Remote Instrumentation: Building an Infrastructure Using the Grid

Thomas Prokosch[1, 2], Jens Volkert[1], Dieter Kranzlmüller[1]

[1] GUP, Joh. Kepler University Linz, Altenbergerstr. 69, A-4040 Linz, Austria/Europe
[2] prokosch@gup.uni-linz.ac.at

Abstract. The EU project "RINGrid" is a research project dealing with Remote Instrumentation in Next-generation Grids, i.e. instruments are seen as grid components and operated from remote sites. While this remote operation possibly entails trendemous benefits, several key components, such as scalable data storage or collaboration tools, are still missing for its realization. This paper outlines the process of remote instrumentation and highlights during this description those components, which are not yet fully understood or available.

I. Introduction

Using remote instruments is often difficult and requires a series of mandatory steps: First, the researcher has to ask for permission and a free time slot; then a journey to the remote site is required, which may take days in the case of an outlying astrophysical observatory; eventually the researcher has to expend considerable amounts of money for this journey. If the experiments fail or do not produce the desired output, the researcher needs to go home, analyze the data and correct the mistakes in the experimental setup. Then another journey is necessary. This tedious procedure could be optimized if the instruments can be used remotely over already existing network connections (for example the Internet).

Advantages of remote usage of instruments include not only better accessability of these instruments, but, as a consequence, the instruments will also experience better utilization. Decoupling the experiment from the local availability of an instrument also benefits developing countries — poorer nations suddenly get access to expensive instruments at a fraction of the initial building costs, which contributes to their education. All those advantages are good reasons to perform research on the remote instrumentation idea.

The EU project "RINGrid"[a] tries to integrate instruments as part of a grid infrastructure. This is a logical step considering that computational and storage resources provided by grids are data processing and data storage facilities for experiments. However, experiments seldomly start with data but the preceding step of data collection, which is done by instruments. Thus, adding an instrument as a grid component allows the researcher to plan the experiment without a rupture in the workflow.

II. Problems to be solved

The domain of remote instrumentation on the grid is relatively new and therefore many problems have to be solved on both the network as well as on the middleware level. The problems arise from the nature of experiments themselves: Most experiments are interactive and produce vast amounts of data which have to be stored and/or transmitted. Another problem results from the fact that the researcher is conducting the experiment remotely, not being present at the instrument's site: Collaboration between researchers working on the same project has to be achieved somehow, so collaboration tools have to be integrated into the researcher's workflow. These are the main problems, with which we want to deal one by one in the next chapters. Consequently, this paper is structured as follows:

- workflow management
- interactive experiment steering
- data transmission
- data storage
- collaboration

[a] The RINGrid project is funded by the FP6-2005-Infrastructures-7 programme of the European Commission under contract number 031891.

A. Workflow management

A workflow describes dependencies inherent within an experimental setup: For example, for being able to process data, data has to be recorded first. In order to automate processes (experiments in this case), the tools involved need to know where necessary data can be retrieved. In grids different solutions already exist for workflow management, however instruments need to be taken into account for representing the first step in a workflow chain.

An example in grid environments is the workflow language "BPEL", which is short for "Business Process Execution Language", and represents the prevailing standard. BPEL is extensible by attaching abstract definitions to BPEL documents. Since BPEL is the de-facto-standard, it is expected that this language will have support for remote instrumentation in the near future.

B. Interactive experiment steering

If data collection and the subsequent processing of the experimental data does not take too long, the researcher can utilize the rapid turnaround and the powerful machines (the grid as well as the office workstations). This allows to try several variations of parameters in order to get optimal results from the experimental setup without having to register for additional timeslots on the instruments. The described approach does not only save costs, but allows for faster development of the experimental series.

