EGI-InSPIRE

INTEGRATION OF THE VISIVO SERVER WITH THE PRODUCTION INFRASTRUCTURE

EU MILESTONE: MS608

Document identifier:	EGI-SA3-MS608-A&A-UB.doc
Date:	**09/06/2011**
Activity:	**SA3**
Lead Partner:	**INAF**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	https://documents.egi.eu/document/519

Abstract

The Visualisation Interface to the Virtual Observatory (VisIVO) is a suite of software tools aimed at creating 3D customized views of many type of wide used data sets. Its peculiar characteristic is that there is no limit for what concerns the size of input tables containing data to be processed, thus they are able to support very large scale datasets (tens of Terabytes and more).

VisIVO Server consists of three core components: VisIVO Importer, VisIVO Filter and VisIVO Viewer respectively. To create customized views of 3D renderings from data tables, a two-stage process is needed.

First, VisIVO Importer is used to convert user datasets into VisIVO Binary Tables (VBTs). Then, VisIVO Viewer is invoked to display customized views of 3D renderings. VisIVO Filters are collections of data processing modules able to explore datasets enhancing and highlighting their hidden properties.

The software is designed to obtain images and movies from user files. Several data formats are currently supported. The process of movie creation could last several hours.

This document describes the activity related to the porting of VisIVO (Visualization Interface to the Virtual Observatory) to gLite Grid middleware with the release of the new version 1.2 made available to the scientific community. This activity has been carried out through two subsequent steps. The first step for this activity consisted in enabling the usage of VisIVO Server directly within a user application during the production phase. To this end, a software layer (API) has been developed for this; the developed API makes use of the internal arrays of VisIVO, so the production of intermediate files is no longer necessary. To achieve this goal, a library for VisIVO was designed and implemented. Through the second step the usage of the grid catalogue within VisIVO Server has been enabled; the user application is enabled to directly read/write user data from/to the catalogue (or from internal user code arrays) to produce a sequence of images (a movie) allowing the user to check the data production while the application is running or at the end of its execution. The new release 1.2 of VisIVO is maintained in sourceforge; it can also be directly downloaded from the VisIVO web site (http://visivo.oact.inaf.it). The release includes the full documentation and demonstration material.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Dr. Ugo Becciani -	INAF/SA3	10-05-2011
Reviewed by	**Moderator:** Nuno Ferreira **Reviewers:** Nicholas A. Walton	EGI.eu external	09-06-2011
Approved by	**AMB & PMB**		09-06-2011

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1	10-05-2011	First draft	Ugo Becciani
2	14-05-2011	Second draft	Ugo Becciani
3	09-06-2011	Final version	Ugo Becciani

IV. APPLICATION AREA

This document is a formal deliverable for the European Commission, applicable to all members of the EGI-InSPIRE project, beneficiaries and Joint Research Unit members, as well as its collaborating projects.

V. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGI-InSPIRE "Document Management Procedure" will be followed: https://wiki.egi.eu/wiki/Procedures

VI. TERMINOLOGY

A complete project glossary is provided at the following page: http://www.egi.eu/about/glossary/.

VII. PROJECT SUMMARY

To support science and innovation, a lasting operational model for e-Science is needed − both for coordinating the infrastructure and for delivering integrated services that cross national borders.

The EGI-InSPIRE project will support the transition from a project-based system to a sustainable pan-European e-Infrastructure, by supporting ‘grids’ of high-performance computing (HPC) and high-throughput computing (HTC) resources. EGI-InSPIRE will also be ideally placed to integrate new Distributed Computing Infrastructures (DCIs) such as clouds, supercomputing networks and desktop grids, to benefit user communities within the European Research Area.

EGI-InSPIRE will collect user requirements and provide support for the current and potential new user communities, for example within the ESFRI projects. Additional support will also be given to the current heavy users of the infrastructure, such as high energy physics, computational chemistry and life sciences, as they move their critical services and tools from a centralised support model to one driven by their own individual communities.

