User Interaction and Workflow Management
in Grid enabled e-VLBI Experiments

Ł. Dolata, M. Okoń, D. Stokłosa, Sz. Trocha,
N. Meyer, M. Stroiński, D. Kaliszan, T. Rajtar and M. Lawenda

Poznań Supercomputing and Networking Center
10 Noskowskiego Street, 61-704 Poznań, Poland
e-mail: office@man.poznan.pl

Abstract: The purpose of this paper is to present the design of the integration of radio-astronomical VLBI experiments with the Grid environment. The current status of the VLBI process is described, and the main part of this document presents the proposed architecture for the development of the next generation e-VLBI system. This design was created for the purpose of the EXPReS project [12], partially funded by the European Commission (FP6 – IST).
Key words: VLBI, Radio Astronomy, Grid Computing, Virtual Laboratory, Distributed Systems

I. INTRODUCTION

Radio astronomy is a subfield of astronomy that studies celestial objects at radio frequencies. Radio astronomers use different techniques to observe objects in radio spectrum. The most common and well-known device used in radio astronomy is a radio telescope pointed at an energetic radio source to analyse what types of emission it makes. Radio telescopes may need to be extremely large in order to receive signals with low signal-to-noise ratio.

Since 1970s, telescopes from all over the world have been combined to perform Very Long Baseline Interferometry. Data received at each antenna is paired with timing information, usually from a local atomic clock, and then stored for later analysis on a magnetic tape or hard disk. At that later time, the data is correlated with similarly recorded data from other antennas to produce a resulting image [7]. Resolution of images depends on the overall size of the network (the maximum separation between telescopes), and the sensitivity depends on a total collecting area of all telescopes involved. After data from all telescopes is collected, it must be put together and correlated in order to produce the final observation result. In Europe it is done by The European VLBI Network [9] which was formed in 1980 by a consortium of five major radio astronomy institutes in Europe (the European Consortium for VLBI). In

1993, the Joint Institute for VLBI in Europe (JIVE) was created with the Netherlands Foundation for Research in Astronomy (Dwingeloo) acting as the host institute. It provides both scientific user support and correlator facility [8].

This functionality has been achieved by constructing a massively parallel, purpose-built 'supercomputer' – usually referred to as Data Processor or Correlator. A decade ago, a 20.000 node PC cluster would have been required to match the processing power of the EVN MkIV Data Processor at JIVE. However, this number has been falling rapidly as processor capabilities continue to develop. The networking infrastructure is also getting more capable of dynamically handling very large data transfers in long periods of time. This has led to the introduction of the e-VLBI system, with the main difference being data sent via network to the central processing facility at JIVE. This has drastically shortened the time required to obtain final results from the actual observation. However, in the e-VLBI approach a new bottleneck has emerged – a single central data processor at JIVE. To solve this problem, a concept of software-based distributed correlation embedded in a Grid Computing environment has been introduced [10].

The Grid approach to e-VLBI required the creation of some form of a user interface, as a typical approach based on creating a text-based single control file for the experiment was not sufficient. In this case, a user should specify

a set of Grid-specific parameters and define required resource types together with their specific connections creating e-VLBI workflow. An interface to network monitoring modules should also allow to modify all required parameters and convert experiment parameters into software-correlator compatible data automatically. In the next paragraphs the basics of the Grid – the e-VLBI architecture is presented and followed by a detailed explanation of the Workflow Manager application with its GUI, functionality and inner workings.

II. e-VLBI AND GRID INTEGRATION ARCHITECTURE

The following diagram presents the general design of the e-VLBI system.
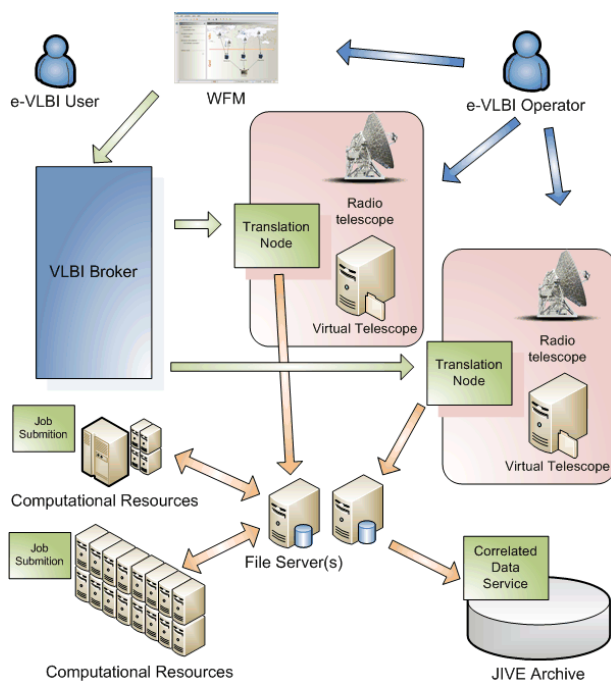