However, for an experiment to be controlled remotely, it is necessary to display the (preliminary) experimental data as well as some control elements on the researcher's host machine. These two screen elements have to be handled distinctly since they have different requirements: The experimental data can be either simple numbers (for example parameters for followup experiments, or coefficients), static images (as it is the case in radio astronomy) or high-resolution video streams. The data rate can be potentially high, therefore the demand on the underlying network heavy. Control elements, on the other side, can be encoded efficiently, the demand on the network speed is rather limited. However, it can be critical to keep the latency low.

C. Data transmission

There are several unrelated issues with data transmission which have to be solved. Fortunately, not all applications require the solution of all problems. Some applications need high connection bandwith, others reasonable round trip time or only small jitter. Some applications require that no packets are lost. Depending on the applications requirements, appropriate network protocols and underlying network hardware has to be chosen. For several application domains, we have determined the requirements for the network connection.

D. Data storage

The main problem with data storage is that some experiments produce huge amounts of data. This is not a problem for a single experiment, however when making dozens of such experiments per day, space requirements outgrow usual storage capabilities (i.e. harddisks). The solution will be a combination of two things: Firstly, reduce the data to a reasonable amount by means of data reduction, compression or preliminary analysis. Secondly, distribute the data among several grid nodes and take precaution that the data can be easily retrieved. The second part is accomplished by a data management service.

E. Collaboration

While in normal experiments scientists are at the instrument's site and can work together by regular human interaction, this interactive element is missing when doing experiments remotely, i.e. each scientist stays in the home office. This missing element has to be emulated somehow, however as there are no technical restrictions, this element can look quite differently. We have had a look at several solutions which promise good results.

We now want to describe the solutions to these five mentioned problems, one at a turn. We found the solutions by doing literature review and by looking at remote instrumentation projects, such as GridCC [1], CIMA [2], VLab [3] and so on.

III. Workflow management

The main reason to integrate workflow management into the remote instrumentation architecture is that scientists cannot manage data produced by the instruments locally on the machines in their offices. The data originates at the instrument, and then it has to be stored at least temporarily or even permanently on hosts near the instrument. From there it needs to be processed and eventually forwarded to other hosts. All these interdependencies need to be managed so that the scientists know where and in what form the data are. We have done an extensive review on existing workflow management systems. Interesting systems are:

- g-Eclipse [4]: This IDE for the grid supports workflows. However, since the workflow management capability is built into the environment and cannot be used independently, it will not be further considered.

- Yet Another Workflow Language (YAWL [5]): This language has been developed at Eindhoven University of Technology. It is also tied to an execution engine and a graphical editor. However, these tools are licenced under an open source licence (LGPL).

- XML Process Definition Language (XPDL [6]): XPDL defines an XML schema for specifying a workflow. The language is special as it does not only define interdependencies between the actors of a workflow, but also allows the user to create diagrams from the XML files. This is possible because within the XML files coordinates can be attributed to the actors.

- Business Process Execution Language (BPEL [7]): The name of this language is misleading: Not only business processes can be described with BPEL but also experimental setups. However, this language has been designed with long-running applications in mind, and these kind of applications are typically encountered in business environments. This language is prepared for workflows that span multiple organizational entities, thus it is the language with most supporters in industry and science.

As some experiments take days or even weeks to finish, it is expected that the RINGrid architecture will use the BPEL workflow language.

IV. Interactive experiment steering

For steering an experiment interactively, the researcher needs access to the experimental data. In case the data volume is too large to be transmitted, it makes sense to visualize (create static images or movies from the data on a host near the instrument) and send this stream instead over the grid to the researcher's host. This scenario can be seen in figure 1 [8, 9].

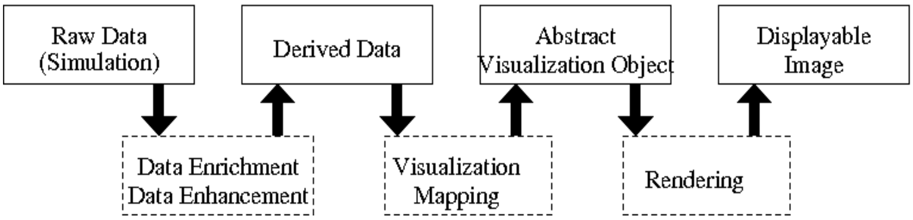

Figure 1. Conceptual visualization model.

