VisIVO–Integrated Tools and Services for Large-Scale Astrophysical Visualization

U. BECCIANI,1 A. COSTA,1 V. ANTONUCCIO-DELOGU,1 G. CANIGLIA,1 M. COMPARATO,2 C. GHELLE,3 Z. JIN,4 M. KROKOS,4 AND P. MASSIMINO1

Received 2009 June 23; accepted 2009 November 30; published 2010 December 18

ABSTRACT. VisIVO is an integrated suite of tools and services specifically designed for the Virtual Observatory. This suite constitutes a software framework for effective visual discovery in currently available (and next-generation) very large-scale astrophysical data sets. VisIVO consists of VisIVO Desktop, a stand alone application for interactive visualization on standard PCs; VisIVO Server, a grid-enabled platform for high performance visualization; and VisIVO Web, a custom designed web portal supporting services based on the VisIVO Server functionality. The main characteristic of VisIVO is support for high-performance, multidimensional visualization of very large-scale astrophysical data sets. Users can obtain meaningful visualizations rapidly while preserving full and intuitive control of the relevant visualization parameters. This paper focuses on newly developed integrated tools in VisIVO Server allowing intuitive visual discovery with three-dimensional (3D) views being created from data tables. VisIVO Server can be installed easily on any web server with a database repository. We discuss briefly aspects of our implementation of VisIVO Server on a computational grid and also outline the functionality of the services offered by VisIVO Web. Finally we conclude with a summary of our work and pointers to future developments.

Online material: color figures

1. INTRODUCTION

An essential part of astrophysical research is the use of computer graphics and scientific visualization tools for displaying appropriately multidimensional data plots and images, both from real-world observations and from modern, highly complex numerical simulations. The latest generation of graphics and visualization software tools offers the astrophysical community robust instruments for data analysis and advanced data exploration by using:

1. High Performance Computing and Multithreading to exploit multicore CPUs and emerging powerful graphics boards, thus allowing real-time interaction with large-scale data sets.
2. Interoperability so as to allow different applications, each specialized for different purposes, to operate on shared data sets.
3. Collaborative Workflows so that several users can work simultaneously—perhaps at different geographical locations—on identical data sets exchanging information and visualization experiences.

Astrophysicists are currently witnessing an unprecedented growth in the quality and quantity of data sets coming from real-world observations and numerical simulations. For example, the increasing availability of high-performance and grid computing facilities (Becciani et al. 2009; Becciani 2007) has given the possibility to perform numerical simulations of several dimensions. Also, current, such as the Sloan Digital Sky Survey (SDSS, 2008; York et al. 2000), and next-generation sky surveys, such as the Low Frequency Array (LOFAR, 2009), the Square Kilometre Array (SKA) (Schilizzi et al. 2008), the Large Synoptic Survey Telescope System (LSST) (Ivezic et al. 2008), and the Dark Energy Survey (DES) (Mohr et al. 2008) are planned to collect large amounts of raw data eventually resulting in several hundreds of terabytes of data in tabular form. These developments place the emphasis on the problems associated with data post-processing. For storage, very large-scale distributed databases will be required and scientists will need to interact with them effectively using very high performance computer graphics and scientific visualisation tools, as simply downloading will be prohibitively costly. The emerging need is not only to provide data analysis and exploration tools that can run on standard PCs satisfactorily, but also to offer appropriate interoperable tools that can run on servers and standard PCs simultaneously.

This paper describes VisIVO Server,5 a new platform for astrophysical visualization of large-scale data sets that can be

5 VisIVO Server Web Sites: http://visivo.port.ac.uk/visivoweb/, http://visivoserver.oact.inaf.it, http://palantir7.oats.inaf.it/visivoweb/.
2. BACKGROUND

Traditionally the common practice among astrophysicists is to employ a plethora of individually created, autonomous applications for data analysis and exploration. However this scenario is not applicable to modern large-scale data sets and over the last few years a generation of software frameworks has emerged, e.g., Aladin (Bonnarel et al. 2000), which is an interactive software sky atlas allowing the user to visualize digitized astronomical images, superimpose entries from astronomical catalogues or databases, and interactively access related data and information from archives, and TopCat (2009), an interactive graphical viewer and editor for tabular data. Other examples of this new generation of software are 3D Slicer, which is capable of displaying volumes and user-defined cut throughs, SPLASH (Price 2007) for visualization of numerical simulations, and VisIt, which is a free parallel visualization and graphical analysis tool—such as isosurfacing and volume rendering algorithms—or even creating and visualization of vectors by combining a user-defined selection of scalar values (Fig. 1).