The objectives of the project are:

1. The continued operation and expansion of today’s production infrastructure by transitioning to a governance model and operational infrastructure that can be increasingly sustained outside of specific project funding.
2. The continued support of researchers within Europe and their international collaborators that are using the current production infrastructure.
3. The support for current heavy users of the infrastructure in earth science, astronomy and astrophysics, fusion, computational chemistry and materials science technology, life sciences and high energy physics as they move to sustainable support models for their own communities.
4. Interfaces that expand access to new user communities including new potential heavy users of the infrastructure from the ESFRI projects.
5. Mechanisms to integrate existing infrastructure providers in Europe and around the world into the production infrastructure, so as to provide transparent access to all authorised users.
6. Establish processes and procedures to allow the integration of new DCI technologies (e.g. clouds, volunteer desktop grids) and heterogeneous resources (e.g. HTC and HPC) into a seamless production infrastructure as they mature and demonstrate value to the EGI community.

The EGI community is a federation of independent national and community resource providers, whose resources support specific research communities and international collaborators both within Europe and worldwide. EGI.eu, coordinator of EGI-InSPIRE, brings together partner institutions established within the community to provide a set of essential human and technical services that enable secure integrated access to distributed resources on behalf of the community.

The production infrastructure supports Virtual Research Communities (VRCs) − structured international user communities − that are grouped into specific research domains. VRCs are formally represented within EGI at both a technical and strategic level.

VIII. EXECUTIVE SUMMARY

VisIVO is a suite of software tools aimed at creating 3D customized views of many type of wide used data sets. Its peculiar characteristic is that there is no limit for what concerns the size of input tables containing data to be processed, thus they are able to support very large scale datasets (tens of Terabytes and more).

The linear scaling observed in our tests suggests that handling large datasets is effectively restricted only by the underlying file system limitations.

VisIVO Server consists of three core components: VisIVO Importer, VisIVO Filter and VisIVO Viewer respectively. To create customized views of 3D renderings from astrophysical data tables, a two-stage process is needed.

First, VisIVO Importer is used to convert user datasets into VisIVO Binary Tables (VBTs). Then, VisIVO Viewer is invoked to display customized views of 3D renderings. VisIVO Filters are collections of data processing modules able to explore datasets enhancing and highlighting their hidden properties.

The software is designed to create images and movies from user files. Several data format are currently supported. The process of movie creation could last several hours.

The first fundamental goal of porting VisIVO to the Grid is to have movies and images directly stored on the Grid, even if intermediate files are not produced, and to reduce the overall time for movie production.

For this reason two main steps are requested to have the first version of VisIVO portable on gLite Grid: the first step is to allow the usage of VisIVO directly within a code during the production phase. A VisIVO library was designed from November to December 2010 and the development phase took place in the first four months of 2011. VisIVO library represent an important step in order to export the usage of the 3D visualization rendering and movie creation tools to other user communities. When this phase will be successfully completed, the MPI-enabled version of VisIVO will be tested on the infrastructure, thereafter it will be deployed.

Through the second step the usage of the grid catalogue within VisIVO Server has been enabled; the user application is enabled to directly read/write user data from/to the catalogue (or from internal user code arrays) to produce a sequence of images (a movie) allowing the user to check the data production while the application is running or at the end of its execution.

This document represents the MS608 of EGI-InSPIRE and is related to the porting of VisIVO server in Grid and to the issue of the release 1.2 of the server maintained in sourceforge. This document therefore reports about the work carried out during the first year of EGI-InSPIRE to release the first Grid-compliant version of VisIVO server, corresponding to the steps reported above.

The work carried out so far does not run out the activity planned for VisIVO server and Grid. Other planned activities concern the development of prototypes for Importers and Filters, the production of an MPI-compliant version of VisIVO and a version of VisIVO for CUDA generated through GPUs on the worker nodes. At the end of this activity an enhanced full-Grid-compatible version of VisIVO will be produced.