Fig. 1. The general architecture of a Grid e-VLBI system

The e-VLBI experiment preparation with a Grid-based correlation consists of the following steps:

- An e-VLBI user creates an experiment description (VEX file) using the SCHED application. This process is no different than in regular VLBI observations.
- The VEX is processed in the Workflow Manager Application (WFM) by an e-VLBI operator. Experiment control parameters can be verified and modified if necessary.
  A central e-VLBI operator will be able to set up more parameters and will also be able to specify if

the data for the experiment should be streamed live from telescopes or read from prerecorded files (so called "Virtual Radio Telescope"). In any case, Translation Node (TN) modules are responsible for converting the data to a useable format and sending them to specified file servers. TN – File Server pairing is done by the e-VLBI operator based on information from Network Monitoring modules (not shown in Fig. **1**). The network information will include information such as throughput, delay, packet loss, etc., between real or virtual radio telescopes (defined in the VEX file) and File Servers (defined in the WFM by the e-VLBI operator). Having the data on File Servers, the e-VLBI operator also defines the workflow for the distributed data correlation. The computational resources can be specifically chosen from a defined pool of resources or dynamically discovered and assigned by a VLBI Broker. After the e-VLBI operator has finished work, an updated VEX file is created and sent to the telescopes participating in observations. The WFM also notifies the telescope operators that a new experiment is scheduled, and sends experiment information to the VLBI Broker module.

- Telescope operators, together with the central e-VLBI operator, are responsible for loading the VEX file in the Field System which controls the telescope, and in the Mark5 system which records the data. In case of Virtual Radio Telescopes they are responsible for configuring the Translation Nodes to stream the data from pre-recorded data sources.

a) e-VLBI experiment execution – VLBI Broker architecture

The VLBI Broker is the central element of the eVLBI system providing the centralised control of the entire experiment. The Broker module processes the experiment definition submitted from the WFM application and forwards tasks' description to telescope sites (Translation Nodes). This module is also responsible for coordination and submission of computational tasks with distributed correlation.

General VLBI Broker module architecture is presented in Fig. 2, together with data structures that are exchanged between modules.

Data structures explained in more details:

Between the WFM application and the VLBI Broker:

- **VEX file** – a file created by a radio astronomer with a complete description of a VLBI experiment (information such as telescope names, experiment time, control parameters, etc.).

- **CCF file** – a Correlator Control File generated by the WFM based on a VEX file. CCF is required to perform software correlation. Some of its parameters can be derived automatically from the VEX file, while the other ones have to be modified by the VLBI Broker as experiment progresses.
- **Workflow Description** – defines which radio telescopes take part in the experiment and what file servers and computational resources were chosen, as well as connections between them defining the flow of data during the observation and distributed correlation.
- **Experiment Description** – additional experiment parameters not present elsewhere.

Between the VLBI Broker and the Translation Node:
- **experimentName** – a common name for an experiment used to identify it by the VLBI operator and end users
- **start&end Time** – the experiment (VLBI observation) start and end time
- **brokerLocation** – an address for return communication with the broker module
- **telescopeName** – specifies a telescope, should be set to a corresponding radio telescope related to given Translation Nodes or set to indicate a "virtual radio telescope"
- **gridFtpLocation** – location of a File Server to which a Translation Node should send streaming data

Between the Translation Node and the VLBI Broker:
- **translationNodeId** – identification of TN sending data
- **chunkId** – the id number of data chunk
- **chunkSize** – the size of data chunk
- **start&endTime** – the observation data start and end time
- **chunkLocation** – the path to a file containing data on a File Server

Between the VLBI Broker and the Computational Node:
- **CCF** – the correlator control file created by WFM, and completed with additional parameters by the VLBI Broker
- **computational job** – other parameters required to run a computational task on a given machine

Between the Computational Node and the VLBI Broker:
- **job completed** – information on the task id and its status (completed, failed)
- **results' location** – the path to computed output data location (if applicable)