Section 3 discusses in detail the functionality of VisIVO Server, which constitutes the latest development in the VisIVO project for exploration of large-scale data sets. We report on performance tests of some VisIVO Server modules as an indication for users interested in installing and running VisIVO Filters on a local or remote server. All our performance tests were executed on an AMD Opteron 2218 rev. F dual-core processor with a clock rate of 2.6 GHz and 2 GB of DDR2 RAM.

3. VisIVO SERVER

VisIVO Server is a suite of software tools for creating customized views of 3D renderings from astrophysical data tables. These tools are founded on the VisIVO Desktop functionality and support the most popular Linux based platforms. Their deﬁning characteristic is that no ﬁxed limits are prescribed regarding the dimensionality of data tables input for processing, thus supporting very large-scale data sets. 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 employed. First, VisIVO Importer is utilized to convert user data sets into VisIVO Binary Tables (referred to as VBTs in the remainder of this article). 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; this is achieved by executing the following VisIVO Server commands:

```
VSIVOImporter -fformat ascii -out VBT.bin Userdataset.txt
VSIVOViewer -x col_1 -y col_2 -z col_3 VBT.bin
```

where Userdataset.txt is the user input ascii data table and VBT.bin is the binary VBT corresponding to it, created by the VisIVOImporter command. VisIVO Filter is a suite of filters used to generate new VBTs from existing VBTs, e.g., consider constructing a new version of a user data set through randomized subsampling in order to fit it in the available RAM. The remainder of this section describes the functionality of the VisIVO Server core components.

A VBT is a highly efﬁcient data representation used by VisIVO Server internally. A VBT is realized through a header file (extension .bin.head) containing all necessary metadata, and

---

6 At http://visivo.oact.inaf.it/.
7 3D Slicer, at http://www.slicer.org/.
8 At https://wci.llnl.gov/codes/visit/.
9 E.g., www.ubuntu.com.
a raw data file (extension .bin) storing actual data values. The header contains information regarding the overall number of fields and number of points for each field (for point data sets) or the number of cells and relevant mesh sizes (for volume data sets). The raw data file is typically a sequence of values, e.g., all X followed by all Y values. The header file contains the following fields:

```
float (double)
```

```
n1
```
```
n2 [ CellX CellY CellZ DX DY DZ ]
```
```
little | big
```
```
x
```
```
y
```
```
z
```
```
Vx
```
```
Vy
```
```
Vz
```

where

- `float (double)` is the data type of the storage variables used;
- `n1` denotes the number of columns in the VBT;
- `n2` denotes the number of rows in the VBT;
- `CellX, CellY, CellZ, DX, DY, DZ` are given only if the VBT represents volumetric data sets. In that case `CellX, CellY, and CellZ` represent the mesh geometry, while `DX, DY, and DZ` represent the x, y, and z size of volumetric cells;
- `little | big` denotes the endianism employed in the VBT.

### 3.1. VisIVO Importer

VisIVO Importer converts user-supplied data sets into VBTs without imposing any limits on size or dimensionality. VBTs are typically 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 be 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. However, if a username and password are specified, the prescribed username and password are employed for remote access.

The current version of VisIVO Importer supports conversion from several popular formats as follows:

**ASCII** and **CSV.** ASCII files normally contain values for a number of variables organized in columns. The columns are typically separated by white space characters, e.g. spaces or tabs. VisIVO Importer expects the first row of ASCII files to contain the names of the corresponding variables. CSV is a delimited data format that has fields/columns separated by the comma character and records/rows separated by newlines. The CSV file format does not require a specific character encoding, byte order, or line terminator format.

**VOTables.** The VOTable format is an XML standard for the interchange of data represented as a set of tables. A table is an unordered set of rows, each having a uniform format, as specified in the table metadata information. Each row in a table is a sequence of table cells, and each of these contains either a primitive data type or an array of such primitives.

**FLY.** FLY is code that uses the tree N-body method, for three-dimensional self-gravitating collisionless systems evolution (Becciani et al. 2007). FLY is a fully parallel code based on the tree Barnes-Hut algorithm; periodical boundary conditions are implemented by means of the Ewald summation method. FLY is founded on the one-side communication paradigm for sharing data among processors that access remote private data while avoiding any kind of synchronism.