VisIVO is, up to now, deployed in the Cometa Consortium Grid (gLite v 3.1). This infrastructure is distributed in seven sites located in the three main towns of Sicily: Catania, Messina and Palermo. All sites have the same hardware and software configuration allowing high interoperability and providing a homogeneous environment. The computing infrastructure is based on IBM Blade Centre each containing up to 14 IBM LS21 blades interconnected with the low latency Infiniband-4X network; such infrastructure mainly provides High Performance Computing (HPC) functionalities on the Grid. There are about 2000 CPU cores and more than 200 TB of disk storage space currently available on the Sicilian e-Infrastructure.

TABLE OF CONTENTS

1 INTRODUCTION

An essential part of modern scientific research is the need to employ computer graphics and scientific visualization tools to appropriately display datasets, so as to allow scientists to perform efficient visual discovery.

VisIVO provides an integrated suite of tools and services that can be used in many scientific fields (i.e. chemistry, nuclear physics, biomedical etc.). It allows users to visualize meaningfully highly-complex, large-scale datasets and to create movies of these visualizations based on distributed infrastructures.

VisIVO is currently maintained by INAF (Istituto Nazionale di Astrofisica) in collaboration with the University of Portsmouth and the CINECA. The first implementation of VisIVO was initiated within the EU funded project VO-TECH (http://www.eurovotech.org/), thereafter it was supported by the Cometa Consortium in Catania, Italy, within the PI2S2 project (http://www.consorzio-cometa.it/).

VisIVO consists of several parts: a) VisIVO Desktop - a stand-alone application for interactive visualization on standard PCs; b) VisIVO Server - a platform for high performance visualization; c) VisIVO Web - a custom designed web portal supporting services based on the VisIVO Server functionality.

VisIVO Web is currently hosted at the University of Portsmouth, UK (visivo.port.ac.uk), at the INAF Astrophysical Observatory of Catania, Italy (visivo.oact.inaf.it) and at INAF Astronomical Observatory of Trieste (http://palantir7.oats.inaf.it/visivoweb/). These web sites offer data management functionalities for registered users; datasets can be uploaded for temporary storage and processing for a period of up to two months. The sites can also be anonymously accessed; in this case datasets can be uploaded and stored for a maximum of four days; to maximize the amount of available resources, a limited dimensionality is only supported.

Once datasets are uploaded, users can operate with tree-like structures (for easy data navigation) containing pointers to **files**, **tables**, **volumes** as well as **visuals**.

Files point to single, or possibly several (for distributed datasets), data tables;

Tables are highly-efficient data representations internal to VisIVO Server; they are typically produced from importing datasets uploaded by users using VisIVO Importer (see below);

Volumes are internal VisIVO Server data representations; they are produced either by directly importing user datasets or by performing operations on already existing tables;

Visuals are collections of highly-customized, user-produced views of 3D renderings of volumes.

VisIVO Server consists of three core components: **VisIVO Importer**, **VisIVO Filter** and **VisIVO Viewer** respectively.

To create customized views of 3D renderings from data tables, a two-stage process is employed. First, VisIVO Importer is utilized to convert user datasets into **VisIVO Binary Tables** (VBTs). Then, VisIVO Viewer is invoked to display customized views of 3D renderings. As an example, consider displaying views from only three columns of an astrophysical data table supplied in ascii form, say col_1, col_2 and col_3, by using the commands

```
VisIVOImporter --fformat ascii UserDataSet.txt
```

```
VisIVOViewer -x col_1 -y col_2 -z col_3 --scale --glyphs pixel
VBT.bin
```

VisIVO Server is distributed with GPL V.2 License for NON COMMERCIAL use. VisIVO Server is hosted by source forge https://sourceforge.net/projects/visivoserver/; it is possible to download it via SVN: svn co https://visivoserver.svn.sourceforge.net/svnroot/visivoserver visivoserver

2 RELEASE DETAILS

In the following we describe the three components of the current version of VisIVO Server (v. 1.2) and the new features we developed for the gLite grid porting. We will start with an overview on the three different components of VisIVO Server (***VisIVO Importer, Filter and Viewer***) and in particular the new library we developed for this milestone.