Between the VLBI Broker and the Correlated data service:
- correlated chunks location – a list with the location of all correlated data chunks for a given VLBI experiment.
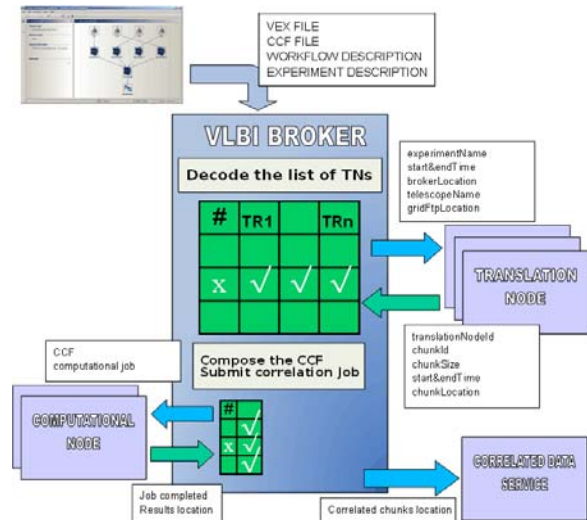

Fig. 2. The VLBI Broker Architecture and Data Structures

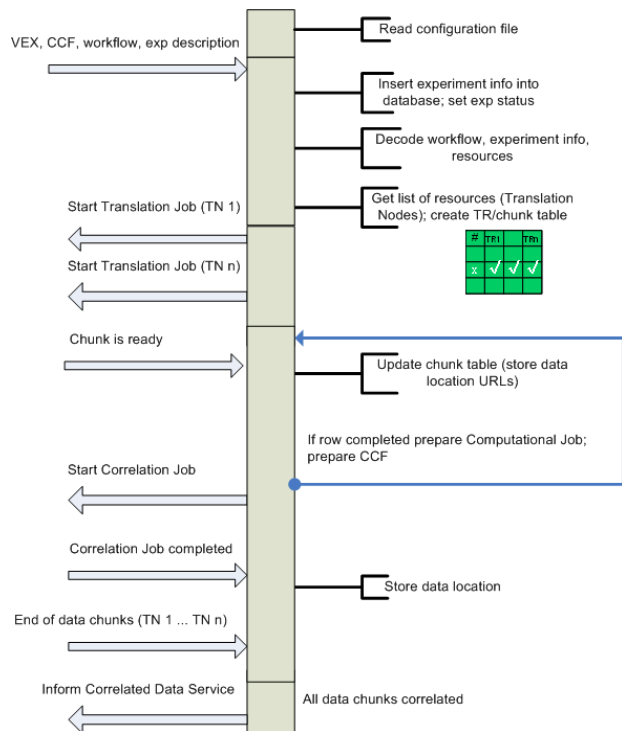The internal e-VLBI control flow in the VLBI Broker was presented in Fig. 3.

Fig. 3. VLBI Broker Data Processing

Upon its start the VLBI Broker reads its control parameters and sets up its environment. Then it awaits for a new experiment. Once new experiment data is received, the incoming data is decoded and experiment information is recorded in the internal database. The database is also used for storing the current experiment status and other experiment-related data. Data from the database would be used in case a broker restarts to restore its internal state.

Once the experiment data is decoded, the list of resources and the table chunks/TNs are created. In this table the information of incoming data chunks are recorded.

After all preparation work all Translation Nodes involved in the experiment are notified some time prior to the experiment start. This time is a configurable option in the broker's configuration settings.

When all TNs are notified, the VLBI Broker starts passive waiting for incoming notifications. When a notification that a particular data chunk is ready arrives, the table is updated. When a single row of a given chunk number from a sequence is completed (which means that all TNs delivered certain frame of data), the correlation job is prepared and sent to the correlation module on a given computational resource for an execution.

After some time, information about completed correlation tasks starts to arrive. The information about the location of correlated data is stored, and when all data chunks are correlated, the notification is sent to the Correlated Data Service with the location of all chunks. The service is responsible for retrieving the data and merging it into the final correlated product.