There exist two middleware components which allow fast and secure access to remote video: glogin [10] and GVid [11].

A. glogin

The tool glogin [10] offers a bi-directional channel between grid hosts and the researcher's desktop machine. It is able to forward arbitrary data in a secure way, both from and to the grid hosts. The functionality is comparable to SSH, providing means for shell access, encrypted X11 and TCP port forwarding, VPNs and so on. While it is perfectly fine to use glogin for grid shell access only, we concentrate on interactive steering and therefore transmission of video content over glogin. This video content is managed by GVid.

B. GVid

GVid allows a visualization to be rendered at a random grid node while showing the visualization output on the researcher's local desktop machine. Because of this work distribution, complex visualization tasks are no longer a problem due to the many computation nodes available in a grid. GVid solves two problems:

1. Transmission of video data to the user's desktop.
2. Communication of interaction events back to the remote rendering processes running on the grid.

Therefore, when using GVid it is possible to have full interactive remote visualization.

V. Data transmission

When talking about data transmission, some characteristics can be identified which have an impact on the usability of the experimental data. These network characteristics are: bandwidth, network latency (describes the timeliness of the network), jitter (which eventually leads to lost samples, depending on the application), packet loss.

For a successful remote instrumentation setup, those network characteristics need to be matched with the requirements which come from the instruments themselves. Those instrument requirements are: need for local processing power, cost per element, number of data sources, operational mode (real-time or normal), data rate and data value [12].

Basically, there is a correlation between the application domain and the network requirements: Different applications require different data transmission capabilities. When for example looking at NMR spectroscopy, data rate is typically high, but real-time data transmission and analysis is normally not needed. Therefore it is mandatory to have a high-speed network, however the use of IPv6 for guaranteeing Quality of Service is not necessary. Conversely, when looking at measurement, control and automation experiments, real-time capabilities are often required, the data rate however is neglectable. Table 2 shows some applications and their requirements.

	measurement, control and automation	**large-scale physics and astronomy**	**sensor networks**	**NMR spectroscopy**
processing power	some ("smart")	high	moderate (RISC microprocessors)	high
operated manually/ automatically	manually	manually	automatically (ad-hoc multihop network)	manually
cost per element	small-medium	expensive	low-moderate	expensive
number of data sources	relatively high	singular	large	singular
operation mode	near real-time to real-time	real-time	normal (not critical)	normal (not critical)
data rate	low-moderate	high	low-moderate	moderate
data value	relatively low	very high	low	high

Table 2. Some applications and their requirements.

VI. Data storage

Since data reduction, compression and preliminary analysis strongly depend on the experiment itself, we concentrate on data management. Data management is typically comprised of the following tasks:

- Data movement between hosts,
- data replication and data access,
- providing data consistency,
- movement planning and bulk data movement prediction, as well as
- sensible replica placement.

Figure 3 [13] shows a possible grid data storage architecture. As it can be seen, the data management service is the central hub for serving a user's request. After receiving a file's metadata from the metadata catalog, it sends a request to the replication location services as well as to the data transmission service. A network monitor watches the current network load and hence controls the replication selection service. The exact duties of each component as well as problems resulting from that architecture will be discussed in the next sections.