The new features of this version (in comparison with the previous version) are mainly based due to the VisIVO Library now available and the gLite grid catalogue that can be directly addressed by VisIVO now, as described in the par. 2.2 and 2.3

2.1 VisIVO main components

VisIVO is based on three main components: Importer, Filter and Viewer. VisIVO Importer converts user-supplied datasets into VBTs without imposing any limit on size or dimensionality. VBTs are employed by VisIVO Filter modules for data processing and by VisIVO Viewer for the final display. The general syntax needs to specify the input data format and the user filename.

The filename can refer to a local or remote file to be converted into the VBT. If a remote filename is given (starting with http://, sftp:// or ftp://) the remote file is downloaded automatically using the curl library. If a username and password are specified, however, they are used for remote accesses, if data are stored in a private account. The current version of VisIVO Importer supports conversion from several popular formats as follows: ASCII and CSV tables, VOTables FITS Tables, Gadget dataset (Gadget is a freely-available code for cosmological N-body/SPH simulations), Raw Binary (a binary memory dump). The operation of VisIVO Importer is highly optimised requiring in most cases a short period of time (a number of seconds) even for large-scale datasets. For example, no more than 30 secs are necessary for the import of 30 million binary elements: a N-body simulation output, having positions and velocity field, stored in the catalogue (local), was used. VisIVO was running on a Grid node having AMD Opteron 2218 rev. F dual-core processors with a clock rate of 2,6 GHz and equipped with Infiniband 4X.

VisIVO Filter is a collection of data processing modules to modify a VBT or to create a new VBT from other existing VBTs. The filters support a range of operations such as scalar distribution, mathematical operations, selections of regions, decimation, randomization and so on. The selection and randomization operations are of particular importance as they are typically employed for constructing reduced VBTs so that they can be used directly by VisIVO Viewer.

The general syntax is:

```
VisIVOFilters --op filterOpCode <options> [--file] InputFile
```

Where *filterOpCode* selects a specific operation to be processed (e.g. randomization), *options* is a set of flags related to the specific selected operation and *InputFile* is an input VBT file or a list of VBT files.

VisIVO Viewer is based on the Visualization ToolKit library for multidimensional visualization and on Splotch, a ray tracing rendering tool developed by the Max Plank Institute. It creates 3D images of datasets; both data points and volumes can be represented. VisIVO Server can manage multidimensional datasets and the visualization process can be given on a large number of elements. On the other hand, a visualization process with many million of elements (ten or more) does not typically add significant visual information than a few million of elements. Usually (with 2 GB RAM) VisIVO Viewer can display up to 16 million of data elements easily, but the visual information got by the user in this case is comparable to the amount of information given by two or four million of displayed elements. The visualization process for large-scale datasets requires some actions.

VisIVO Filters run the randomization operation. This operation reduces the original dataset to fit it into the available memory, and produces a new VBT used for visualization purposes. VisIVO Viewer then runs using the VBT produced through the aforementioned process.

VisIVO Viewer can render points, volumes and iso-surfaces within a bounding box used for the representation of the coordinate system employed. Moreover, support is supplied for customized look up tables enabling visualizations that make use of a variety of glyphs, such as cubes, spheres or cones. The standard output of VisIVO Viewer consists of images corresponding to fixed camera positions and zoom factors and another image corresponding to user-defined line command options. A camera path can also be provided: in this case VisIVO Viewer will generate a sequence of images that can be mounted to obtain a movie.

The VisIVO tool can be used to explore large dataset as the results of queries to astrophysical catalogues or data coming from cosmological simulations. The software however is general-purpose; it manages and explores data in tabular format, so that it can be adopted in many other fields where large datasets must be explored or, using the VisIVO Library, during the execution of a user application.

A detailed description of the VisIVO components can be found in the VisIVO User Manuals distributed with the code.

2.2 VisIVO API

The VisIVO Library was developed to port VisIVO on gLite middleware. It allows a job running on a grid node to produce a set of images or movies directly using VisIVO with its internal data arrays without the need to produce intermediate files. This is particularly important when running on the grid, where the user wants to have a quick look of the results during the data production phase. The images in this way can be produced directly in the grid catalogue, while the user code is running in a system that cannot be directly accessed by the user (a worker node).