III. WFM AND WORKFLOWS

The e-VLBI system allows astronomers to plan, execute and monitor their observations in the form of so called workflows. However, the e-VLBI experiment is not only the observation itself. The e-VLBI experiment consists of a definition of storage elements, a definition of data flows or a definition of computation resources, etc. Such e-VLBI workflow has to be created for each observation. The Workflow Manager Application (WFM) has been created to allow users to design and execute their observation workflows easily.

a) Graphical User Interface

The Workflow Manager Application (WFM) – the main interface between users and the system has been constructed as a Java window-based application. The advantage of Java technology is that standalone Java software

applications can be deployed with a single click over the network. Moreover, the application can be run at every computer connected to the Internet and equipped with a web browser. The following chapter presents the main features of the WFM application.
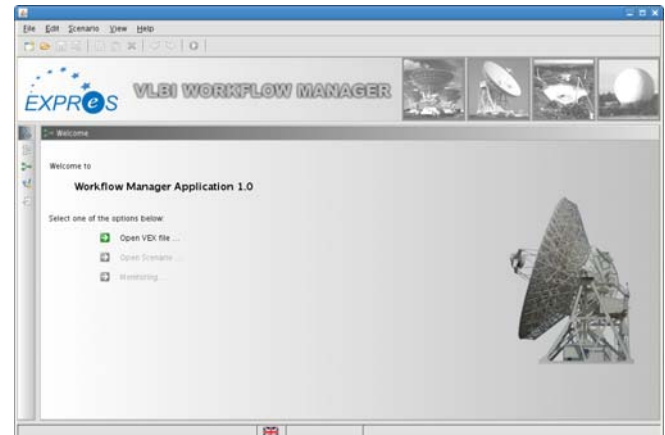

Fig. 4. WFM – main view

The Workflow Manager Application (see Fig. 4) is divided into several working panes: *Information Pane, Design Pane* and *Log Message Pane.* Each view has its own place in the application window. Users can manage the visibility of different views adjusting windows layout.

The Information Pane is the only "dynamic" pane, which means that it presents different information based on the application state. This allows users to get the most important (more general) data quickly and without tiring and time consuming interactions.
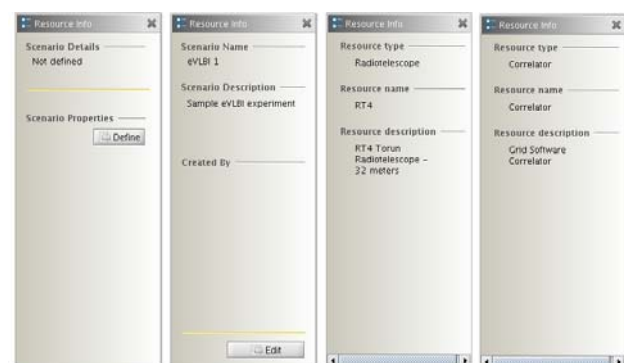

Fig. 5. Information panes – examples

Fig. 5 presents several combinations of information panes. They contain data such as a short summary of the current experiment or resources: radio telescope and correlator.

b) Workflow creation

Observation workflow can be created in the WFM application based on an observation schedule. An observation schedule has to be created by an astronomer on a regular basis using the SCHED application. The description of the SCHED application is not in the scope of this article. Let us assume that the observation schedule created by this application is written in the form of a 'VEX file' which will be used by the e-VLBI system for further processing.

Because of the complexity of the VEX file, the e-VLBI system is based on the subset of the VEX file which contains all parameters required by the software correlation module. The parameter's set is stored in the JSON format in a file called a correlator control file (CFF). The CCF file controls the behaviour of the software correlator and can be edited and modified in the Workflow Manager Application.

The workflow creation can be divided into the following phases:

- Opening and loading a VEX file,
- Converting the VEX file to the CCF file,
- Definition of workflow parameters,
- Definition of file servers,
- Definition of correlation nodes,
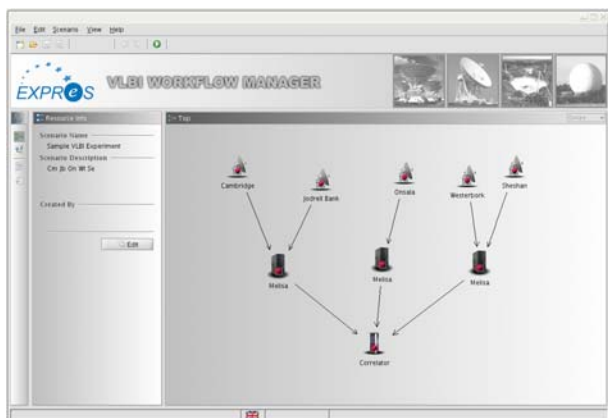- Definition of data flows between components.