**FITS Tables.** FITS (2009) is a codification into a formal standard, by the NASA/Science Office of Standards and Technology, of the FITS rules endorsed by the IAU. FITS supports tabular data with named columns and multidimensional rows.

**Gadget.** It is a freely available code for cosmological N-body/SPH simulations on massively parallel computers with distributed memory (Springel 2005). GADGET uses an explicit communication model that is implemented with the standardized MPI communication paradigm.

**Raw Binary and Raw Grid.** Raw files are simply binary memory dumps of the data points. The content of the Raw Binary data points file is a sequence of x, y, and z coordinates for each point, then a sequence of fields, one scalar for each field. Raw Grid contains only one quantity: the content of a volume file is a sequence of values, one value for each mesh point.

**TVO XML.** The TVO XML format is essentially the VOTable format enhanced with all necessary information to allow data downloading. It is used to describe the content of a generic binary file, and it contains the reference to the input files. If a filename starts with http://, sftp:// or ftp:// the remote file is downloaded using the CURL library. The TVO XML document is currently under revision of the TVO Interest group in the IVOA framework (Costa et al. 2008).

The operation of VisIVO Importer is highly optimized, requiring in most cases a short period of time (a number of seconds) even for large-scale data sets. For example, no more than 30 s are necessary for importing 30 million binary Gadget elements.

---

10 The cURL Project at http://curl.haxx.se.

11 VOTable is an XML format defined for the exchange of tabular data in the context of the Virtual Observatory: http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOAVoVoTable.
3.2. VisIVO Filters

VisIVO Filter is a collection of data processing modules to modify a VBT or to create a new VBT from 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

\[
\text{VisIVOFilters -op filterOpCode <options> [-file] InputFile}
\]

where \textit{filterOpCode} selects a specific operation to be processed (e.g., randomization), \textit{options} is a set of flags related to the specific selected operation, and \textit{InputFile} is an input VBT file or a list of VBT files. The following subsections describe in detail the operation of the main VisIVO Filter modules.

3.2.1. Randomization

This operation creates a new VBT table as a random subsample from the input table. The operation allows the user to produce a statistically identical version of the original table in order to allow efficient data analysis, or to visualize data using VisIVO Viewer. A similar VisIVO Filter operation (\textit{decimator}, see § 3.2.10) allows selection of fixed subsets of input data sets.

3.2.2. Merge Tables

This operation creates a new VBT from two or more VBTs. For each input table the operation can use either all or a subset of the existing columns. In the case of VBTs that do not have the same number of rows, the operation creates a new VBT having the size of the smallest or of the larger VBT. In the first case, longer columns will be truncated, while in the second case the smallest columns are padded with user-given values.

3.2.3. Extract Subregions

This operation creates a new VBT by extracting a subregion from an input VBT using one of two geometric primitives, namely:

- \textit{Sphere}. The center and the radius of the sphere determines the sub region to be extracted.
- \textit{Box}. The length of the box side and its corner position (or center position) determines the sub region to be extracted.

This operation can be used if the user needs to explore only a subvolume from the original one.

3.2.4. Select Rows

This operation creates a new VBT by setting limits on one or more fields of the input VBT. The operation must have a list of column names, and the limits (max-min) to define the extent of its action. Limits on more than one columns can be combined with OR or AND operators. The rows that satisfy the given limits are reported on the output VBT. The output VBT contains all the columns of the input VBT. For volumes, the Extract Subvolumes operation filter provides similar functionality (see § 3.2.8).

3.2.5. Mathematical Operations

These operations allow the user to derive new fields starting from the fields of an input VBT. As a result new columns are appended to the input VBT or (optionally) a new VBT is created. These operations are based on the function parser for C++ version 2.83 by Warp\textsuperscript{12} with some minor modifications. As reported in the function parser library, a function string is very similar to the C-syntax. Arithmetic float expressions can be created from float literals, variables or functions using the usual mathematical operators and the standard mathematical functions.

3.2.6. Point Distribution

This operation is used for distributing a scalar value throughout a regularly spaced 3D mesh adhering to a user prescribed resolution. The result is a volume computed by using a 3D Cloud-in-Cell (CIC) smoothing algorithm (Hockney et al. 1981). The user can choose the column of an input VBT to be used for distributing and the 3D mesh resolution.

3.2.7. Point Property

This operation assigns a new property to each data point on a given VBT. The operation performs the following:

1. A temporary volume is created using a field distribution (CIC algorithm) on a regular mesh through the aforementioned Point Distribution operation.
2. Using a CIC algorithm again, the new property for each data point is computed taking into account all the 3D cells the point is associated with.
3. The property is finally added as a new column into the input VBT, or a new VBT is created containing this new field only.

