observation.

The exact time slot is scheduled and the information is returned to the user.

- The user uses the SCHED program to produce a single VEX file, which contains the complete description of the VLBI experiment. The file is of the ASCII text format. SCHED uses a text file as an input and has a command line interface and some graphical output.
- The VEX file is transferred to the radio telescopes, via an FTP system. The telescope operators are responsible for loading the file and setting up the observation.
- During the observation the Field System controls the telescope using settings obtained from the VEX file. Data is recorded in the Mk4 data format on Mk5A disks. These disks are inside a Mark5 computer and can only be accessed through special software. More information on these subjects can be found at the Haystack website: http://www.haystack.mit.edu/tech/vlbi/mark5/index.html.
- After the observation the collected data is sent to the central JIVE processor at Dwingeloo, via dedicated connections as is done for e-VLBI or still in most cases by physical transport of disks or tapes.
- In order to perform the correlation, the JIVE hardware correlator needs the following:
  - Data from all the telescopes
  - User VEX file
  - EOP (earth orientation parameters)
  - Additional control parameters, supplied by the user.
- After the correlation, the raw output data is verified by the JIVE staff, and if no errors are detected, the data is converted into the FITS format. This file format can be read by standard astronomical data processing software.
- Finally the converted correlation results are put in the JIVE archive where the radio astronomer who asked for the observation can access the data. After one year others will gain access to the data.

Currently JIVE can guarantee for regular science experiments e-VLBI at 128 Mbps for six radio-telescopes (Torun, Onsala, Westerbork, Medicina, Jodrell Bank, Cambridge). EVN aims at one regular e-VLBI session every 6 weeks. In practice this schedule may vary. Occasionally 256 Mbps for 6 and 512 Mbps for 3 radio-telescopes were achieved in research experiments. Also the Arecibo telescope has participated now and then at 32 Mbps.

For the data transfer e-VLBI relies on dedicated light-paths (Wb, Jb, Cam) and the classical IP switched network (Med, Ons, Tor).

In the future the guaranteed data rate has to increase and more telescopes have to join the e–VLBI network to make it more interesting for the astronomers to use it for their science experiments. Especially when the Effelsberg telescope (diameter 100 m) joins the sensitivity of the e-VLBI instrument will increase tremendously. .

3 Overview and analysis of the Grid resources

3.1 The Grid resource brokers

We are planning to design and develop a model of a distributed radio telescopes data correlation. As it is described in section 2, data from radio telescopes are sent via a regular mail to the Data Processor at Jive. In order to automate the process and make it faster we will transmit this data over the network and correlate using grid resources.

In this section we are providing a detailed analysis of the Grid Resource Brokers which could be used in our Grid – VLBI design. There are several factors we keep in mind during the analysis. These are collected in the summary below:

- *Basic resource broker functionality* – this contains a basic set of functions like job submitting, job monitoring, resource discovery, etc.
- *Integration with the Globus toolkit*
- *Web Service interfaces* – well-defined and documented API, which allows to control and interact with the broker from the low level
- *Open Source* - possibility to add new functionality to the resource broker is required, which implies that open source is a great advantage.

3.1.1 Condor-G

Condor – G	
Type	Fault-tolerant job submission system
Vendor	The University of Wisconsin-Madison (UW-Madison)
Licence	Condor Public Licence
Home Page	http://www.cs.wisc.edu/condor/
Compatible middleware	Globus Toolkit (2.4.x – 4.0.x); Unicore, NorduGrid

Condor has been developed at the University of Wisconsin-Madison, USA as an artefact of the Condor Research Project. It was first installed as a production system in the UW – Madison Department of Computer Science and nowadays it serves and manages more than 1000 workstations.

Condor is a specialized workload management system for compute-intensive jobs. It provides a job queuing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their serial or parallel jobs to Condor, Condor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion. Condor can be configured to design and build Grid computing environments.

Condor consists of two parts:

- job management part (keeps track of a user's jobs). Available operations: to show you the job queue, to submit new jobs to the system, to put jobs on hold, and to request information about jobs that have been completed
- resource management part: keeps track of which machines are available to run jobs, how the available machines should be utilized given all the users who want to run jobs on them, and when a machine is no longer available.

3.1.1.1 Licence

Condor has been published with a Condor public licence, which means that installations, use, reproduction, modification and redistribution of this software, with or without changes in source and binary formats are permitted.

In addition there are also two levels of support available: free through the Condor-users mailing list and fee-based support for Condor, which is mainly problem-oriented support.

3.1.1.2 Functionality

Condor-G is the job management part of Condor and it allows the user to treat the Grid as a local resource. It was integrated with the Globus Toolkit environment, which is used in order to start a job on a remote machine. There is also a set of command line tools that performs basic job management such as:

- Job submission, indicating an executable, input and output files, as well as various arguments
- Query for a job status
- Cancel a job
- Notifications about jobs and errors
- Complete history of a job (by a log file)

The following figure shows the interaction between Condor-G and Globus protocols.

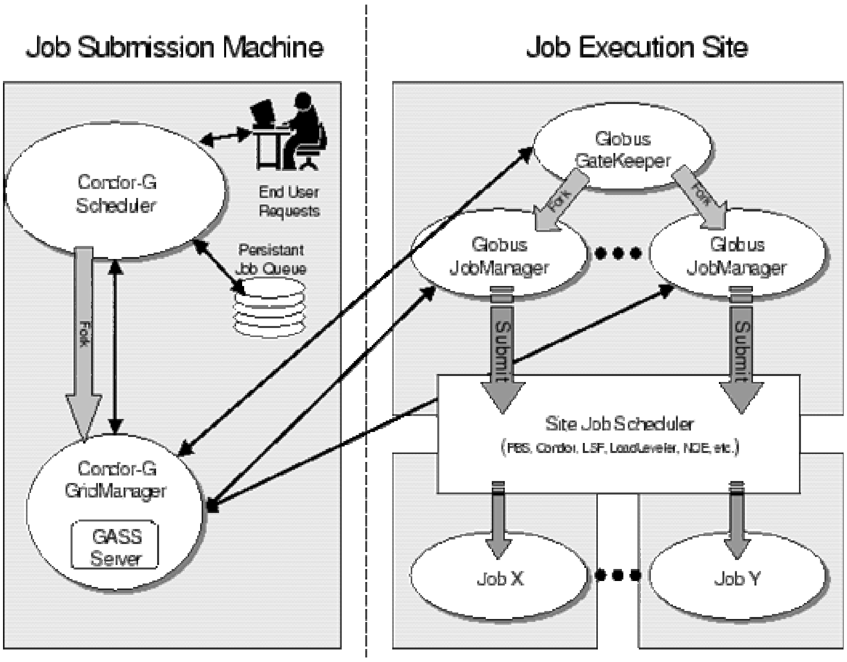


Figure 2. Remote job execution by the Condor-G

A GASS server is used to transfer the executable, input and output files to and from the remote execution site. Condor-G uses the GRAM protocol to contact Globus and request a new job manager as well as to monitor the job's progress. Moreover, the system is equipped with an intelligent fault-tolerant mechanism.

From the developer's point of view several interfaces have been defined to access Condor-G. One of them is Web Service API, additionally there is support for C, DRMAA API, Perl module and Grid ASCII helper protocol (GAHP). GAPH contains bindings for Java, C and Perl.

3.1.2 GRMS

GRMS	
Type	GridLab Resource Management System; open source meta-scheduling system
Vendor	Poznań Supercomputing and Networking Center; part of the GridLab Project
Licence	open source software developed under the GridLab Project.
Home Page	http://www.gridlab.org/WorkPackages/wp-9/releases.html
Compatible middleware	Globus Toolkit (2.4.x – 4.0.x)

The GridLab Resource Management System (GRMS) is an open source meta-scheduling system, developed under the GridLab Project, which allows developers to build and deploy resource management systems for large scale distributed computing infrastructures.

3.1.2.1 Licence

The GRMS is an open source software developed under the GridLab Project. The GridLab license applies to the current version of the GRMS software. The GridLab licence allows software to be used by anyone and for any purpose, without restriction. Installation, use, reproduction, display, modification and redistribution with or without modification, in source and binary forms are permitted.

3.1.2.2 Functionality

GRMS provides developers of user-level functionalities with a more abstract view of low level and complex Grid technologies. Based on dynamic resource selection and discovery, mapping and advanced scheduling methodology, combined with a feedback control architecture and support from other GridLab Middleware Services, it deals with dynamic Grid environment and resource management challenges, e.g. load-balancing among clusters and various work-load systems, remote job control or file staging support. Therefore, the main goal of GRMS is to manage the whole process of remote job submission and control to various queuing systems (e.g. Condor, PBS, LSF, N1 Grid Engine), clusters systems or resources directly.

GRMS has been designed as an independent set of components for resource management processes which can take advantage of various low-level Core Services, e.g. GRAM, GridFTP and the Mercury Grid Monitoring System, as well as various grid middleware services, e.g. GridLab Authorization Service, Replica Management Service, File Movement Service or Delphoi Services.

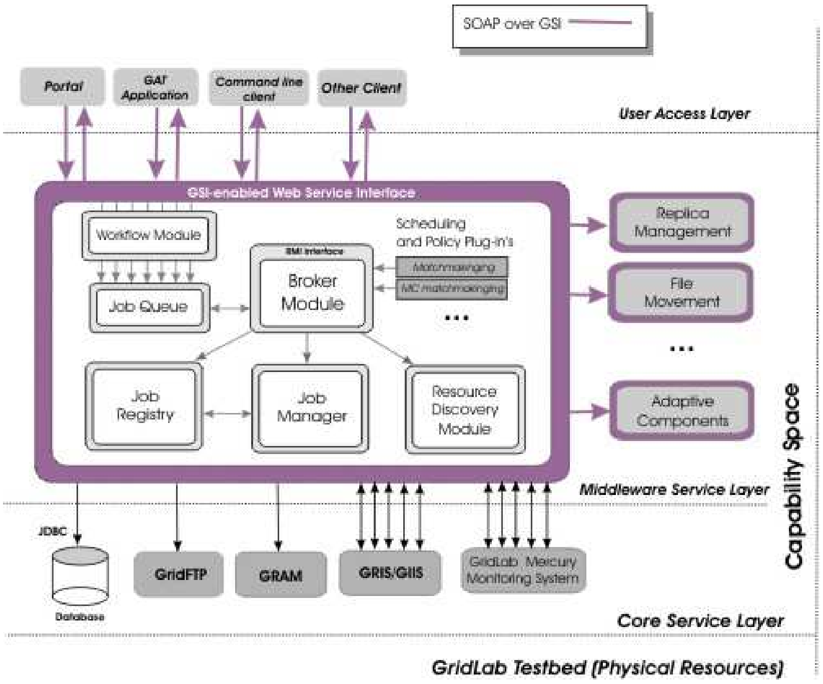


Figure 3. GRMS architecture

The architecture of the GRMS System is shown in the figure above.

According to authors, the GRMS system can be easily integrated with different grid middleware environment available on the market. Moreover, all the basic components of the GRMS systems have been written in Java.

3.1.3 GridWay

GridWay	
Type	Open-source component for meta-scheduling in the Grid Ecosystem
Vendor	Distributed Systems Architecture Group from Universidad Complutense de Madrid
Licence	Open Source Project
Home Page	http://www.gridway.org/
Compatible middleware	GridWay is Globus ProtoProject – adheres to the Globus philosophy

GridWay is an open-source meta-scheduler developed by the Distributed Systems Architecture Group from Universidad Complutense de Madrid, Spain. It allows unattended, reliable, and efficient execution of jobs, array jobs, or complex jobs on heterogeneous, dynamic and loosely-coupled Grids. GridWay performs all the job scheduling and submission steps transparently to the end user and adapts job execution to changing Grid conditions by providing fault recovery mechanisms, dynamic scheduling, migration on-request and opportunistic migration.

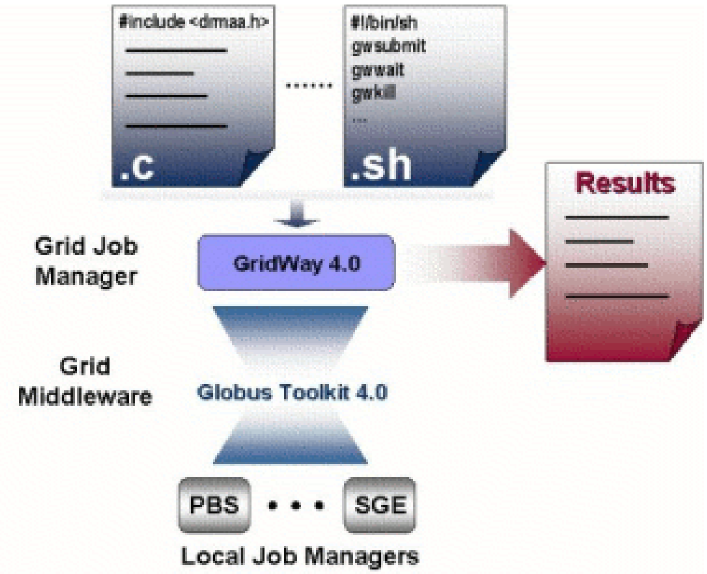


Figure 4. GridWay concept

GridWay was build on top of Globus services and it enables large-scale, secure, reliable and efficient sharing of computing resources. The system is capable of cooperation with different distributed resource management systems like Condor, LSF or PBS.