The purpose of the VisIVO Library is a direct use of the VisIVO Server features within the user code. There are four library sections:

1. VisIVO Importer: contains all the environment variables to import a user file into a VBT
2. VisIVO Filter: contains all the environment variables to use the VisIVO Filters on a VBT
3. VisIVO Viewer: contains all the environment variables to create an image or a movie from a VBT.
4. General Utilities.

Images and movies can be produced from an existing user file or directly from internal arrays.

VisIVO Library makes use of environments each of them represented through a variable. More than one environment may be set for each library section.

All function names of a library section start with a prefix VI_ ,VF_ or VV_ respectively for Importer, Filter and Viewer sections.

Typically to set the working environment the user must declare the variable (e.g. VisIVOViewer env3;) and then initialize it (VV_Init(&env3)); finally, the attributes of the environment have to be set through primitives such as VV_SetAtt(VisIVOViewer *env, int code, char *value). The variable contains all settings for the operation. For instance, the env3 variable will contain the camera position or a movie camera path, the palette, the filename or the arrays pointers used to create the image. The image creation will be done by the function VV_View(&env3).

The specific actions are executed by the following functions:

- VI_Importer(&env1) where env1 is a variable of type VisIVOImporter;
- VF_Filter(&env2) where env2 is a variable of type VisIVOFilter;
- VV_View(&env3) where env3 is a variable of type VisIVOViewer.

There are also some general utilities allowing direct actions on data.
The above primitives are blocking functions; they return only when finished. The operation performed by VI_Import, VF_Filter, VV_View however could last several minutes or hours. To avoid the calling program to wait for their completion, we have implemented asynchronous calls; new processes or new threads are forked in this case.

The user can set the environment variables to use the multi-thread or the multi-process features. By default the multi-process features are adopted. The user application may call several instances for VA_Importer, VA_Filter, VA_View and several threads or processes will be started simultaneously. In case of execution with multi-thread only, the first call is started as a running thread whereas subsequent threads will be queued: only one new single thread will run on the system, the others will wait for completion. This is due to the fact that VTK is not fully thread safe. The multi-process, however, can be started simultaneously on the server.

More details can be found on the VisIVO API user manual.

2.3 VisIVO for gLite middleware

VisIVO may be compiled on a gLite system 3.1. In this case the *GLITE* option must be given when compiling the tool.

We used lcg_util APIs, a high-level gLite data management APIs. They are able to provide a single interface to deal with Storage Elements and LFC. They actually provide the same functionality available through the lcg_util command line tools (`lcg-cr`, `lcg-cp`, `lcg-rep`, `lcg-del`, etc). We used these functions to create a local copy of the table in the local WN.

The main purpose is to read a VBT from catalogue or to use the internal arrays and/or to create the images and movies in the catalogue.

When running on gLite:

1) the option `--VO` has to be used to specify the VO name;
2) the input VBT file, to be imported, can be a logical filename (lfn); in this case it must start with `lfn://`;
3) the output VBT may be local (`--out` option) or the output `lfn` can be given. The option `--out` must be followed by the name of the local file where temporary images will be created;
4) `--lfnout` is the output logical filename. In this case an image will be saved in the grid catalogue and deleted from the local system.
5) the option `--se` is used to specify the Storage Element (SE)

The following options can be applied:

`--VO` `[value] (optional)` is used to set the virtual organization (VO) when running on gLite grid. It is mandatory when running on gLite using the grid catalogue.

`--se` `[value] (optional)` is used to set the Storage Element (SE) when running on gLite grid. Default value is DPM_HOST (site default storage element).

`--lfnout` `[value] (optional)` is used to set the logical filename (lfn) of the output file when running on gLite grid.

2.4 Performances

In this paragraph we report the performances got from VisIVO using the Cometa Consortium gLite grid. We created movies using the VisIVO API from a user application. The user application reads a file from the grid catalogue with 70 Million of data points (x,y,z float coordinates) created by a cosmological N-body simulation in a 70 Mpc computational box. The number of movies created, using this file, range from 1 to 8. Each movie has 110 frames. In doing that we used the multi-process features of the VisIVO API.