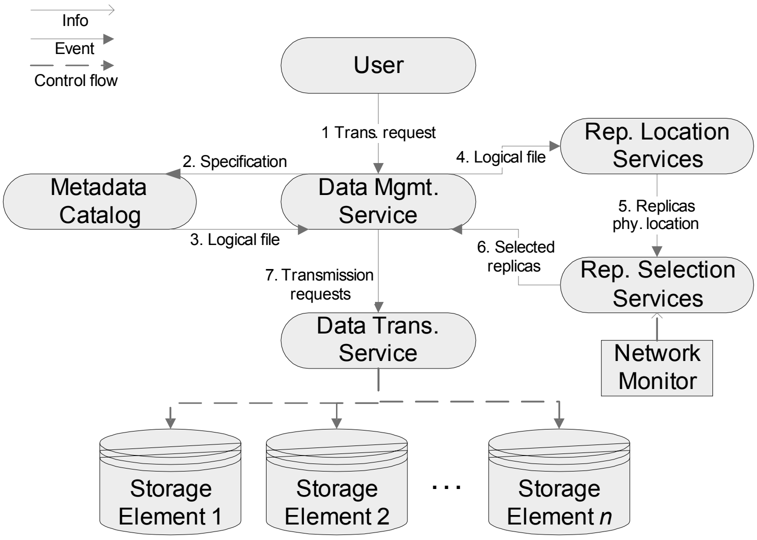

Figure 3. Data storage architecture.

A. Data movement

For low-level data movement, GridFTP is used instead of normal FTP or HTTP. GridFTP offers an advantage in wide-area networks where vast amounts of data have to be moved: The data is moved directly from the originating and the destination host, not bothering the host which initiated the data transfer. This way even a laptop or a PDA can work with grid resources and initiate data transfers without penalizing the overall data rate.

However, the movement initiating host needs to have knowledge about the current state of the transfer, a notion of the global state of data transfers between two hosts is not available. Having a global state would be advantageous in order to balance the network load more evently. Therefore, a service-based architecture is needed. The Globus Toolkit provides a stateless web service to users in form of the Reliable File Transfer (RFT). The gLite middleware implements File Transfer Services (FTS) which basically serve the same purpose.

B. Data replication and access

Data replication brings benefits in form of robustness, scalability and performance. However, there are penalties, too: Data will be harder to locate; data consistency is a hard problem which has not been solved completely; network traffic and security considerations need also to be looked at. The key to successful data replication is a good management of data metadata.

C. Data consistency

As soon as the data are read-write (in contrast to read-only) and there are several copies of the same data, one faces the same consistency problems as with any other cache. The possible solutions are as follows:

- Strict consistency: All data are kept in sync all the time. This requires locking the data before writing and releasing the lock thereafter. Of course, deadlocks are possible.
- Lazy-copy: Replicas are only updated when someone tries to access an outdated copy. While saving network bandwidth, lazy-copy inevitably leads to latencies.
- Aggressive copy: Replicas are updated as soon as modifications are made on one of the copies. While reducing the delay time and not being able to suffer from deadlocks, this method has the same disadvantage as lazy-copy, i.e. the consistency cannot be guaranteed.

It is probably best to support different levels of consistency within the middleware, depending on the need of the grid application.

D. Movement planning

Data movement is needed to guarantee fast data access even if the specific grid nodes, which access the data, changes over the lifetime of the grid application. Such changes in the access pattern need to be detected and the data needs to be moved to hosts which are nearer to the accessing nodes. Data movement prediction utilizes statistical methods which can be:

- Mean-based model: Some portion of the access history is used to estimate future behaviour.
- Median-based model: This model is useful if only few grid nodes randomly access the data, but are not generating as much traffic as other, more nearby nodes.
- Auto-regression model: A weighted average of past values is calculated, therefore giving a more precise prediction. However, the computational cost is higher.

E. Replica placement

Instead of just having one copy of the data, and using data movement to optimize communication time, one could create replicas and place them strategically so that communication time is minimized again. However, the data consistency problem is again introduced. Additionally, finding an optimal place for the replicas is not trivial: Wolfson and Milo showed that the replica placement problem is NP-complete for arbitrary graphs (as the grid is one) when read and update costs are equal [14].