3.1.3.1 Licence

The GridWay project started in September 2002. The first releases of the metascheduler were developed for research purposes in adaptive and dynamic scheduling and were only distributed on request in a binary format. The first open source version was released in January 2005. The code is currently distributed under an Apache license and is freely available.

Since May 2006, the GridWay Metascheduler has been a Globus ProtoProject, so it adheres to Globus philosophy and guidelines for collaborative development.

3.1.3.2 Functionality

GridWay gives end users, application developers and managers of Globus infrastructures a scheduling functionality similar to that found on local DRM systems:

- *Advanced scheduling capabilities* on a grid consisting of Globus services: dynamic discovery & selection, opportunistic migration, performance slowdown detection, support for self-adaptive applications and support for the definition of new scheduling policies
- *Detection and recovery* from remote and local failure situations
- *DRM-like commands* to submit, monitor, synchronize and control single, array and interdependent jobs; monitor Globus resources and users; and extract Grid accounting information
- Full support for C and JAVA DRMAA GGF standard for the development of distributed applications on Globus services

- *Straightforward deployment* that does not require new services apart from those provided by the Globus Toolkit: MDS, GRAM, GridFTP and RFT
- *Modular architecture* that allows easy incorporation of new grid services and interoperability between different grid infrastructures (Globus WS, Globus pre-WS and EGEE)

3.1.4 Moab Grid Scheduler

Moab Grid Scheduler	
Type	An advance reservation-based grid/meta-scheduler
Vendor	Cluster Resources
Licence	Commercial, Proprietary licence
Home Page	http://www.clusterresources.com/
Compatible middleware	Moab Workload Manager ™

The Moab Grid metascheduler (code name => Silver) was developed by Cluster Resources Inc. Silver provides single point access to the distributed independently managed systems. Because of the advance resource reservation mechanism, the system can guarantee the start time of submitted jobs. Silver can manage the resources on any system where the Moab Workload Manager (a part of Moab Cluster Suite) is installed.

The Silver Grid Scheduler was created to allow organizations to better utilize their shared resources and provide a full suite of scheduling services in a very non-intrusive manner allowing local autonomy and transparent end-user usage.

3.1.4.1 Licence

The metascheduler is available on a proprietary license. A limited variant, called Maui Grid Cluster Scheduler, is available on a specific Open Source, non OSI-compliant, license. Both commercial and community support for Silver are available. There are two levels of commercial support: standard support and premier-account support.

3.1.4.2 Functionality

The metascheduler supports fine-grained grid-level fairness policies. Using these policies, the system manager may configure complex throttling rules, fairshare, a hierarchical prioritization and cooperation with allocation managers. The metascheduler and the related Moa Cluster Suite offer wide monitoring, diagnostic and statistical capabilities by means of the GUI tools and a Web-based user portal. One of the most interesting features of Silver is support for advanced reservations. This feature enables the use of scheduling techniques like backfill, deadline based scheduling, QoS support and grid scheduling.

- **Scalability** - Many grid systems have been designed to be infinitely scalable. Silver is not one of them. Silver has been designed to target the campus or enterprise level organizations, groups with up to 128 independently managed clusters. However, due to Silver's design, these clusters can be of any size, potentially supporting tens to hundreds of thousands of processors.

- **Minimal Software Stack** - Only components required are activated. Users do not need to change their behaviour in any way to effectively utilize distributed resources
- **Flexible Security Middleware** - Silver can operate with or without Globus
- **Flexible Data Management** - Data staging can be accomplished using site selected data management facilities
- **Flexible Account Management** - Account management can be handled using Globus, proxy accounts, simple mapping, or other approaches
- **Global Level Policies** - Specify prioritization, fairshare, usage limits, resource reservations, and other policies at the global or per cluster level
-

3.1.5 MP Synergy

MP Synergy	
Type	Job scheduler across heterogeneous distributed systems
Vendor	United Devices
Licence	Commercial, Proprietary licence
Home Page	http://www.ud.com/
Compatible middleware	Globus Tookit

MP Synergy, a product of United Devices, is designed for virtualized management of the entire enterprise infrastructure. It is a solution for managing job scheduling across heterogeneous, widely distributed enterprise resources.

3.1.5.1 Licence

MP Synergy is a commercial product released on a proprietary licence. Support and software updates are available only for the registered customers.

3.1.5.2 Functionality

The list of main features is presented below:

- **Heterogeneous meta-scheduling and management -** HPC Synergy can schedule jobs across multiple clusters and resource groups managed by different schedulers

- **Virtualized access to resources -** HPC Synergy provides an intuitive environment via a web portal and web service APIs, with jobs optimally executed at runtime. competencies. Automated workload balancing has also been introduced for improved productivity

- **Virtualized access to applications** – users can also submit through its portal, HPC Synergy provides views for submitting applications and following their internal progress.

- **Capture and reuse of computing best practices -** With HPC Synergy the computing processes need to be defined only once. Jobs can be tagged with business context data and archived in a dedicated data space for later search and reuse -- at any site and by any project.

3.1.6 EGEE Workload Manager Service (WMS)

Workload Manager	
Type	
Vendor	Part of EGEE (Enabling Grids for E-sciencE) project
Licence	Open Source Project
Home Page	http://public.eu-egee.org/
Compatible middleware	

The Workload Management System (WMS) comprises a set of Grid middleware components responsible for the distribution and management of tasks across Grid resources, in such a way that applications are conveniently, efficiently and effectively executed. The core component of the Workload Management System is the Workload Manager (WM), whose purpose is to accept and satisfy requests for job management coming from its clients. For a computation job there are two main types of request: submission and cancellation. In particular the meaning of the submission request is to pass the responsibility of the job to the WM. The WM will then pass the job to an appropriate CE for execution, taking into account the requirements and the preferences expressed in the job description. The decision on which resource should be used is the outcome of a matchmaking process between submission requests and available resources.

3.1.7 Grid Service Broker

Workload Manager	
Type	Metascheduler
Vendor	The Grid Computing and Distributed Systems (GRIDS) Laboratory, Australia

Licence	Open Source Project
Home Page	http://www.gridbus.org/broker/
Compatible middleware	Globus, Alchemi, Unicore

The Grid Service Broker is a metascheduler developed in the Grid Computing and Distributed Systems Laboratory at the University of Melbourne, Australia. It was created as a part of the Gridbus project and supports access to both computational and data grids. There have been a few successful installations in the academic community.

3.1.7.1 Licence

The Grid Service Broker is a part of a Gridbus project – a scientific research project -- and is available under the GNU Lesser General Public License (LGPL). There is also a mailing list support for users as well as developers.

3.1.7.2 Functionality

The Grid Service Broker mediates access to distributed resources by;

- discovering suitable data sources for a given analysis scenario,
- suitable computational resources,
- optimally mapping analysis jobs to resources,
- deploying and monitoring job execution on selected resources,
- accessing data from local or remote data source during job execution and
- collating and presenting results.

The broker supports a declarative and dynamic parametric programming model for creating grid applications. This model has been used in grid-enabling a high energy physics analysis application (Belle Analysis Software Framework). The broker has been used in deploying Belle experiment data analysis jobs on a grid testbed, called Belle Analysis Data Grid, having resources distributed across Australia interconnected through GrangeNet.

Gridbus can transparently access computational resources that are exposed by various low-level grid middleware solutions, such as Globus Toolkit. It supports scheduling systems such as Condor, PBS and Sun Grid Engine (SGE). Gridbus can interact with the scheduling systems, either with the help of Globus Toolkit or it can use remote SSH to launch jobs itself. The former approach to interaction relies on the services provided by the Globus Toolkit Pre-WebServices GRAM (Grid Resource and Allocation Management) interface. The Gridbus metascheduling solution improves the overall performance, scalability and robustness of computational grids of which it is a part.

Additionally, the design of the broker allows for writing a custom scheduler that implements a custom scheduling algorithm. Job-monitoring and status-reporting features are provided. Gridbus can gather the output of completed jobs and transfer it to user-defined locations. The user is capable of managing jobs by means of a command line interface or through Java portlets. The Gridbus Broker also features a WSRF-compliant service to allow access to most of the features of the broker through a WSRF interface.

3.1.8 PBS – Grid enabled PBS

Portable Batch System	
Type	Flexible batch queuing system developed for NASA
Vendor	Altair Engineering, USA
Licence	Free unsupported version, professional enterprise quality version
Home Page	http://www.openpbs.org/
Compatible middleware	

The OpenPBS is the original version of the Portable Batch System. The system was developed for NASA in the early to mid-1990s. It operates on networked, multi-platform UNIX environments.

3.1.8.1 Licence

Nowadays, the developers of the Portable Batch System offer two versions: OpenPBS, the unsupported older original version and PBS Pro, the enterprise-quality professional version

3.1.8.2 OpenPBS features

Below one will find a list features that, according to authors, made OpenPBS popular.

- Job priority: users are able to specify the priority of their jobs
- Jobs interdependency: the OpenPBS system has been equipped with the possibility of setting dependencies between batch jobs. Examples of such dependencies are execution order, synchronization, etc.
- Automatic file staging: this feature deals with the data transfer to and from the destination of the job execution
- Single and multiple queue support
- Multiple scheduling algorithm support

3.1.8.3 Grid enabled PBS – the PBS Globus interface

The PBS – Globus interface allows users to take advantage of Globus job facilities, while retaining the well known PBS interfaces like job monitoring, accounting, POSIX-compliance and others.

There are no additional requirements when a user wants to submit his/her job in the Grid. The only request is to tell the system that a job is meant to be run on Grid by setting - *Lsite=globus:resourcename* flag. The Job is treated as a regular PBS job, than it is sent to the PBS MOM - Globus by the PBS scheduler. This is the place where the mapping between PBS and the Grid (Resource Specification Language – RSL) takes place. The converted job description to the RSL can be submitted to the Globus using the Grid Resource Allocation and Management (GRAM). The system is taking care of required data transmission before as well as after job execution. When the job is in progress, normal PBS status commands can be used to monitor its status.

The great advantage of the Grid-enabled PBS is the possibility of accessing grid resources without the need of learning another interface.

3.1.9 Summary

In this chapter we have described several resource brokers. The aim of this section was to give a brief information about different resource brokers and their functionality. Following there is a summary table attached, which shows the basic features of all the resource brokers.

Features	Condor-G	GRMS	Grid Way	Silver	MP Synergy	Grid Service Broker	Open PBS
Open Source	Yes	Yes	Yes	No	No	Yes	(*)Yes
Globus Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Integration with Authorization Services	Yes	Yes	Yes	?	?	?	?
Interfaces for Web Service application	Yes	Yes	No ?	?	Yes	?	?
Command-line interface	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Java support	Yes	Yes	Yes	?	Yes	Yes	?

(*) – a simpler, currently not supported, version of the PBS is open source.

Our aim as a research project is to use open source tools and libraries whenever it is possible. According to this policy we are inclined not to take into consideration tools like Silver, PBS or MP Synergy. However, there are several other brokers which are worth being investigated further.

3.2 Grid Security

3.2.1 Introduction

Security is very important aspect in every distributed, grid-related system. But this “openness” of grid systems can cause many disturbances. On the one hand there is necessity to protect resources like computational servers or laboratory equipment from intruders on the other hand these facilities should be easily accessible to authorized users.

The very popular solution for developing distributed and heterogeneous systems which solves most of mentioned problems is Globus Toolkit (GT). GT is a set of tools which provides distinct WS and pre-WS authentication and authorization capabilities. Both build on the same base, namely standard X.509 end entity certificates and proxy certificates, which are used to identify persistent entities such as users and servers and to support the temporary delegation of privileges to other entities, respectively. Globus Toolkit in the confines of security aspects provides the following functionality:

- message-level security mechanisms - implement the WS-Security standard and the WS-SecureConversation specification to provide message protection for GT’s SOAP messages
- transport-level security mechanisms - uses transport-level security (TLS) mechanisms,
- authorization framework - allows for a variety of authorization schemes, including a “grid-mapfile” access control list, an access control list defined by a service, a custom authorization handler, and access to an authorization service via the SAML protocol.

For non-WS components, GT provides similar authentication, delegation, and authorization mechanisms, although with fewer authorization options.

3.2.2 Grid Security Infrastructure

The Grid Security Infrastructure (GSI) is used by the Globus Toolkit for enabling secure authentication and communication in distributed and heterogeneous environment. It provides useful services for Grids, including mutual authentication and single sign-on, which basis on public key cryptography. The main GSI functionality cover:

- secure communication between nodes in Grid.
- supporting security across organizational boundaries,
- supporting "single sign-on" for users of the Grid.

Every user and service on the Grid is equipped with a certificate and in this way identified. The certificate contains the following information:

- a subject name, which identifies the person or object that the certificate represents,
- the public key belonging to the subject,
- the identity of a Certificate Authority (CA) that has signed the certificate,
- the digital signature of the named CA.

CA is used to certify the link between the public key and the subject in the certificate. GSI certificates are encoded in the X.509 certificate format, a standard data format for certificates established by the Internet Engineering Task Force (IETF).

Mutual authentication is very important operation in establishing secure connection. It is used for proving that the two parties are who they say they are. The Secure Sockets Layer (SSL) is used in Grid to perform mutual authentication.

To protect communication link against intruders an encryption can be enabled. But this operation is not performed by default in GSI. If confidential communication is desired the GSI functionality be used to establish shared key for encryption.

Another advantage of this infrastructure is communication integrity. It assures that communicates sent between two parties can not be modified. Communication integrity, which is enabled by default, introduces some overhead in communication, but not as large an overhead as encryption.