Fig. 6. Sample VLBI scenario

The Design Pane (see Fig. 6) is used by the VLBI operator for the VLBI scenario creation and management. It allows users to construct their observation workflow and specify file servers which will be responsible for handling data streams from radio telescopes. It also allows to add correlation nodes and define correlation parameters. Finally, all the workflow components can be linked with each other, which implies data flow and relationship between them. Moreover, each workflow component is described by a certain set of attributes that can be set by the VLBI opera-

tor. The pane itself is divided into two parts: *VLBI* and the *Grid* part. The *VLBI pane* is reserved only for radio telescopes. The system analyses the CCF file and draws all radio telescopes which take part in the VLBI experiment.

The VLBI part cannot be changed. This means that it is not allowed to add or remove radio telescopes. This limitation has been introduced so the list of stations defined in the CCF file corresponds to the stations drawn in the Workflow Manager Application. The *Grid part* is intended to be set up by the VLBI Operator who is responsible for defining a file server layer and a correlation node layer. However, the number of resources used and their origin depends on the VLBI operator, and should be based on experience gained from previous experiments. The Workflow Manager Application supports the process of defining data flows between workflow blocks by presenting a list of best connection sets between components based on network statistics. More information about network measurements can be found in the *Network Monitoring* chapter.

c) CCF editor

As it was stated before, the Mk4 hardware correlator has been replaced with a software version of the correlation module in the e-VLBI system. The correlation parameters are stored in the JSON format text file called a correlator control file (CCF).
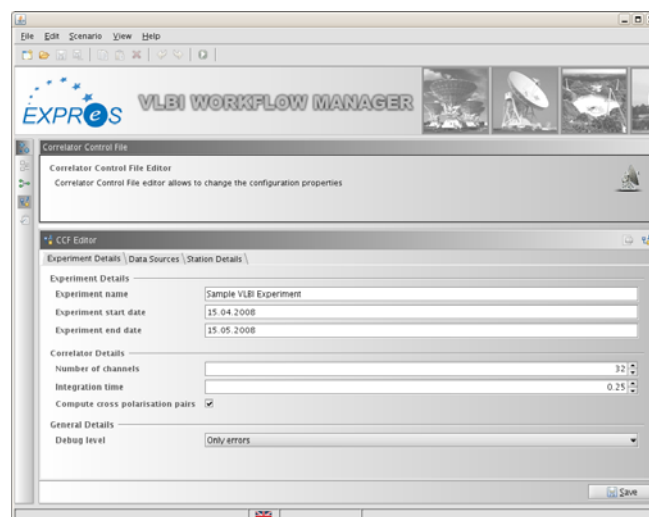

Fig. 7. CCF Editor

A special CCF perspective has been created in the WFM application which allows to change correlation parameters such as the experiment name, start and end time, the number of frequency channels or array of frequencies at which the telescopes observe (see Fig. 7).

IV. NETWORK MONITORING

In the e-VLBI architecture, in order to automate and make the process faster, the data is transmitted over with a high-speed reliable network and correlated using distributed GRID resources. The limited number of GRID computational and network resources as well as the need to monitor the service meet in a demand for a network monitoring system. The resource allocation mechanism and workflow management software will distribute the correlation over GRID computers where sufficient computational resources are available. In order to control the data streams as well as parameters of GRID links established between distributed nodes of the correlator, a dedicated network performance monitoring software module is necessary to provide support for data management and insight view into the e-VLBI network traffic.

Performance monitoring applications ensure availability and efficiency within the infrastructure by monitoring various performance metrics like latency, jitter or packet loss. To unify the frequently distributed task of monitoring, the concept of a Service Oriented Architecture (SOA) offers the ability for specialized, autonomous services to join under a common access scheme. Thus, it is possible to separate the roles of monitoring, storage, processing, and visualization of data into specialized service instances. In GRIDs, with the widespread of multi-domain applications, the quality of service becomes important. Network connections have to be monitored and managed to satisfy all the requirements and to provide appropriate network performance parameters to research communities using specialized applications based on distributed computations like the aforementioned workflow management software.