Using a node with 4 cores (AMD Opteron 2218 rev. F dual-core processors with a clock rate of 2,6 GHz) the CPU time requested to produce one movie was 2,000 secs; this time remains constant to produce up to 4 simultaneous movies (we used four cores). The code takes 4,000 secs to produce up to 8 movies as we expected.

In case of multithread there are no gains for movie generation: each movie production takes 2,000 secs, the global time for 8 movies is 16,000 sec. The movies production is queued (VTK is not fully thread safe) but the user program can continue to perform other works.

The results we obtained are coherent with the time requested to produce a single frame: this is about 20 secs (on the above platform) for an user table with three float fields (columns) having in total 800 MB file size. The time scales linearly with the data dimension.

Note: the global performances, using gLite, may depend on the grid latency time, network latency and so on, but does not depend on VisIVO Server that uses a local copy of data.

2.5 VisIVO support and documentation

VisIVO can be downloaded from source forge

```
"svn co
https://visivoserver.svn.sourceforge.net/svnroot/visivoserver/branch
es/1.2 visivoserver"
```

or using the VisIVO portal (http://visivo.oact.inaf.it).

The distribution contains:

- ***VisIVO.HowToBuild***. This document illustrates the procedure to install VisIVO on a generic server.
- ***VisIVOServerGlite.HowToBuild***. This document reports about the procedure to install VisIVO on a gLite worker node.
- The directory ***test***. This directory contains a template applications written in C to test and to use the VisIVO Library
- The directory ***testGLite***. This directory contains the following jdls:
  - VisIVO.jdl to test VisIVO Importer, VisIVO Filters and VisIVO Viewer.
  - VisIVOApi.jdl to test VisIVO Library.

The distribution contains the following user manuals:

- VisIVOApi1.2.pdf (Library User Manual)
- VisIVOFilters1.2.pdf
- VisIVOImporter1.2.pdf
- VisIVOUtils1.2.pdf
- VisIVOViewer1.2.pdf

The VisIVO support can be obtained by sending an e-mail to **visivo-support@oact.inaf.it**. Experts will answer for the reported bugs or installation help requests.

On the VisIVO Web site (http://visivoweb.oact.inaf.it/) it is possible to find and download the material used during the on-line tutorial of June 2009. This tutorial, although related to the version 1.0 is still valid and can be used to obtain basic information on the usage of VisIVO. This material will be updated in the near future. This material is also published on ***YouTube*** searching for VisIVO Server and VisIVO Web tutorial.

3 CONCLUSION

Our future work will aim at fully porting VisIVO Server and VisIVO Web to grid environments: in particular, in the next months we will produce the MPI and CUDA enabled-version of VisIVO (optional features). The exploitation of grid technologies will enhance performance significantly by supplying the required computational capability. A very exciting possibility is the deployment of this technology for visual discovery in scientific disciplines apart from astrophysics. Furthermore, a pilot application built upon VisIVO Smartphone is underway for public engagement activities in astrophysics. In our ultimate vision, users (researchers, citizen scientists or even visitors of science centres) shall be able to reserve resources for data analysis and visual discovery as required (possibly involving several production grids) and to retrieve results on smartphones on-the-go irrespective of internet access or geographical location.

4 REFERENCES

R 1	U. Becciani et al. The Publications of the Astronomical Society of the Pacific Vol. 122, pp. 119-130, 2010
R 2	*U. Becciani et al.* ASP, Conference Series. Vol. 411, pp. 555, 2009
R 3	G. Caniglia et al. Second International Conference in Visualisation VIZ-09, IEEE transaction 2009 pp.10-15 ISBN: 978-0-7695-3734-4
R 4	http://visivo.oact.inaf.it, http://visivoweb.oact.inaf.it
R 5	http://www.vtk.org/
R 6	http://glite.cern.ch/
R 7	http://www.egi.eu/