Every user which is intending to use the GSI software has to be equipped with private key - usually stored in a file in the local computer's storage. To prevent this key from stealing it is additionally secured by password and user must enter pass phrase to decrypt the file containing their private key to use it in GSI.

The single sign-on concept rely on idea that the user should authenticate only once and next should be able to access to many different resources in the Grid without the necessity of providing the pass phrase again and again. In situation where a Grid computation requires several Grid resources to be used the need to re-enter the user's pass phrase can be avoided by creating a certificate proxy. This proxy consists of a new certificate and a new private key, which is signed by the owner, rather than a CA. Some characteristic feature of the proxy is its limited lifetime.

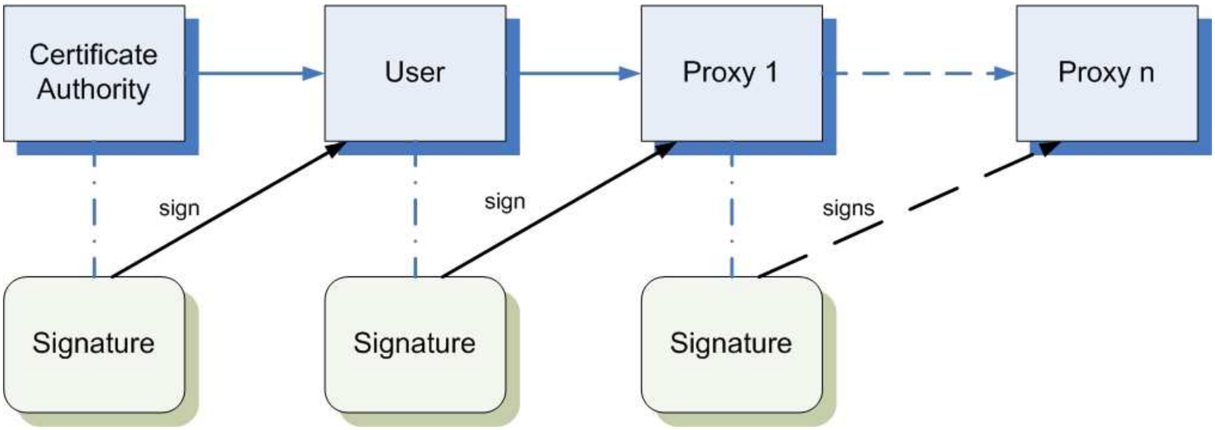


Figure 1 The idea of using the proxy certificates

As it was said before the proxy’s lifetime is not very long so it does not have to be kept as secure as the owner's private key so this mean that encryption is not obligated. Moreover, it is enough to store the proxy's private key either in a local storage system (with appropriate file’s permission) or in on-line credential repository like MyProxy. This repository, protected by a pass phrase, eliminates the need for manually copying private key and certificate files between machines. MyProxy can also be used for authentication to grid portals and credential renewal with job managers.

3.3 Data transport and management

3.3.1 Introduction

An enormous amount of data is collected under VLBI experiment. It causes that the designed system has to have additional, specialized functionality connected with data transport and management. Data retrieved from radio telescopes are then transferred through many processing stages of experiment scenario to get the final correlation product (e.g. visualization). Due to the huge size of data and assurance of efficient post-processing computation a set of grid tools must be involved in this process. It is not enough to simply store data on some machine in the grid and download it when computation will be performed. Retrieving and downloading process must be optimized for speedy access in different geographic locations, catalogued with descriptive information for easy retrieval, and made available to computation jobs running on the Grid.

Based on the high-capacity storage systems and broadband networks, the Grid community has produced a set of components for working with and managing data on the Grid.

Several components in the Grid space are aimed specifically at providing uniform Grid interfaces to various types of data:

- GridFTP - A uniform, secure, high-performance interface to file-based storage systems on the Grid,
- OGSA-DAI - An OGSA interface for accessing XML and relational data stores,
- Metadata Catalog Service (MCS) - A stand-alone metadata catalog service with an OGSA service interface.

These tools specialize in moving and transferring data between Grid systems. Each tool meets specialized application or user needs and some also provides interfaces to specialized storage systems:

- `globus-url-copy` - a command-line tool for requesting GridFTP transfers,
- Reliable File Transfer (RFT) Service - an OGSA service that allows clients to request data transfers and then "disconnect" while the transfer takes place,
- UberFTP - A text-based interactive client for GridFTP,
- GSI-SCP/SFTP - Popular OpenSSH tools that support Grid authentication.

Moreover, there were implemented a set of tools which help to optimize the use of storage systems for specialized user communities:

- Replica Location Service (RLS) - a distributed mechanism for keeping track of the locations of replicated data on a Grid,
- NeST - a "storage appliance" that provides remote access to local data when computation jobs are running,
- DataCutter - a system that uses data filters and streams to segment datasets in efficient ways on a Grid.

3.3.2 GT data management

Data management tools are concerned with the location, transfer, and management of distributed data.

3.3.2.1 GridFTP

GridFTP is a protocol defined by Global Grid Forum Recommendation GFD.020, RFC 959, RFC 2228, RFC 2389, and a draft before the IETF FTP working group. The GridFTP protocol provides for the secure, robust, fast and efficient transfer of (especially bulk) data. The

Globus Toolkit provides the most commonly used implementation of that protocol, though others do exist (primarily tied to proprietary internal systems).

3.3.2.2 Reliable File Transfer

The Reliable Transfer Service (RFT) is a web service that provides interfaces for controlling and monitoring third party file transfers using GridFTP servers. The client controlling the transfer is hosted inside of a Grid service so it can be managed using the soft state model and queried using the ServiceData interfaces available to all Grid services.

RFT implementation in version 4.0 of GT uses standard SOAP messages over HTTP to submit and manage a set of 3rd party GridFTP transfers and deletion of files and directories using GridFTP. The service also provides an interface to control various transfer parameters over GridFTP control channel like TCP buffer size, parallel streams, DCAU etc. The user creates a RFT resource by submitting a Transfer Request (consists of a set of third-party gridftp transfers) to RFT Factory service.

The resource is created after the user is properly authorized and authenticated. RFT service implementation exposes operations to control and manage the transfers (the resource). The resource the user created exposes the state of the transfer as a resource property to which the user can either subscribe for changes or poll for the changes in state periodically using standard WS-RF command line clients and other resource properties.

3.3.2.3 Replica Location Service

The Replica Location Service (RLS) maintains and provides access to mapping information from logical names for data items to target names. These target names may represent physical locations of data items, or an entry in the RLS may map to another level of logical naming for the data item.

The RLS is intended to be one of a set of services for providing data replication management in grids. By itself, it does not guarantee consistency among replicated data or guarantee the uniqueness of filenames registered in the directory. The RLS is intended to be used by higher-level grid services that provide these functionalities.

3.3.2.4 Data Replication Service

Data Replication Service (DRS) provides a pull-based replication capability for Grid files. The DRS is a higher-level data management service that is built on top of two GT data management components: the Reliable File Transfer (RFT) Service and the Replica Location Service (RLS).

The function of the DRS is to ensure that a specified set of files exists on a storage site. The DRS begins by querying RLS to discover where the desired files exist in the Grid. After the files are located, the DRS creates a transfer request that is executed by RFT. After the transfers are completed, DRS registers the new replicas with RLS.

DRS is implemented as a Web service and complies with the Web Services Resource Framework (WSRF) specifications. When a DRS request is received, it creates a WS-Resource that is used to maintain state about each file being replicated, including which operations on the file have succeeded or failed.

3.4 Efficiency

During analysis of the e-VLBI environment security tools we need to take care also about the efficiency parameters. Additional functionality (like communication encryption, mutual authentication etc.) cause time overheads. It is especially important in systems like e-VLBI where data transfer is a very time-consuming process.

Basing on experiments and results presented in the paper [BBRS2004] we can conclude the following:

- the average CPU time required to perform data transmission with integrity checking is over four times greater, than the time required for the clear text transmission,
- transferring data via encrypted connections requires more resources - the average CPU time for this connection is over seven times greater than for insecure ones,
- the CPU time required for the most resource consuming connection is in the order of 1/10 milliseconds; thus, even the connection that requires the most computing resources should not cause significant overhead, neither in the CPU consumption nor in the delay of message delivery.
- the size of the transmitted packet is significant for the transmission efficiency,
- the average CPU time required to transmit 100 bytes significantly decreases with packet size increasing to the size of 1.5 kilobytes, so it is more efficient to transmit small amount of large packets than huge amount of small packets,
- regularity presented in previous note can be also used for the raw TCP connections (however, this factor seems to be more significant for the secured ones). The establishment of a secure connection is an exceptionally resource consuming process. Therefore, secure connections servers can handle fewer connections than those which use raw TCP sockets.
- the amount of connections which can be handled by such a server in one second is large enough to respond requirements of the OCM-G.

Designing the security policy of a system we should consider which security level is really required for data transfer.

The differences in CPU time consumption between particular levels of security is significant. Therefore, it is desirable to restrict the security aspects only to those which are really required by the user’s requirements and system.

4 Network monitoring and management

4.1 Network monitoring

The European VLBI network is spread over European countries and beyond thus data transmission will cross several administrative domains including National Research Networks (NRENs) as well as pan-European GÉANT2 network [GÉANT2]. The development of GRID system and eVLBI data transmissions between different geographically distributed partners will require proper resource management and network monitoring in order to allow for automatic resource distribution and QoS provisioning.

4.1.1 Monitoring requirements

Based on the set-up of the eVLBI network and initial requirements these are the highlights of the monitoring architecture:

- It must operate inter-domain and end-to-end. There is also a need for intra-network measurements in order to identify the exact amount of resources. Therefore, metrics should be provided on a hop-by-hop basis
- It should be flexible enough to accommodate different types of metrics e.g. capacity, available bandwidth, delay
- The architecture must encompass using different existing monitoring tools to measure various metrics e.g. MRTG [MRTG], Cacti [Cacti], BWCTL [BWCTL]
- It must standardise interfaces between various architecture components and provide a defined interface to integrate GRID resource brokers or QoS provisioning systems
- It should carefully cover security issues. The architecture must be able to deal with customised requirements on what metrics and with whom they can be shared

4.1.2 Monitoring architecture

Out of the existing worldwide projects, probably only the E2E Performance Initiative from Internet2 [Internet2][e2ePIPEs] and the European perfSONAR are important for eVLBI.

4.1.2.1 E2E piPEs

End-to-End Performance Initiative (E2E piPEs) is a framework designed for the Internet2 network. Currently a subset of it called piPEs is implemented and used in Internet2 enabling BWCTL and OWAMP measurements. The main objective of the piPEs project is to enable end-users and network operators to determine end-to-end (E2E) performance capabilities, locate E2E problems, and contact the right person (with evidence) to get an E2E problem resolved. PiPEs is a set of Perl scripts relatively complex to extend with no multi-domain functionality. It uses common active measurement tools namely iperf (wrapped in BWCTL) for throughput and OWAMP for delay and loss.

The piPEs architecture considers the E2E path to be composed of a series of partial paths composed of layer 3 network components (e.g. end-hosts and routers). It assumes the presence of a network measurement node (NMN) topologically “next to” some (if not all) of the network components along the E2E path. Figure 5 depicts the idea of partial tests between NMS. A remote machine can then initiate regularly scheduled or on-demand tests between any two such NMNs, subject to authorization and policy restrictions. The results of such tests are stored in a database of performance results, and a remote machine can request such results from said database. PiPEs has introduced an interesting concept of on-demand test triggered by someone not having local access to the machine on which the measurement point is located.

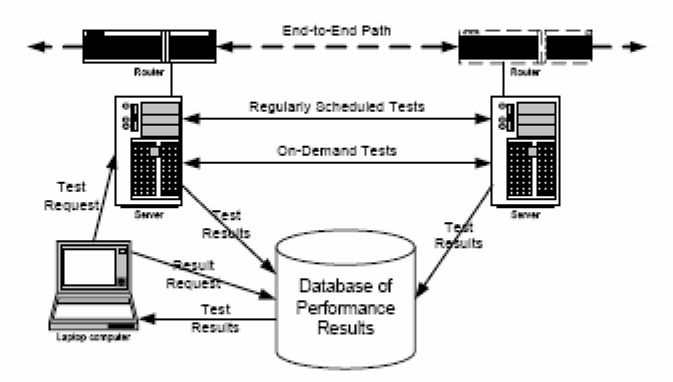


Figure 5. Partial path testing between NMNs in piPEs.

The main disadvantage is that the E2E piPEs project currently exists in prototype form and has been deployed only across Internet2's Abilene backbone. The initial idea and efforts spent on piPEs has been finally put by Internet2 into the joint project perfSONAR described in the following chapter. Similarly to perfSONAR piPEs was also working through the Global Grid Forum (GGF) Network Measurement Working Group (NMWG) to standardize request and report schemas that would enable interoperability between network performance data producers and consumers.

4.1.2.2 perfSONAR

The current development of state-of-the-art network monitoring architectures within Europe is led by perfSONAR project [perfSONAR] which supports end-to-end network monitoring across geographically distributed domains.

IST project GN2 [GN2] supported by partners from Internet2 or ESNet (United States) develops a framework that is based on distributed services providing a common interface for applications to interact with different monitoring tools and enables the users to obtain and manipulate measurement data. The measurement framework receives requests for monitoring tests or archived data from the data visualisation tool (the monitoring application) or user tools e.g. QoS provisioning systems or resource brokers. It then publishes monitoring data to the tool through a standardised interface. The framework is gathering, transforming, storing network measurement information as well. The actions required for that are a set of services, each performing specific actions, offering a set of functionalities necessary to assist in the monitoring activity and accessible via defined interfaces. The measurement framework is based on the SOA (Services-Oriented Architecture) [SOA] concept and the services are built using a number of elementary components which are assembled into a larger framework. Some of these components are similar (if not identical) in multiple services. The design of these components was made in GN2 JRA1 design document [JRA1].