The e-VLBI requirements for network monitoring fit best into the deployment of perfSONAR [1] which may be adopted for GRID resource management and QoS provisioning. perfSONAR (Performance-Focused Service-Oriented Network Monitoring Architecture) is a distributed performance monitoring architecture designed to aid in the discovery, troubleshooting, and solution of network performance problems potentially among numerous networks. This framework is comprised of collaborating entities meant to collect, store, analyze, and deliver data using well-defined protocols. perfSONAR is the subproject Joint Research Activity 1 of the GN2 [2] EU-founded project and aims at providing a framework for performing multi-domain measurements in the European Research Network (GÉANT2) and the connected National Research and Education Networks (NRENs).

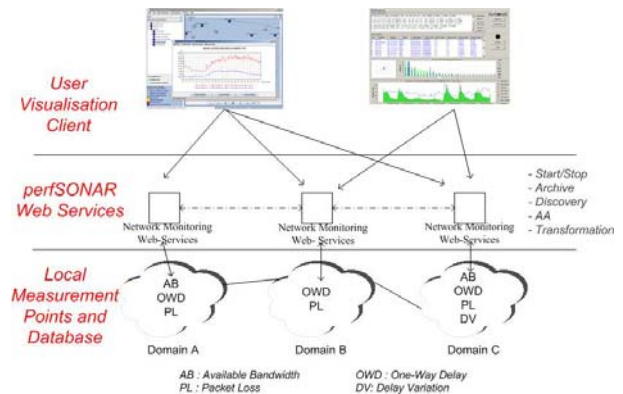The general monitoring infrastructure implemented by perfSONAR is illustrated in Fig. 8

Fig. 8. perfSONAR architecture

The Measurement Points (MPs) are the base layer of this framework and have the responsibility of measuring storing different network metrics as well. The Service Layer is the middle layer of the system that is divided across administrative boundaries. This layer may aggregate, filter or buffer data and can be seen as a set of data archives and transformers. On top of it, the User Interface Layer consists of visualization tools which present the data in customizable ways. In addition, this layer allows users to perform tests using services available at the lower layers of the framework. More than a dozen of perfSONAR installations are currently available and they have enabled the first view of the European networks, making the first step towards effective access to the available data placed in various administrative domains.

We plan to provide network monitoring support for workflow management in GRID-enabled e-VLBI experiments with a software module which will make use of the existing and newly installed perfSONAR monitoring and measurement nodes. Open architecture of perfSONAR allows external applications for making use of data collected by the perfSONAR system through the NMWG XML schema. The measurement framework will collect performance data (bandwidth utilization and Round Trip Time) using perfSONAR CommandLine MP service installed in each involved domain and covering network paths between GRID computational resources. The level of coverage of the e-VLBI network by perfSONAR measurements will significantly impact the ability to take optimal decisions. Ideally, all the used domains should deploy the system.

a) Implemented solution

The main purpose of developing the Express Network Monitor is to provide an optimal connection between the given Express resources that should assure the fastest pos-

sible and reliable data transfer within the e-VLBI GRID environment. The module performs on-demand tests, analyses network performance along a transporting path, and provides the information about end point pairs for e-VLBI workflow management system. The overall Express Network Monitor architecture is shown in Fig. 9.
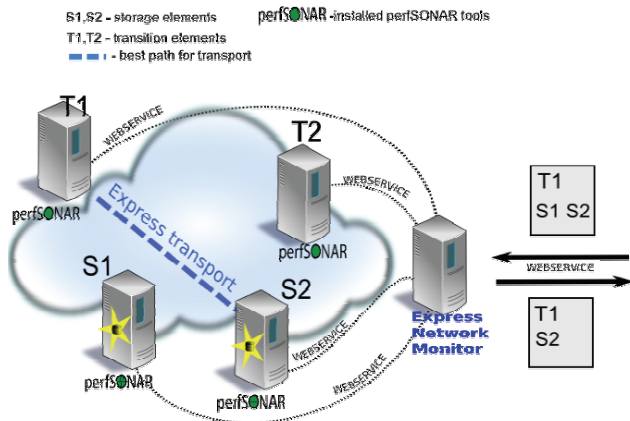

Fig. 9. The Express Network Monitor architecture

The main part of the Express Network Monitor system is the **Express Network Monitor** application. This application works as a request point for the e-VLBI workflow management system. It controls all configured measurements nodes, requests all needed on-demand tests, analyses the results and provides an ordered list of Express resources pairs. Communication is based on the web services technology and JSON protocol.

In order to provide monitoring capability, each Express data transport resource has to deploy the following tools:

- **Iperf** [3] is a tool to measure the maximum TCP bandwidth, allowing the tuning of various parameters and UDP characteristics. Iperf reports bandwidth, delay jitter, and datagram loss.