VII. Collaboration

Collaboration is closely coupled with the surrounding environment. The GridCC [1] project has infrastructure which includes the so-called "Multipurpose Collaboration Environment" (MCE). The Storage Resource Broker (SRB, [15]), developed at the San Diego Supercomputer Center (SDSC), fosters collaboration by presenting the user a single file hierarchy for storing data. This file hierarchy is distributed among many storage systems but the user does not perceive the boundaries. VLab [3] tries to foster collaboration between scientists by integrating instant messaging software such as Skype and Gadu-Gadu.

As one can see, the collaboration facilities are as different as the projects. This is why we did not come to a conclusion which system is best suited for remote instrumentation experiments.

VIII. Conclusions

This paper provides an overview of remote instrumentation on the grid. While many problems are already addressed, there are still many gaps which need to be filled. This paper tries to highlight what components can already be used and where research still has to be done.

Unfortunately, the remote instrumentation grid infrastructure is currently a patchwork where key middleware components are still missing, and other components which are less important are fully understood and implemented. Upcoming research needs to address the missing key components, so that the scientific community can reap the rewards of longstanding research as soon as possible.

References

[1] GridCC web site, http://www.gridcc.org/

[2] D. F. McMullen, K. Chiu, "CIMA: Scientific Instruments as First Class Members of the Grid", *Molecular and Materials Structure Network Workshop*, Marysville, Australia, 2005.

[3] M. Lawenda, N. Meyer, T. Rajtar, et al, "General Conception of the Virtual Laboratory", *International Conference on Computational Science 2004, LNCS 3038*, pp. 1013-1016, 2004.

[4] g-Eclipse web site, http://www.g-eclipse.org/

[5] W. M. P. van der Aalst, L. Aldred, M. Dumas, A. H. M. ter Hofstede, "Design and Implementation of the YAWL system", *Proceedings of the 16th International Conference on Advanced Information Systems Engineering*, Springer Verlag, 2004.

[6] Jiang Ping, Q. Mair, J. Newman, "Using UML to design distributed collaborative workflows: from UML to XPDL", *12th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003.

[7] OASIS, *WS-BPEL 2.0 Specification*, http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf, 2007.

[8] R. B. Haber, D. A. McNabb, "Visualization Idioms: A Conceptual Model for Scientific Visualization Systems", in: G. M. Nielson, B. Shriver, L. J. Rosenblum (eds.), *Visualization in Scientific Computing*, IEEE Computer Society Press, pp. 74–93, 1990.

[9] Paul Heinzlreiter, Dieter Kranzlmüller, "Visualization Services on the Grid: The Grid Visualization Kernel", *Parallel Processing Letters* 13(2), pp. 135-148, 2003.

[10] Herbert Rosmanith, Dieter Kranzlmüller, "glogin: A Multifunctional, Interactive Tunnel into the Grid", *GRID 2004*, pp. 266-272, 2004.

[11] T. Köckerbauer, M. Polak, T. Stütz, A. Uhl, "GVid: video coding and encryption for advanced Grid visualization", in: J. Volkert, T. Fahringer, D. Kranzlmüller, W. Schreiner (eds.), *Proceedings of the 1st Austrian Grid Symposium*, volume 210, pp. 204-218, 2006.

[12] Bramley et al, "Instruments and Sensors as Network Services: Making Instruments First Class Members of the Grid", Technical Report TR588, Indiana University Department of Computer Science, December 2003.

[13] Thomas Prokosch (ed.), *Status of Grid Middleware and Corresponding Emerging Standards for Potential Usage in Sharing Scientific Instruments via (International) Networks*, RINGrid Deliverable D3.2, EU project 031891, 2007.

[14] Ouri Wolfson, Amir Milo, "The Multicast Policy and Its Relationship to Replicated Data Placement." *ACM Trans. Database Syst.* 16(1), pp. 181-205, 1991.

[15] The San Diego Supercomputer Center Storage Resource Broker (SRB), http://www.sdsc.edu/srb/