Each type of measurement tool has its own unique interface that can be used to acquire monitoring data. A unique interface to these types of tools is developed for ease of integration into the monitoring framework as a whole. The Measurement Point (MP) service is a standard interface "wrapper" around one or more measurement tools responsible for providing measurement data which are currently not being measured or not stored in an archive. It acquires measurement data either by initiating active measurement tests or querying passive measurement devices. Another type of MP allows administrators to access network equipment and retrieve network information via unique interface. Figure 6 depicts perfSONAR monitoring architecture.

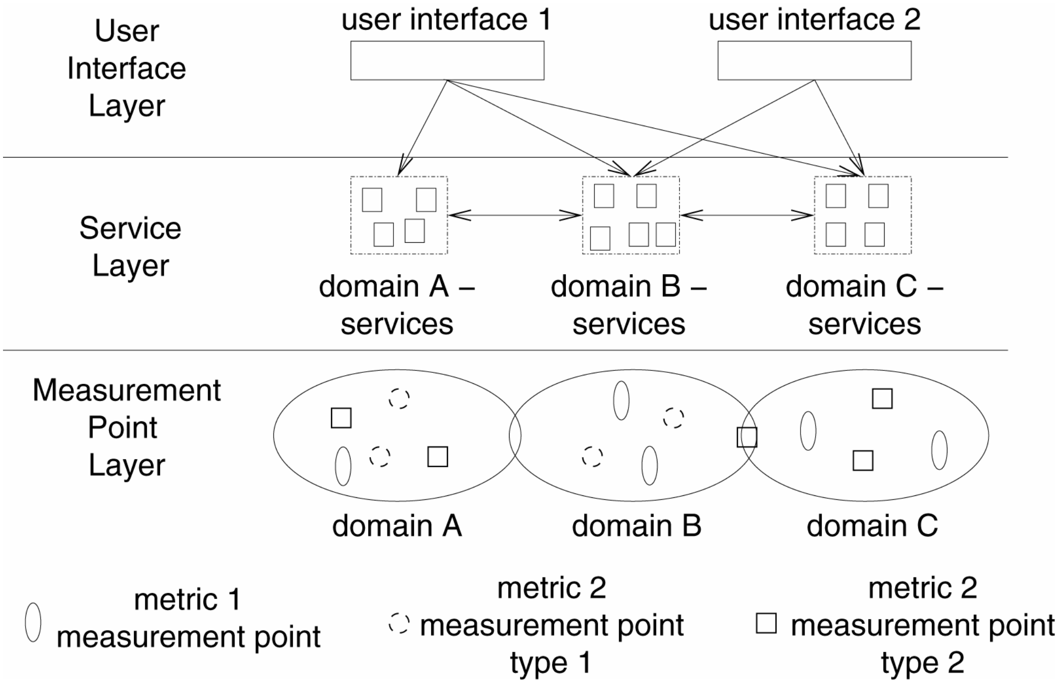


Figure 6. perfSONAR monitoring architecture.

4.1.3 Framework integration and development status

The eVLBI requirements for network monitoring fit best into the design and deployment of perfSONAR which may be adopted for GRID resource management and QoS provisioning.

The measurement framework receives monitoring test/data requests from other tools and applications through a standardised interface. This open architecture allows external applications for making use of data collected from the perfSONAR framework thorough the NMWG (The Global Grid Forum's Network Measurement Working Group [NMWG]) XML schema. XML schema is used to represent measurement data and can be easily integrated with tools willing to interact with the framework. The latest perfSONAR schema definition is available in project deliverable DJ1.2.2 [JRA1DJ1.2.2].

More than dozen installations currently running the first release of perfSONAR have enabled first view of the European networks, making effective troubleshooting of the performance problems and access to the available data placed in various administrative domains possible. The crucial portion of the prototype system is the MA service, which is a wrapper around commonly used Round Robin Databases (RRD) [RRD] in the first place and provides link utilisation statistics through a Web Service [WebService] interface. The current monitoring infrastructure enables also capacity, packet loss and delay measurements among several European and North American partners. The availability of measurement data depends on the scope of implementation of perfSONAR as well as data access policy of a particular domain.

The number of services and their complexity will increase over time while adding additional modules, features and measurement types so that it will cover all domains involved in eVLBI network. If requested eVLBI partners could deploy monitoring nodes within their domains to provide necessary measurement data.

4.2 Quality of Service

EVLBI, as a application supposed to transmit huge amounts of real-time data, will impose very strict requirements on the wide-area transmission networks which can be used by eVLBI. Such requirements are identified in the networking community as QoS (Quality of Service). There are four major parameters of QoS which will determine the quality of transmission services for eVLBI:

1. capacity – the number of bits per second which is available to a flow or an aggregation of flows
2. one-way packet delay – the time between the transmission of a packet at its source at the moment it is fully received by the destination
3. one-way packet delay variation – the difference in the one-way packet delay of two successive packets in the same IP flow
4. packet loss – the ratio of packets sent by source and not delivered to destination (or delivered in errors)

Capacity seams to be the most crucial QoS parameter for eVLBI as it determines the amount of data which may be transmitted over the network. EVLBI is going to transmit large portions of data, which will require that the network offers high capacity transmission services with sufficient values of the other QoS parameters.

There are two networking technologies that may be used by eVLBI, which have adopted different approaches to QoS. Both technologies are used in research networks (GÉANT2, national research and education networks) which can offer transmission services for eVLBI. In a traditional IP network traffic is transited on best-effort basis without any guarantee of QoS; QoS can be achieved by reservations of bandwidth and admission control which requires that some additional mechanisms are introduced into the network. In the other approach, which is adopted in optical networks, every transmission requires prior reservation of bandwidth and the QoS parameters are guaranteed upon the reservation is accepted by the network.

4.2.1 QoS in IP networks

Legacy IP networks use best-effort paradigm towards data transmission. This means that the network is obliged to do its best to transmit the data however the network does not guarantee any QoS parameters. The best example of such network is the Internet. The network tries to deliver all packets in the shortest time possible but in case of congestion the traffic can experience significant packet loss and substantial increase of packet transmission time. The capacity available for a given data stream in a best-effort network is also not guaranteed and depends on the fluctuating network load. The best-effort service is not suitable for applications which require large portions of data to be transmitted in a determined time – like eVLBI. The European research networking community has anticipated the appearance of this type of applications and proposed a new service which may provide the most demanding applications with the suitable transmission parameters. This service was proposed by the IST project SEQUIN [SEQUIN] and is named Premium IP. According to a SEQUIN deliverable:

The IP Premium service is defined such that, for the selected packets, capacity is conserved and, hence, packet loss is zero or negligible, apart from bit error rate and other similar causes. However, that loss is never due to congestion.

IP Premium will require that, for the selected packets, the delay and delay variation along a path is independent of the load of the path and will be very similar to the values obtained at empty network.[SEQUIN D2.1]

All routers in an IP network with Premium IP service enabled give priority to premium traffic over the normal traffic (which is still treated as best-effort). To conserve the QoS parameters of Premium traffic, the amount of such traffic must be limited in relation to the capacity of the network and the bandwidth of any Premium IP transmission must be reserved in the network. This way Premium traffic will never suffer from congestion in the network; even when best-effort traffic is congested, the Premium traffic will pass the network unaffected.

Premium IP service is available in the European research network GÉANT2 and is planned in several national research and education networks (NRENs) in Europe. The introduction of Premium IP in NRENs will allow for end-to-end bandwidth reservations – from a research centre to another research centre in another European country.

One of service activities of the GN2 [GN2] project (SA3) is preparing a provisioning system, which will be able to automatically provision IP Premium services in GÉANT2 and NRENs (in the future) on request of a user or an application. This system is named Advanced Multi-domain Provisioning System. The first version of this system is already running in GÉANT2 and allows for automatic IP Premium reservations. The next version will support also automatic provisioning of IP Premium services. The system will be able to provision services in multiple domains – like in GÉANT2 and the NRENs to which IP Premium users are connected.

It is envisaged that for proper operation of Premium IP service, the amount of Premium traffic on any ingress port of GÉANT2 network should be limited to 5% of the port capacity. As the fastest IP access ports in GÉANT2 have the capacity of 10 Gbps, the limitation means that this service is suitable for transmission that requires capacity up to 500 Mbps. For transmissions with higher bit rates optical networks may be used.

4.2.2 QoS in Optical Networks

The research applications which require more capacity than offered by IP Premium services will use optical transmission directly, without the routed IP layer. Optical networks in conjunction with TDM technology may offer virtually any transmission capacity up to 10Gbps, while technology for higher capacities (up to 40Gbps) is available but not implemented in European research networks yet. Optical technology has been recently introduced in the GÉANT2 network and several NRENs and allows for establishing of end-to-end optical paths between research centres in Europe. European research projects will also be able to use cross-border optical links, which are being developed by GN2 research activity JRA4. Such links will connect NRENs of neighbouring counties and will be used (together with NRENs) to establish international optical paths for research projects.

In optical networks a data path must be established prior to any transmission of user's data. When the path is established, the capacity and other QoS parameters are guaranteed for the time agreed by the user and the network. When bandwidth is reserved, congestion in an optical network is not possible, thus one-way packet delay will be constant and packet loss will be independent of network load.

There are ongoing efforts to create a system that will support automatic provisioning of optical paths in multiple domains on the request of an application. The legacy management systems for optical networks are limited to one domain only (devices from one vendor and managed by a single entity) and do not have open interfaces to applications. Some new developments, which address the issue of integration between applications and networks were identified and extensively described by the GN2 project. In this document only the most promising are mentioned in order to indicate the possibility of using them for provisioning bandwidth for eVLBI transmissions. More detailed descriptions of those systems (and others) can be found in [JRA3 DJ3.2.2].

The Dynamic Resource Allocation Controller (DRAC) [DRAC] system, which is developed by Nortel Networks, is used in the Dutch NREN – SURFnet [SURFnet]. This system aims at full automation of the bandwidth provisioning process, which can be initiated by an application and executed without any human operator involved. DRAC supports DWDM networks, TDM networks, Ethernet (with VLAN) and next-generation SONET/SDH. Other network elements and protocols can be supported by additional adapters.

The Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON) [DRAGON] project is funded by the United States National Science Foundation and developed by several American institutions (mostly research institutions). The goals of DRAGON project are to develop infrastructures, technologies, and software to provide dedicated paths across heterogeneous network technologies (such as DWDM, Ethernet, next-generation SONET/SDH).

Optical Dynamic Intelligent Network (ODIN) is a part of Optical Metro Network Initiative (OMNInet) project [OMNINET]. ODIN is a bandwidth broker able to allocate lightpaths or optical VPNs for clients' application. It can control a single administrative domain and is able to communicate with bandwidth brokers of other domains in order to establish a multidomain path.

The objective of the User Controlled LightPath Provisioning (UCLP) [UCLP] project is to provide a software system that allows users to own and control end-to-end lightpaths. The UCLP software allows end users to self provision and dynamically reconfigure optical networks within a single domain or across multiple independent management domains. It gives users the opportunity to control (including dynamic reconfiguration) part of optical network without interaction with the operator of the optical network.

The GN2 project is developing a new bandwidth provisioning system with intention of introducing it as a service in GÉANT2 or its successor. This system will be technology independent and will be able to provision data path crossing various technology domains (like optical network, TDM networks, Ethernet networks, MPLS network, IP Premium network) and administrative domains. This system will support proxy interfaces to other bandwidth provisioning systems (e.g. UCLP) and other networking technologies. The system developed by GN2 is of special interest for the EXPRES project as it is the most likely to be deployed on European scale in a production network used by research projects (such as GÉANT2).

There are also other IST projects working on bandwidth provisioning systems intended for some specific network technologies. The MUPBED [MUPBED] project is working on a provisioning system for ASON/GMPLS networks which will also be able to deal with native Ethernet networks. Another IST project – PHOSPHORUS [PHOSPHORUS] – is going to address the integration between GRIDs and the control planes of optical networks (GMPLS, UCLP and DRAC) thus the network will be seen by GRIDs as one of GRID resources and applications will be aware of their complete grid resources (computational and networking).

4.3 Implementation example

4.3.1 Network management in National CLUSTER of LInuX Systems (CLUSTERIX)

Clusterix project is a Polish enterprise initiated at the end of year 2003, to create productive, efficient, and secure GRID environment. The project involves twelve major Polish educational centers engaged in distributed computing research, including four supercomputing centers. Each of them deploys a local cluster built in 64 bits architecture, attached with dedicated channels to national optical network PIONIER (see Figure 7).

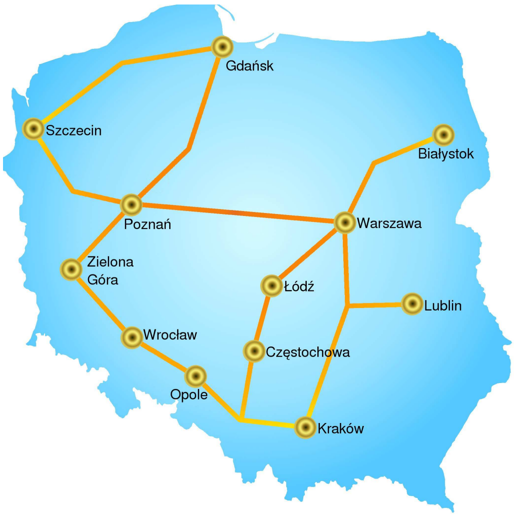


Figure 7. Clusterix connections in PIONIER network.