- **Bandwidth Test Controller (BWCTL)** [4] is a command line client application and a scheduling and policy daemon that wraps the Iperf application. Currently BWCTL wraps *Iperf* by actually executing the Iperf command line program on the system. The bwctl client application works by contacting a bwctld process on the two test endpoint systems. The client can arrange a test between two servers on two different systems. If the local system is intended to be one of the endpoints of the test, a local bwctld is not required, bwctl will detect that there is no local server and execute the required bwctld functionality directly. The bwctld manages and schedules the resources of the host on which it runs.

- **perfSONAR Command Line Measurement Point (CL-MP) 2.0** [5] is used to integrate any command-line measurement tool into the perfSONAR framework. Currently the application has support for ping, traceroute, bwctl, and owamp tools. The CL-MP 2.0 provides the web service interfaces and NMWG [6] messages parsers to be fully perfSONAR-capable.

The Express Network Monitor waits for requests from the e-VLBI workflow management system. Each time an Express transport session is scheduled, the system creates a list of transition nodes and storage elements related to this session. The unordered list is then sent to the Express Network Monitor together with the value of the requested bandwidth for transport. The Express Network Monitor creates a measurement session and starts to engage on-demand tests. At first, the stage system performs a number of parallel round trip times (RTT) measurements between all transition nodes and each storage element. A list of results is sorted in the ascending order. The system performs a single bandwidth utilization test to storage nodes in the order resulting from the earlier list of measured RTTs. If the measured achievable bandwidth is higher than the requested bandwidth, the transition node and storage element pair is created. The e-VLBI workflow management system simultaneously checks the measurement session progress and when it is completed, it updates information on which transition node should be paired with which storage element.

b) Limitations and work to be done

The main limitations for the Express Network Monitor are:
- the time needed to perform and analyse on-demand tests, and

- the impact on performance of the tested network. Each bandwidth utilization test floods the measured network with a huge number of packets to check the maximum transport speed.

The implemented session algorithm for the Express Network Monitor takes into account these two limitations. When possible, it performs parallel measurements. This reduces the time of the overall measurement session duration. The algorithm also reduces the number of active bandwidth utilization measurements in order to minimize the impact on the network performance.

Currently the Express Network Monitor is at the development stage. Full integration of the module and the Express workflow management system is planned for January 2008. Before it is released, the system will be tested on a virtual network to show the proper behaviour of the implemented algorithm.

V. SUMMARY

The paper describes a Grid-based approach to VLBI observations. It has been shown how the Mk4 hardware correlator can be replaced with a software version and how VLBI observations can be conducted using Grid resources. However, the e-VLBI system is in the development phase. Therefore, the concept of performing VLBI experiments in the Grid environment has not been verified in the field yet. However, we trust that this is the optimal solution for this kind of system as it was a result of careful analysis of problem domain as well as radio astronomy hardware specification. Unfortunately the nature and complexity of this problem may raise many questions and uncertainties that may have to be solved only by experimenting and creating working models and prototypes.

References

[1] A. Hanemann, J. Boote, E. Boyd, J. Durand, L. Kudarimoti, R. Łapacz, M. Swany, J. Zurawski and S. Trocha, *PerfSO-NAR: A Service Oriented Architecture for Multi–Domain Network Monitoring*. In: Proceedings of the Third International Conference on Service Oriented Computing, Springer Verlag, LNCS 3826, pp. 241–254, ACM Sigsoft and Sigweb, Amsterdam, The Netherlands, December, 2005

[2] DANTE homepage including information about GN2 project, available at: http://www.dante.net/

[3] Iperf homepage, available at: http://sourceforge.net/projects/iperf

[4] Bandwidth Test Controller homepage, available at: http://e2epi.internet2.edu/bwctl/

[5] perfSONAR Command Line Measurement Point documentation, available at: http://wiki.perfsonar.net/jra1-wiki/index.php/CLMP-Brief-Installation-Guide

[6] The Network Measurements Working Group (NMWG) of the Global Grid Forum, available at: http://nmwg.internet2.edu/

[7] National Radio Astronomy Observatory: http://www.nrao.edu/index.php/learn/radioastronomy/radiotelescopes

[8] JIVE website – http://jive.nl

[9] EVN website – http://evlbi.org

[10] M. Okon, D. Stoklosa, R. Oerlemans, H. J. van Langevelde, D. Kaliszan, M. Lawenda, T. Rajtar, N. Meyer and M. Stroinski, *Grid Integration of Future Arrays of Broadband Radio-Telescopes moving towards e-VLBI*. Grid Enabled Remote Instrumentation. Springer, 2008. p. 571.