The MANs involved in Clusterix project are interconnected through dedicated channels within the PIONIER network. One channel is provided as Ethernet VLAN with 1 Gbit/s bandwidth guarantee, second as 100 Mbit/s. Computation data and network measurements are transmitted through the first VLAN. The second one is used to manage network measurements and upgrade software.

Figure 8 shows an example of computing infrastructures which are installed in Clusterix partners.

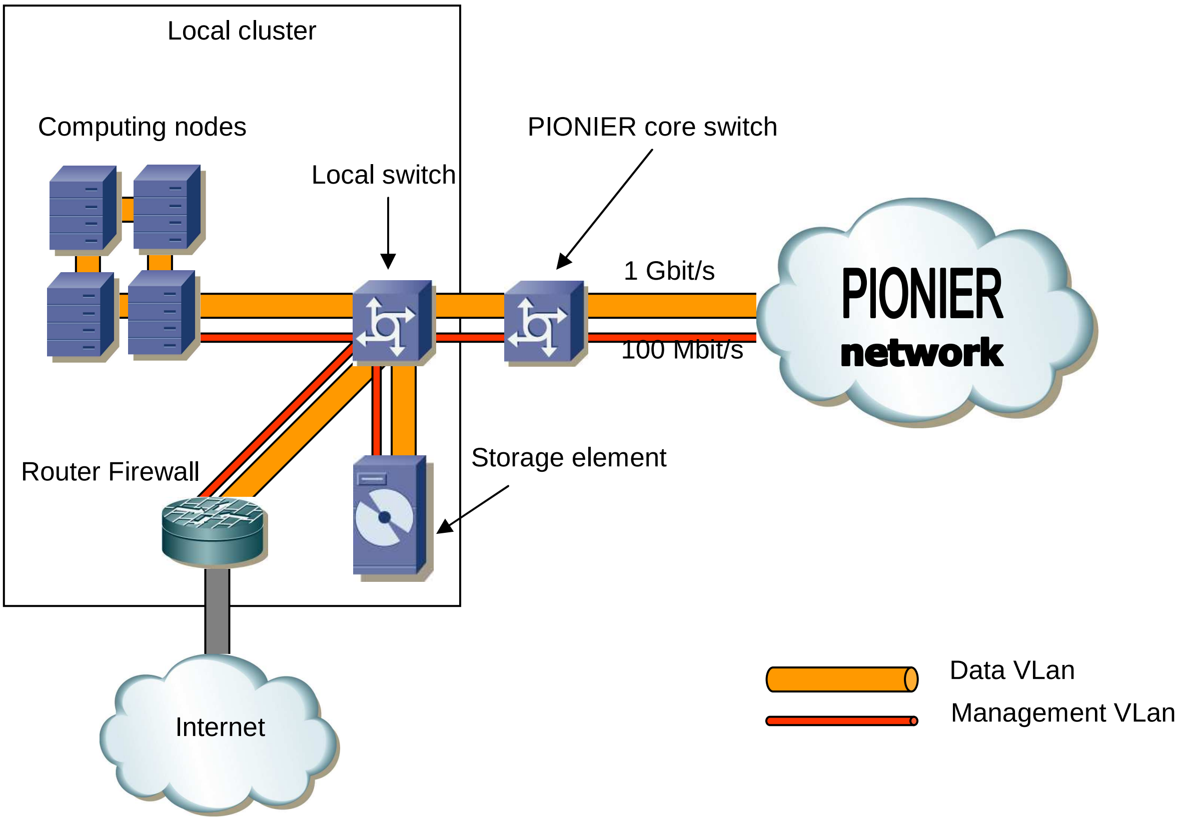


Figure 8. Local cluster architecture.

One of Clusterix tasks is focused on the network issues, including active measurements and management. This task product is an integrated tool (see Figure 9), able to track and show network status, where both devices and network parameters are in scope. To retrieve such information, the tool relies on a measurement agent mesh, centrally controlled by a group of redundant managers. The mesh consists of two agent types – backbone agents placed in backbone network, and local agents installed within computing clusters. Measurements are restricted to the agents of the same type only, which separates local and backbone infrastructures. Clusterix measurements are organized in sessions, that always involve two agents. The service is able to create sessions automatically between proper pairs of agents, in order to cover all network with required measurements. Five common network parameters are collected, including packet RTT, jitter, loss, reordering, and duplication. The measurement protocol is based on the Internet2 experiences with One Way Delay Measurement Protocol (OWAMP), but due to many modifications and lack of precise time synchronization, Clusterix protocol is used for two way measurements. UDP protocol is used for test packet transmissions, and TCP with SSL enhancements is used to control the agents by manager. To ensure service reliability and avoid single point of failure, several manager instances are running at the time. Despite of the number of instances only one manager is active and is allowed to communicate and control the agents mesh. In case of failure, the active manager is selected using simple selection algorithms, therefore service is available as long as there is at least one manager running. To provide more detailed network information, SNMP monitoring is also performed for backbone and local switches, where port statuses and current interface loads are collected. Again Clusterix's approach differs from some other grid solutions, where measurements are performed by external tools like IPerf or UDPMon. Instead of that, here measurement results are collected all the time by agents, which are system part. Moreover, not only connections characteristics are collected, but also network infrastructure data, like link and network device states.

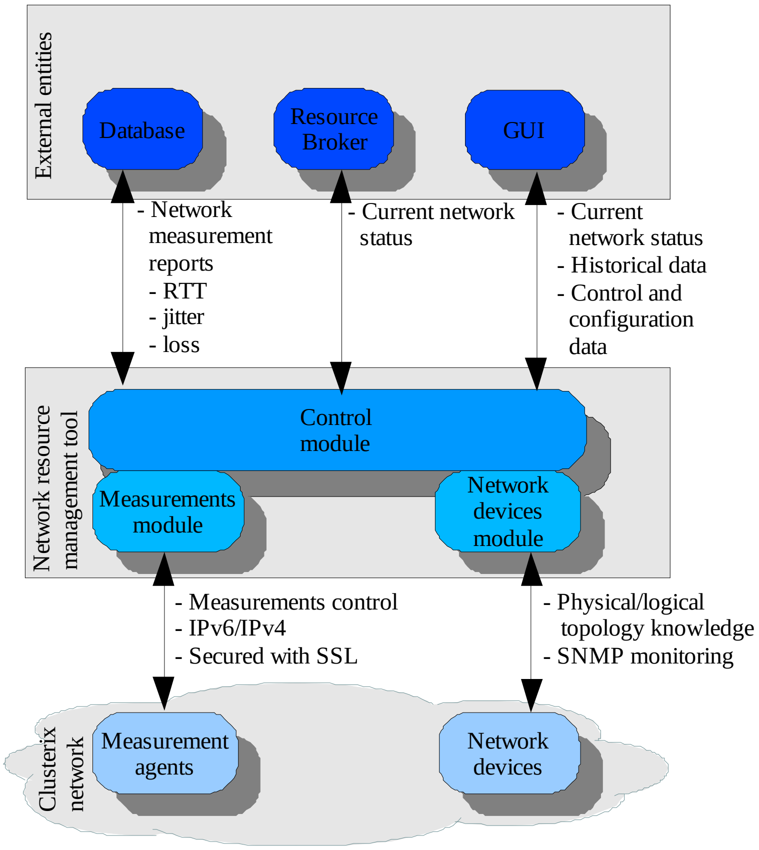


Figure 9. Network management tool components.

All monitored and measured values are stored in a single database, placed within the Clusterix backbone. Historical views and statistics are available at any time, in order to build network utilization reports or for failure investigation purposes. In case of database failure recent measurement results are still available for external clients, as manager keeps a cached data backup. Usually Grid solutions distribute gathered results through LDAP database, but Clusterix data are placed in dedicated PostgreSQL database, which is more flexible solution. Those data are available via web services interface for two main clients which are interested in information delivered by network management tool – task broker and user interface.

Task broker is responsible for assigning tasks to specific local clusters, according to the results of decision procedure. The input for this procedure can be a CPU usage, memory available, free disk space, and what is new, also the network parameters. Moreover, network management service is able to provide information not only about current and past parameter values, but also to predict things like traffic load according to historical data. In the future it will be also possible to perform resource reservation and to establish a direct intercluster connection with guaranteed service quality.

Graphical user interface implemented as Java applet, is available for Clusterix administrators in order to track current network state and failures. This interface allows to see all measured and monitored values, regarding physical and logical interconnectivity. The sessions and monitoring options are also configurable through the user interface, which can be accessed with any Java enabled web browser.

5 The software correlator

In January 2005 the descent of the Huygens probe in the Titan atmosphere was successfully tracked using a specially developed software correlator. This correlator was transformed into a much simpler broadband correlator mimicking the EVN Mark IV hardware correlator operated by JIVE.

5.1 Software correlator – current status

The software correlator, further called SFXC, is a set of applications which are working together to obtain a correlation product from a raw data files, collected from radio telescopes. Each application is controlled by a control file with settings, file names, processing options, etc. This file is of the keyword-value type. All keywords are unique so the control files for the different applications can be merged into one.

The following diagram depicts the general architecture of the software correlation process.

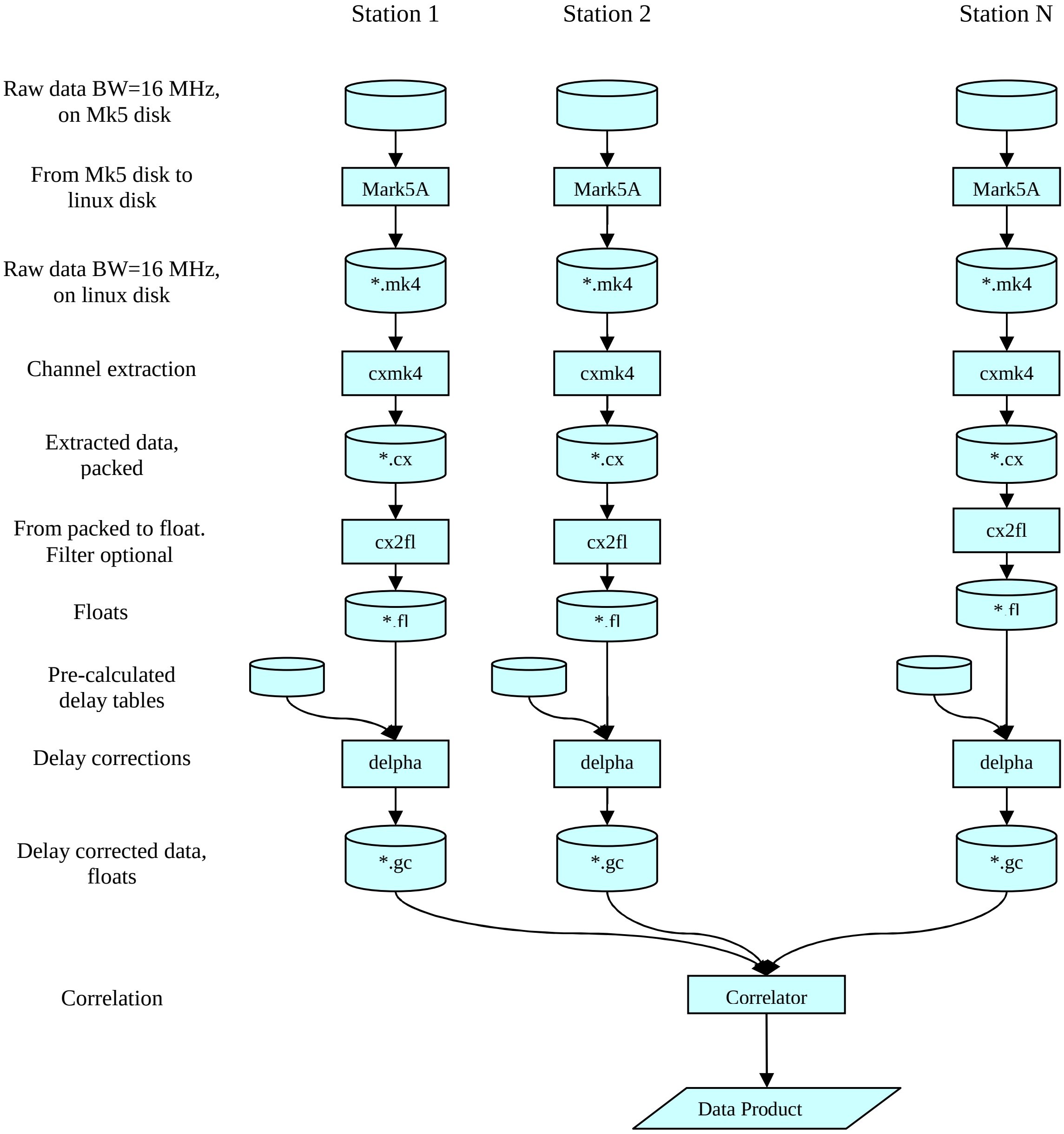

Station 1
Station 2
Station N
Raw data BW=16 MHz, on Mk5 disk
From Mk5 disk to linux disk
Mark5A
Raw data BW=16 MHz, on linux disk
*.mk4
Channel extraction
cxmk4
Extracted data, packed
*.cx
From packed to float. Filter optional
cx2fl
Floats
*.fl
Pre-calculated delay tables
Delay corrections
delpha
Delay corrected data, floats
*.gc
Correlation
Correlator
Data Product

Figure 10. Correlation process with intermediate file storage

A brief description of each processing step.

1. Mark5A: transfer data (Mk4 formatted) from Mk5A disks to linux type disk using the Haystack proprietary software
2. cxmk4: extract channel pairs from the Mk4 formatted data files. The extraction is necessary because the data in the Mk4 file it not contiguous.
3. cx2fl: data is still packed, 2 channels per byte, in order to reduce the data volume. However floating point numbers are needed by the next step. Therefore cx2fl converts a single cx file containing data from two channels into two separate file each containing single channel data in float format. The data volume by now is increased by a factor of 8. cx2fl optionally can filter down the original bandwidth.
4. delpha: during a VLBI session the participating telescopes receive the signals at

different times. Because the VLBI technique requires the correlation of the same wave fronts, delay corrections have to be applied to the original signal. delpha performs the delay corrections on a single fl file based on a pre-calculated delay file.

5. When the data from all participating telescopes has gone through the previous processing steps, data is correlated into a correlator product.

5.2 Software correlator –future plans

During PSNC-JIVE meetings it was concluded that this architecture was not so very well suited to be used in a distributed fashion on the Grid. Especially the very large (temporary) intermediate files slow down the whole process and put a heavy load on file systems.

Therefore the current software correlator architecture was redesigned in such a way it does not use files as storage for intermediate results. All separate applications as discussed in section 5.1 will be combined into a single application. The MPI protocol is used in the source to exploit a number of processors.

The Mark5A software responsible for getting the Mk4 formatted data from the Mk5 disk or Mk5 I/O board will not be incorporated in the software correlator application. Section 6.1 will discuss the role of the Mark5A software and hardware in more detail.

A brief description of each processing step.

1. Mark5A: transfer data (Mk4 formatted) from Mk5A disks to linux type disk using the Haystack proprietary software
2. Extract for all stations the data for the wanted channel.
3. Apply the necessary delay and phase corrections (station specific)
4. Correlate the delay corrected data (auto correlations and cross-correlations)

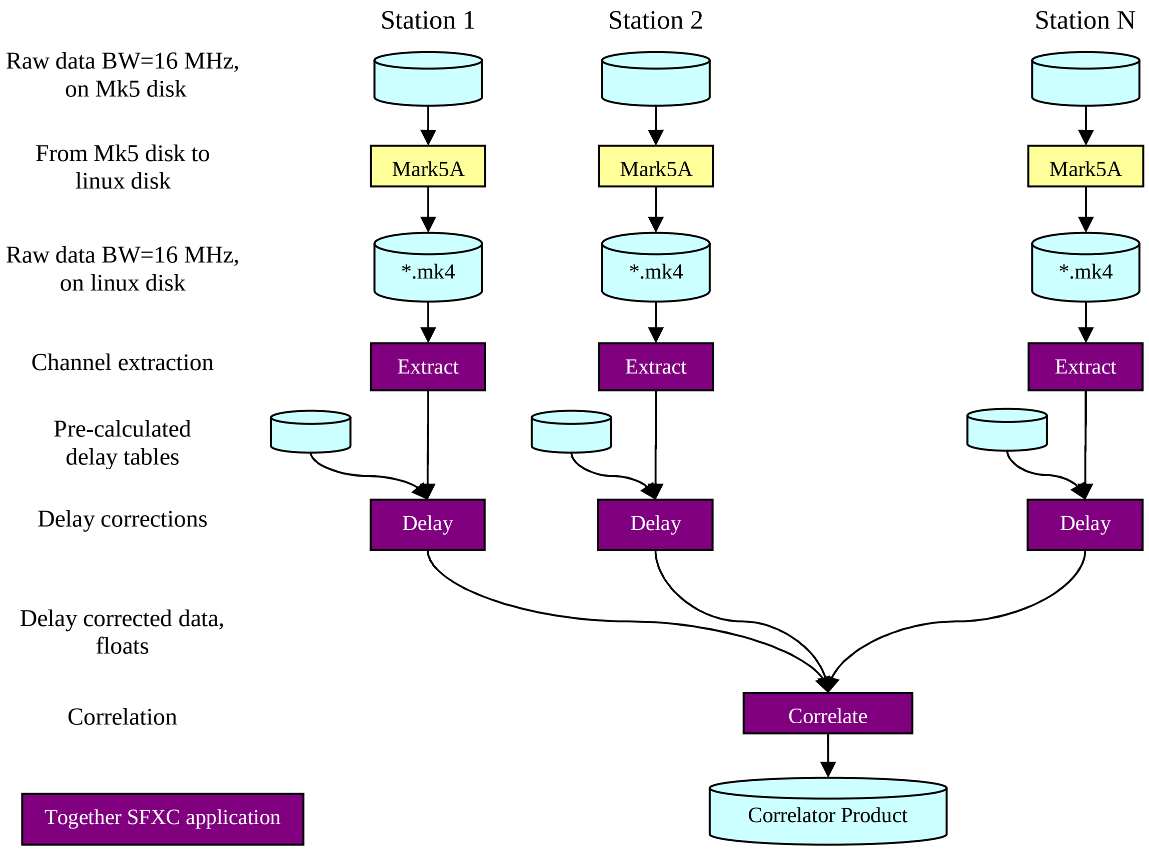
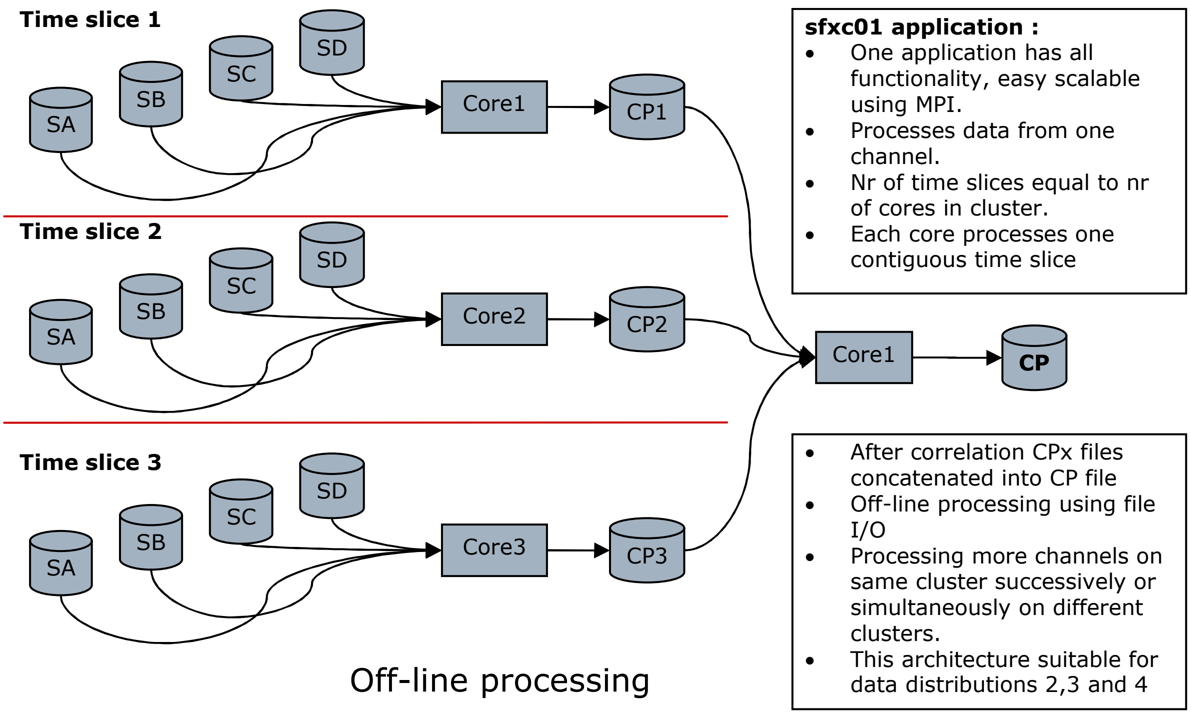

Station 1
Station 2
Station N
Raw data BW=16 MHz, on Mk5 disk
From Mk5 disk to linux disk
Raw data BW=16 MHz, on linux disk
Channel extraction
Pre-calculated delay tables
Delay corrections
Delay corrected data, floats
Correlation
Mark5A
*.mk4
Extract
Delay
Correlate
Correlator Product
Together SFXC application

Figure 11. Correlation process without intermediate file storage

Various scenarios for the distribution of the computational workload over the available processors are possible. Scalability is one of the most important factors when it comes to designing a distributed software correlator. The time slicing principle of the input data leads to a easy scalable software correlator. The discussion of this distribution scenario and others is the subject of workpackage “2.2.3 Scaled up version for clusters”. Here we proceed with the description of software architecture for time slicing

Suppose that 3 processors are available. The complete time interval of the astronomical expriment is then divided into 3 contigous chunks of equal length and each processor gets one of these chunks to process. After the correlation of the 3 chunks the separate correlator products a concatenated into one file. This approach requires a completed astronomical observation and the availability of the input data files.



sfxc01 application :

- One application has all functionality, easy scalable using MPI.
- Processes data from one channel.
- Nr of time slices equal to nr of cores in cluster.
- Each core processes one contiguous time slice

- After correlation CPx files concatenated into CP file
- Off-line processing using file I/O
- Processing more channels on same cluster successively or simultaneously on different clusters.
- This architecture suitable for data distributions 2,3 and 4

Figure 12. Time slicing for off-line processing

For real time processing a different approach of time slicing is necessary. Suppose that 4 processors are available and each processor can handle 1 sec of data from all involved telescopes. Data for second 1 goes to processor 1, data for second 2 goes to processor and data for second 3 goes to processor 3. After processing the 3 seconds of data the correlator products are put in a file in the right order by a 4th processor. In the meantime processor 1, 2 and 3 are processing the data for second 4, 5 and 6.

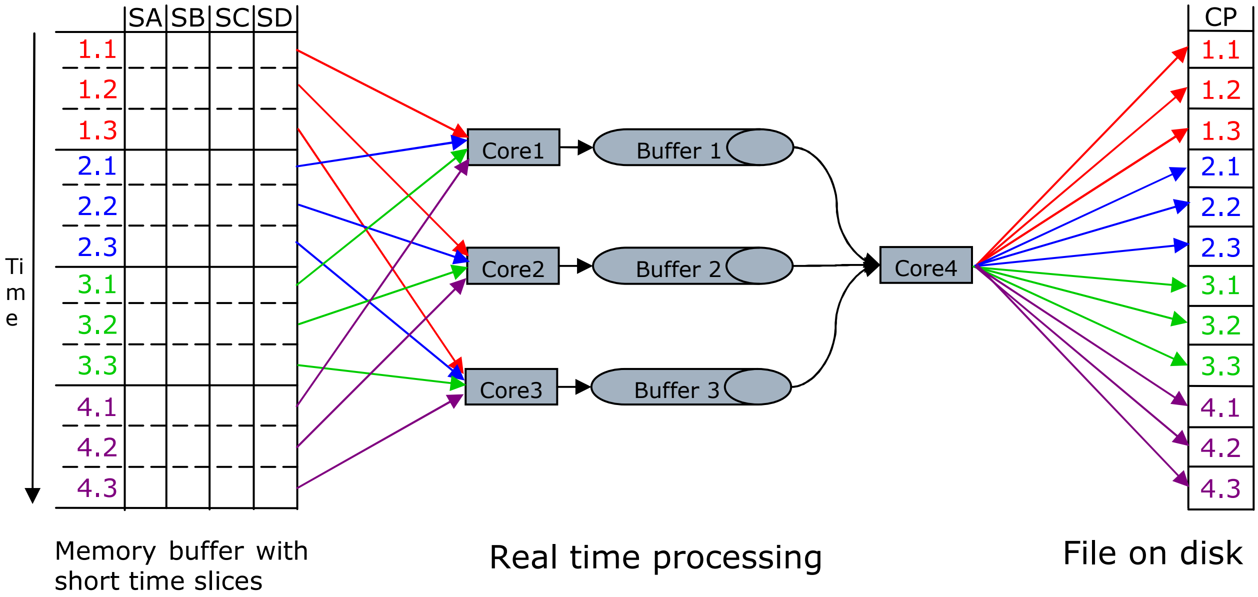


Figure 13. Time slicing for real-time processing

Currently the architecture for off-line processing is being implemented because this is a simpler short term goal. In a later stage the real time processing version has to be implemented as well.

6 Grid based eVLBI – overall design

In this section, the overall design of the proposed solution is presented and analysed. The final design was a result of careful requirement analysis, current and future technology specification. However, the presented solution had to take into account some constraints and limitations, implied by the specific nature of hardware (Mark5) used in radio astronomy, and VLBI in particular. This was described in chapter 5

For now we aim to connect to the current operational system Mark5. It poses some specific problems, which may disappear when other systems can be employed (e.g. those developed in FARBIC WP1).

6.1 eVLBI – design limitations

The Mark5 is a disk based VLBI data recording and playback system capable of handling 1 Gbps data rates. It is widely used at the telescopes participating in the EVN. A software correlator running on some grid node should therefore conform to this interface of the radio-telescopes to the outside world.

Before describing ways to get the observed data onto some grid node for further processing first the relevant data transfer modes of the Mark5 computer will be described. For the current e-VLBI practice a Mark5 computer is connected to the radio-telescope through a formatter. The data enters at the I/O board and is transmitted to another Mark5 computer. At the receiving Mark5 data can be recorded on disks or an I/O board can transmit the data directly to the EVN correlator. The appropriate Mark5A commands have to be activated on both sides.