[11] JGoodies focuses on Java look, UI design and usability http://jgoodies.com/

[12] EXPReS website -http://www.expres-eu.org/

ŁUKASZ DOLATA received the M.Sc. degree in Electrical Science from Poznań University of Technology in 2002. He is mainly involved in the network management and provisioning applications designing and developing. He was involved in creating the PIONIER Monitoring and Management System. Since 2004, he was involved in GEANT2 JRA3 activity and take place in implementing Automated Bandwidth Allocation across Heterogeneous Networks (AutoBAHN) system. He is also participating in the EXPReS project.

MARCIN OKOŃ graduated from the Poznań University of Technology and received the M.Sc. in Computer Science in 2002. His research interests include software engineering, distributed systems design and implementation, scheduling, and database modeling. Since December 2002, he has been participating in the Virtual Laboratory project.

DOMINIK STOKŁOSA works for Poznań Supercomputing and Networking Center on the Virtual Laboratory Project (http://vlab.psnc.pl). His research interests include the Software Engineering area, distributed environment computing, task scheduling and new Java-related technologies. He is also interested in design patterns and refactoring.

SZYMON TROCHA received the M.Sc. degree in Computer Science from Poznań University of Technology in 1998. He is the Head of the Management Unit in PSNC. He is mainly involved in the network management applications planning and implementing. He is also responsible for traffic analysis and measurement technology research and implementation. Since 2004, he has been leading the perfSONAR activity in PSNC. He is also participating in the EXPReS project.

DR. NORBERT MEYER is currently the Head of the Supercomputing Department in Poznań Supercomputing and Networking Center (http://www.man.poznan.pl). His research interests concern resource management in the GRID environment, GRID accounting, data management, technology of development graphical user interfaces and network security, mainly in the aspects of connecting independent, geographically distant Grid domains. He conceived the idea of connecting Polish supercomputing centres, vision of dedicated application servers and distributed storage infrastructure. He is the author and co-author of 60+ conference papers and articles in international journals, member of programme committees of international conferences related high performance computing and grid computing. Norbert Meyer was the leader of RINGrid EU project, currently leading the DORII project.

DR. MACIEJ STROIŃSKI is the Technical Director of the Poznań Supercomputing and Networking Center, Vicepresident of the PIONIER Consortium – Polish Optical Internet and member of the Supervisory Board. He is also lecturer in the Institute of Computing Science of the Poznań University of Technology. His research interests concern networking, quality of service, network protocols and management, grid related topics. He is author and co-author of over 150 papers in major professional journals and conference proceedings. Maciej Stroiński is the co-author of the Pionier Programme, accepted by the Polish State Committee for Scientific Research and announced as the official Polish programme in 2000. He is a member of Programme and Steering Committees of national and international conferences and editor of Computational Methods in Science and Technology journal. Maciej Stroiński participated in several networking and grid-related EU and national projects, co-operating with well known IT companies. He established Sun Center of Excellence and Microsoft Innovation Center in PSNC.

DAMIAN KALISZAN graduated from the Poznań University of Technology and received his M.Sc. in Computer Science (Computer Integrated Management and Production Systems) in 2001. He currently works at Poznań Supercomputing and Networking Center on the Virtual Laboratory Project (http://vlab.psnc.pl). His research interests include data mining, web and Java related technologies.

TOMASZ RAJTAR currently works at the Supercomputing Department in Poznań Supercomputing and Networking Center (PSNC) as a software analyst. His interests focus on high performance computing, object programming, visualisation, web technologies. Now he is directly involved in the Virtual Laboratory research project developed as a part of the project entitled High performance computations and visualisation for virtual laboratory purposes with the usage of SGI cluster.

MARCIN LAWENDA graduated from the Poznań University of Technology and received M.Sc. in Computer Science (Parallel and Distributed Computation area) in 2000. He defended the Ph.D. thesis at the same university in the Faculty of Computing Science and Management, Institute of Computing Science, in 2006. He currently works for Poznań Supercomputing and Networking Center on the project manager position on the Virtual Laboratory project. His research interests include parallel and distributed computing, scheduling and Grid technologies. He has been a member of the Polish Information Processing Society since 2000.