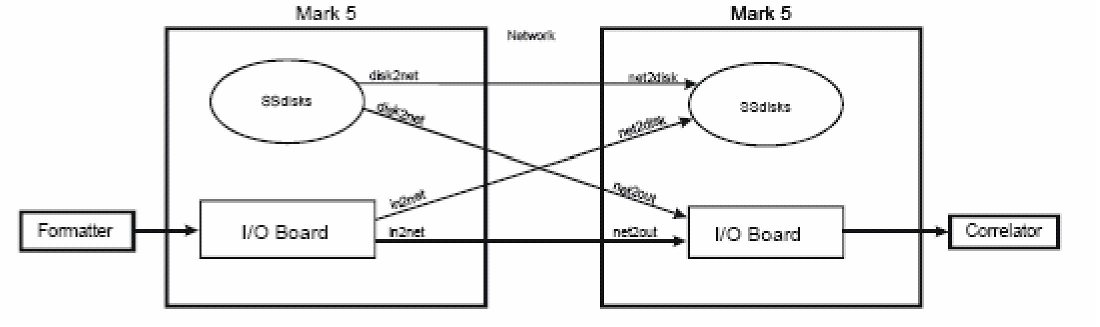


Figure 14. Mark5 to Mark5 transfer through network

Usually a grid node does not have a Mark5 computer at the receiving side but UNIX/Linux like computers. There are two ways of getting the data onto a UNIX/Linux machine. First the data is recorded on disks in the Mark5 computer and later it is played back. In this case the disk2net command puts the data to the network and Net2file gets the data from the network.

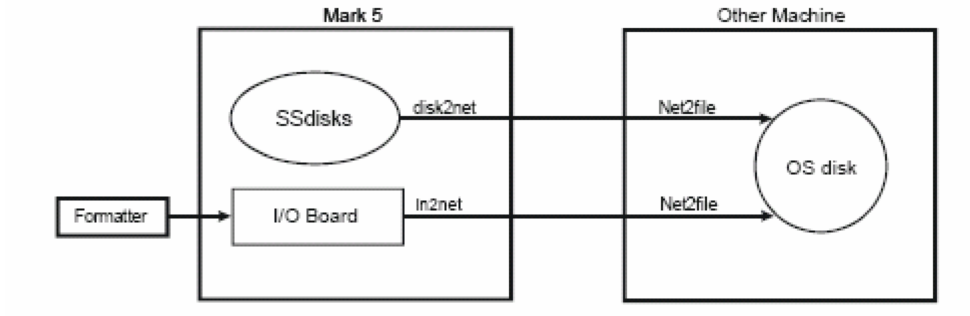


Figure 15. Mark5 to file transfer through network

The data stream can also be put on the network by the Mark5 during the astronomical observations by in2net. At the receiving side Net2file gets the data from the network onto a disk.

So somewhere a UNIX/Linux file server is necessary, either at the telescope site or at a remote grid node. In the first case a powerful file server at the telescope site is required, and it operates as a data buffer. The data is sent over the network to a compute node. The second case requires a point to point internet connection between the Mark5 system and a remote file server. Buffering can optionally be done at the telescope site on the Mark5 system itself. The connection between radio telescope and appropriate file server must be set up manually by telescope operator, which receives the relevant information from the experiment description. The connections will be defined by VLBI operator using the Workflow Manager application. It is described in more detail in the next paragraph.

On of the problems lies in the effective switching of the routing path from the telescopes (there is no automated way to do that, it requires manual adjustment of networking equipment). Furthermore, once the data acquisition has started the data destination point cannot be changed.

This limitation has a significant impact on the design, because the initial design assumptions were based on the idea of direct and dynamic routing of the data streams from the radio telescopes to the computational nodes. Considering the above, this is technically impossible with the current hardware. Unfortunately, the routing problem is difficult to solve at this point, as it requires the hardware modification at the telescopes sites. A workaround is necessary, and is presented below in this section.

The whole process is presented in a more general view in the figure below:

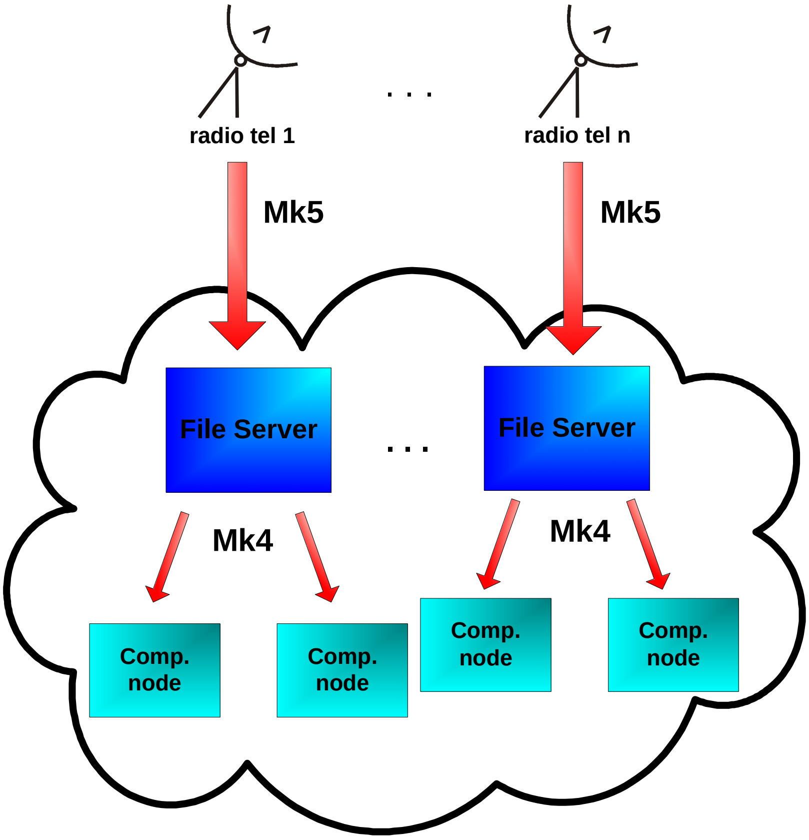


Figure 16. The overview of the solution with file servers

Next figure presents the more detailed view on a single telescope:

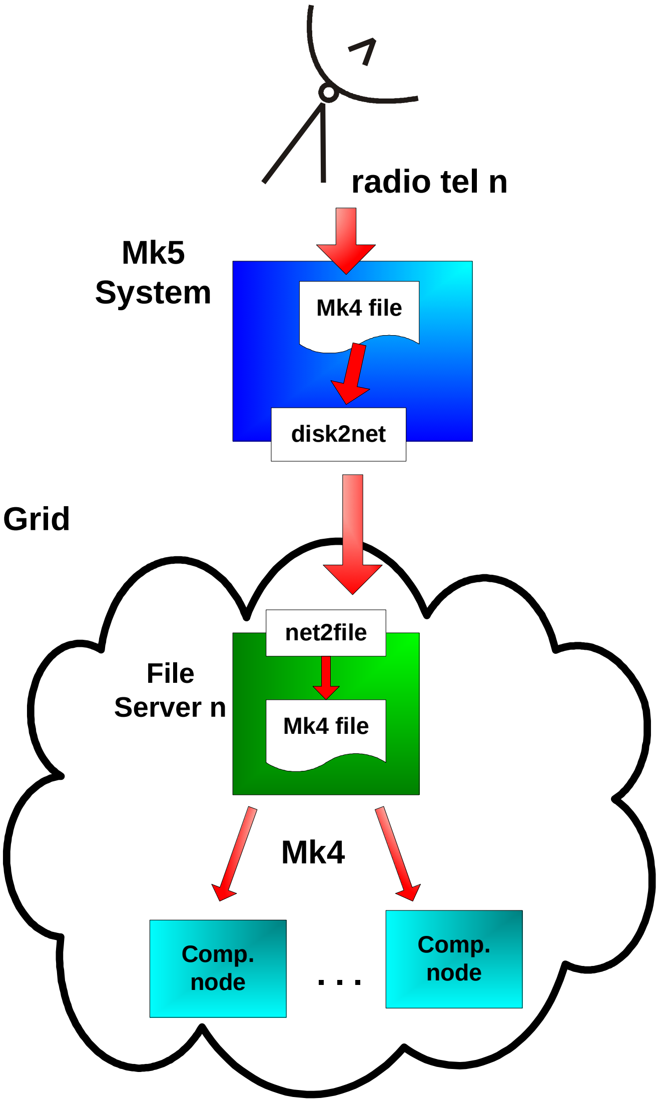


Figure 17. The close-up view on the file server solution

6.2 System architecture

The final system design and architecture is a result of careful problem analysis and requirements specification. It would be impossible to present a successful solution to such complicated problem without the help and advice with radio astronomers (JIVE staff in particular). The proposed architecture for meeting the FABRIC goal is based on the current VLBI and e-VLIBI status (chapter 2) and is presented in the diagram below:

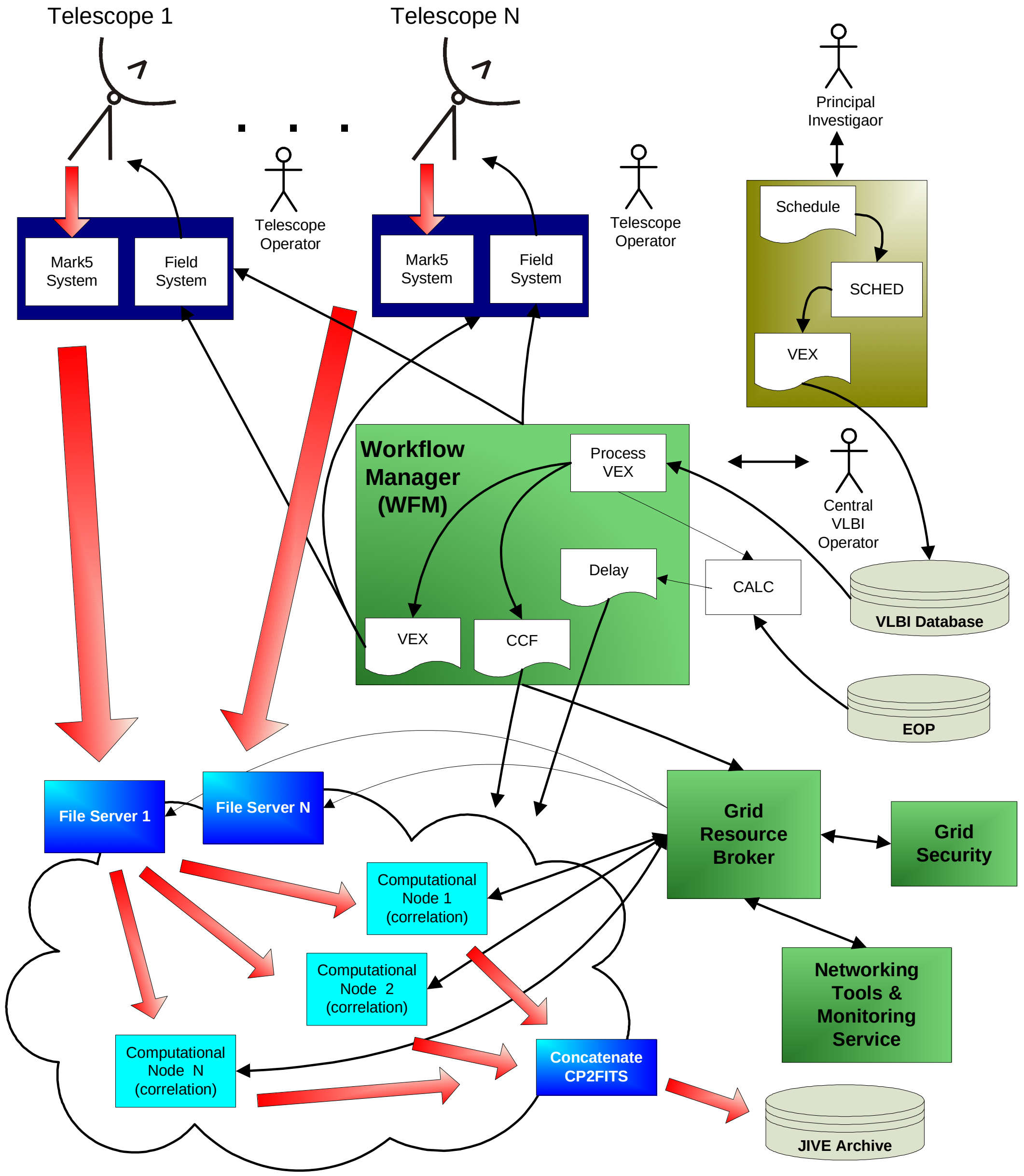


Figure 18. System architecture for distributed broadband correlation

The whole process can be divided into 4 major parts:

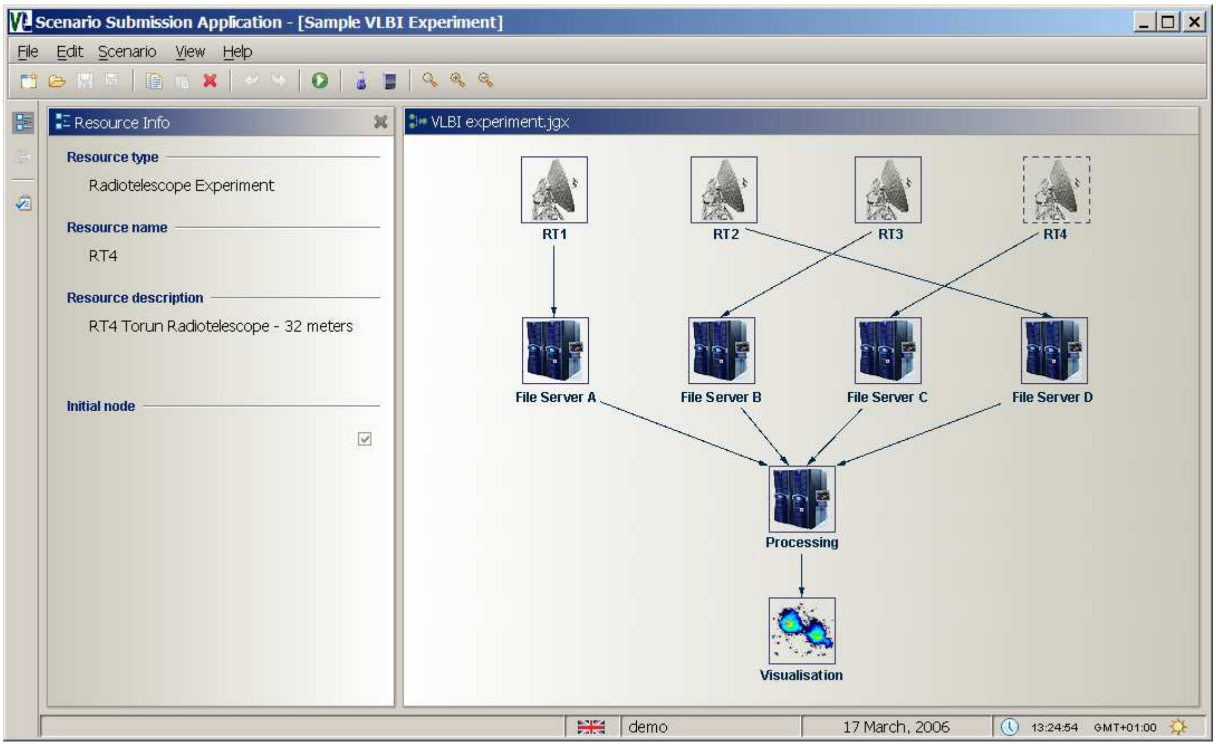
Part I. Experiment preparation.

The A Principal Investigator (PI) will be granted observation time on various VLBI telescopes. Before the actual observation can start the PI will create a schedule file containing all the details of the observation, like what source to observe, which telescopes to use and when to observe. This schedule file will be used to create a VEX file with the SCHED application. A central operator will be notified of the experiment. Then the PI will have to wait until the data arrives in the JIVE archive.

Part II. Experiment set-up.

The central operator will use the Workflow Manager (WFM) to process the VEX file. The WFM will be a graphical Java application, downloaded and executed from the VLBI web portal. The information contained in the VEX file will be displayed in the appropriate forms, in which the experiment control parameters can be verified and modified if necessary. The central operator will be able to set up more eVLBI parameters, according to his domain knowledge and some specific experiment conditions. Another important role of the WFM (and central operator) will be to associate specific file servers with radio telescope locations and create a workflow for the post-experiment distributed data correlation. The description of connections between file servers and radio telescopes will be sent to telescope operators together with experiment definition (VEX file). It will be the operators responsibility to manually set up the routing from their sites to the file servers before the experiment starts. Additionally, the operator will build an experiment workflow graph with virtual correlation tasks, which will divide the correlation process logically, on a number of processing instances. Each of the instance will be treated as a single and independent sub-task and can be allocated on a different computational node.

The WFM will also calculate the necessary delay tables before the correlation takes place by calling the external program CALC and Earth Orientation Parameters (EOP). CALC is a standard application developed by geodesists to accurately determine positions on earth.



Scenario Submission Application - [Sample VLBI Experiment]

Resource Info

Resource type
Radiotelescope Experiment

Resource name
RT4

Resource description
RT4 Torun Radiotelescope - 32 meters

Initial node
☑

demo | 17 March, 2006 | 13:24:54 GMT+01:00

Figure 19. An example view of WFM application (model)

Part III. Experiment start.

After the central operator's work will be done, a new VEX file will be created and sent to the telescopes participating in the observations. The WFM will also notify the telescope operators that a new experiment is scheduled. As it was mentioned before, the WFM will also send the planned routing information between telescopes and file servers and it will also contact the dedicated Grid Resource Broker to allocate the necessary computational nodes for each of the correlation tasks, and for the necessary broadband network connections between the file servers and computational nodes. The resource broker will use the mechanisms described in section 4 (networking) in order to plan and schedule the optimal route for the data transfer between fileservers and nodes.

In the meantime, the telescope operator will load the VEX file in the Field System which controls the telescope and in the Mark5 system which records the data. At the start of the experiment the allocated compute nodes should be ready and waiting for data. The WFM has already send the Correlator Control File (CCF) and Delay file to the compute nodes. The data recorded by the Mark5 system will be sent to the allocated compute nodes via the file servers (see section 6.1 for details). Additionally, the WFM may also provide the operators (as well as principle investigator) with the possibility of verifying the status of the entire eVLBI experiment (by displaying current status of all radio telescopes and the status of grid broker).

Part IV. Distributed correlation.

One of the goals of the FABRIC project is to enable distributed correlation using grid facilities. In section 5.2 was concluded that the implementation of the first version of the software correlator will be based on the off-line time sliced correlation scenario. Other software correlator architectures and implementations will bedeveloped and tested in the FABRIC project.

As it is mentioned above, the presented solution is a result of careful analysis of problem domain and radio astronomy hardware specification. It was created with a great help from JIVE staff, but the nature and complexity of this problem still raise many questions and uncertainties that may have to be solved only by the experimenting and creating working models and prototypes. Therefore, in the end the final, implemented system architecture might look different is some areas.

7 Conclusion

The main goal of this document was to analyse how VLBI experiments can be supported by the grid solutions. In the second chapter the current state of the art in the range of VLBI and e-VLBI operations has been presented. Next, characteristic of several grid brokers has been described. The grid broker is a module which plays a significant role in a e-VLBI system. It is responsible for optimizing data transfer from radio telescopes and for launching correlation process in an efficient way. Security and data transport solutions for grid environment which can be used for the proposed system are discussed in the next chapter. These aspects are also analyzed from efficiency point of view. Higher level tools like grid brokers need up-to-date information to make a reasonable decisions. How to monitor the network and pass the current status to the broker is a topic of chapter 4.1. In chapter 4.2 aspect of assuring appropriate quality of service in broadband networks is analyzed. Some practical approach to network management is described basing on experiences in Clusterix project. Next, concept of integration of the software correlator (developed in other task of EXPReS project) with the grid environment is proposed. Finally, overall design of the e-VLBI system which takes advantage of grid solutions is outlined in section 6. Diagrams of the architecture and work-flow have been presented.

Definitions, abbreviations, acronyms

Condor – Workload management system for compute intensive jobs. Condor is the product of the Condor Research Project at the University of Wisconsin-Madison (UW-Madison).

DRMAA – Distributed Resource Management Application API (DRMAA) is a high-level Global Grid Forum API specification for the submission and control of jobs to one or more Distributed Resource Management (DRM) systems within a Grid architecture.

GAHP – Grid ASCII Helper Protocol (GAHP) provides services to a variety of grid interfaces (including Globus Toolkit) via simpe ASCII based protocol.

GASS – The Globus Toolkit's Global Access to Secondary Storage (GASS) service provides mechanisms for transferring data to and from a remote HTTP, FTP, or GASS server.

GRAM – The Grid Resource Allocation and Management (GRAM) protocol supports remote submission of a computational request (for example, to run program P) to a remote computational resource, and it supports subsequent monitoring and control of the resulting computation.

GridWay – GridWay is an open-source component for meta-scheduling in the Grid Ecosystem

GRMS – Open source meta-scheduling system, developed under the GridLab Project, which allows developers to build and deploy resource management systems for large scale distributed computing infrastructures

MP Synergy – Job scheduler across heterogeneous distributed systems

OpenPBS – OpenPBS system is to provide additional controls over initiating or scheduling execution of batch jobs; and to allow routing of those jobs between different hosts. The batch system allows a site to define and implement policy as to what types of resources and how much of each resource can be used by different jobs

References

BBRS2004 Bartosz Baliś, Marian Bubak, Wojciech Rząsa, and Tomasz Szepieniec "Efficiency of the GSI Secured Network Transmission", International Conference on Computational Science 2004, Kraków, Poland, June 6-9, 2004

BWCTL – Bandwidth Test Controller, http://e2epi.internet2.edu/bwctl/

Cacti – The complete RRD-tool based network graphing solution, http://cacti.net/

CALC – http://gemini.gsfc.nasa.gov/solve/

Condor – http://www.cs.wisc.edu/condor/

DRAC – http://www.nortel.com/drac/

DRAGON – http://dragon.maxgigapop.net/twiki/bin/view/DRAGON/WebHome

E2ePIPEs – Internet2 End-to-End Performance Initiative http://e2epi.internet2.edu

GÉANT2 – GÉANT2 network homepage http://www.geant2.net/

GN2 – The IST project GN2 – GÉANT2 (contract number 544082) http://www.dante.net/server/show/nav.1119

GridLab – http://www.gridlab.org/

GridNM – Yee-Ting Li, "GridNM – Grid Network Monitoring Infrastructure", 2002

GridWay – http://www.gridway.org/

Internet2 – http://www.interne2.edu

JRA1 – J. Boote, E. Boyd, M. Campanella, J. Durand, S. Evett, M. Glowiak, A. Hanemann, R. Karch, S. Kraft, L. Kudarimoti, O. Kvittem, R. Łapacz, A. Liakopoulos, L. Marta, J. Metzger, M. Molina, M. Swany, S. Trocha, S. Ubik; GN2 project deliverable: "DJ.1.2.1 - GEANT2 General Monitoring Framework Design",

http://www.geant.net/

JRA1DJ1.2.2 – J. Boote, J. Durand, M. Głowiak, A. Hanemann, V. Jeliazkov, L. Kudarimoti, P. Louridas, R. Łapacz, L. Marta, N. Simar, M.Swany, S. Trocha, I. Tsompanidis, V. Venus; GN2 project deliverable: "D.J.1.3.1 - Phase I - Implementation Report", http://www.geant.net/

JRA3DJ3.2.2 – G. Alyfantis, M. Büchli, E. Camisard, M. Campanella, V. Gazis, G. Ivanszky, E. Kenny, S. Muyal, J. Radil, R. Nuijts, S. Paskalis, G. Priggouris, E. Robles, L. Serrano, A. Sevasti, Ch. Tziouvaras; GN2 project deliverable: "DJ3.2.2: Initial Review of Technologies Related to the Provision of Bandwidth-on-Demand (BoD) Services", http://www.geant.net/

Mark5 – http://www.haystack.mit.edu/tech/vlbi/mark5/index.html

MP Synergy – http://www.ud.com/products/hpcsynergy.php

MRTG – The Multi Router Traffic Grapher, http://oss.oetiker.ch/mrtg/

MUPBED – The IST project MUPBED (contract number 511780) – http://www.ist-mupbed.org

Network performance – B. Lowekamp, B. Tierney, L. Cottrell, R. Hughes-Jones, T. Kielmann, M. Swany, "A Hierarchy of Network Performance Characteristics for Grid Applications and Services", 2004 GGF Network Measurements Working Group

Network performance – R. Wolski, N. T. Spring, J. Hayes, "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing", Journal of Future Generation Computing Systems,Volume 15, Numbers 5-6, pp. 757-768, October, 1999

NREN – National Research and Education Network

NWMG – The Network Measurements Working Group (NMWG) of the Global Grid Forum, http://www-didc.lbl.gov/NMWG/

OMNINET – http://www.icair.org/omninet/

OWAMP – S. Shalunov, B. Teitelbaum, "A One-way Active Measurement Protocol Requirements", 2003 Internet2 draft

perfSONAR – Performance focused Service Oriented Network monitoring Architecture, http://www.perfsonar.net/

PHOSPHORUS – The IST project PHOSPHORUS (contract number 034115); this project is starting from Oct. 1, 2006 – http://www.ist-phosphorus.eu

QoS – Quality of Service

SEQUIN – The IST project SEQUIN (IST-1999-20841) – http://www.dante.net/sequin

SEQUIN D2.1 – Campanella, M., Chivalier, P., Sevasti, A., Simar, N., "Quality of Service Definition", SEQUIN Project, Deliverable 2.1, March 2001

SOA – Service-Oriented Architecture (SOA), http://java.sun.com/reference/soawebservices/

SURFnet – http://www.surfnet.nl

UCLP – http://www.canarie.ca/canet4/uclp/

WebService – Web Services activity, http://www.w3.org/2002/ws/

Contact Information

All authors affiliation:
Poznań Supercomputing and Networking Center
ul. Noskowskiego 10
61-704 Poznań, Poland

URL: http://www.man.poznan.pl
Tel. (+48 61) 858-20-00
Fax (+48 61) 852-59-54

Marcin Okoń	marcin.okon@man.poznan.pl
Dominik Stokłosa	d.stoklosa@man.poznan.pl
Marcin Lawenda	lawenda@man.poznan.pl
Norbert Meyer	meyer@man.poznan.pl
Marcin Garstka	marcinga@man.poznan.pl
Szymon Trocha	szymon.trocha@man.poznan.pl