This operation could have a typical application in case we want to assign a color to each point starting from any property. For example, if we want to assign a color depending on the mass density, we need to distribute the mass associated with points on a regular mesh. Then the color of each data point will be determined considering the nearest cell where each point is located.

\textsuperscript{12}The Function Parser library, http://iki.fi/warp/FunctionParser/.
3.2.8. Extract Subvolumes

This operation produces a VBT representing a subvolume of an existing VBT. The operation needs a starting cell and a number of cells along each coordinate direction. It can be used to investigate only a subvolume from the original one. All the columns of the input VBT are reported on the output VBT, and the analysis on all the volume fields of a given data set could be done only on a specific region but with a reduced data size.

3.2.9. Coarse Volumes

This operation produces a coarse subvolume from an existing VBT using a plane extraction method. Planes are extracted uniformly along each of the coordinate directions. The operation needs the percentage of the input VBT the user wants to obtain. For example, if the original VBT is a mesh of $320 \times 320 \times 640$ cells and the percentage is 10%, the output VBT will be a mesh of $32 \times 32 \times 64$ cells. This operation is a mandatory task for visualizing very large-scale data sets using VisIVO Viewer.

FIG. 1.—Typical user session with VisIVO Desktop illustrating (from left to right and from top to bottom): (a) the user interface window, (b) an isosurface rendering, (c) a rendering of unstructured data points from a cosmological simulation, (d) a volume rendering and cutting planes visualisation of structured data from a fluidodynamical simulation, (e) a geometrical representation of data points from astrophysical catalogues. See the electronic edition of the PASP for a color version of this figure.
A VBT is created appending two or more VBTs.

Append Tables. A VBT is created appending two or more VBTs.

Polar Transformation. This module performs a spherical polar transformation.

Decimator. A VBT is created as a regular (i.e., not randomized) subsample from an input VBT.

Interpolate. Intermediate VBTs are produced with a linear interpolation from two input VBTs.

Show Table. An ascii table from a VBT is printed.

Sigma Contours. A VBT is produced where one or more fields of the input VBT have values within N sigma contours

Statistic. The average, min and max values, and histogram of a VBT field are produced.

Visual. A new VBT from one or more input VBTs is created, appropriate for the visualization with VisIVO Viewer. All input VBTs must have the same number of rows.

We are currently working on incorporating into VisIVO Filter a range of other modules, already available on VisIVO Desktop, such as calculation for correlation function, power spectrum, and Minkowsky functionals.

3.2.11. Performance Discussion

We tested the most common filters, namely Randomization, Select Rows, and Point Property to assess overall performance timings. Figure 2 shows the timings obtained with data sets that can be fitted entirely within the available RAM, but also those for data sets that are larger, thus requiring multiple disk read/write access. The randomization slope shown in the figures is determined by the number of read/write operations. The time to read the 3 billion input files is fixed, however, the time to write the output files depends upon the percentage value used for the randomizer operation. The actual randomization process is however linear.

The figure also demonstrates the performance of the Select Row and Point Property filters. The performance of these operations is linear, a typical behavior for all other filters in VisIVO Server.

3.3. VisIVO Viewer

VisIVO Viewer is founded on the Visualization ToolKit (VTK) version 5.2\cite{vtk2008} library for multidimensional visualization using the Mesa library version 7.0.3\cite{mesa2008} to avoid dependency upon an X Server connection. It creates 3D images of data sets, both data points and volumes can be represented. Though VisIVO Server can manage multidimensional data sets, 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 give more significant visual information than some million of elements. Usually (with 2 GBRAM) VisIVO Viewer can display up to 16 million of data elements easily, but the visual information the user can have is not more rich than that given by two or four million of displayed elements. The visualization process for large-scale data sets requires the following actions:

1. VisIVO Filters running the Randomization operation. This operation reduces the original data set to fit it into the available memory, and produces a new VBT for the visualization.
2. VisIVO Viewer running with the VBT given by the aforementioned process.

VisIVO Viewer can render points, volumes, and isosurfaces within a bounding box used for representing the coordinate system employed. Also there is support for customized look-up tables for visualization using a variety of glyphs, such as cubes, spheres, or cones. The standard output of VisIVO Viewer consists of four images corresponding to fixed camera positions and zoom factors and another image corresponding to user-defined line command options.

The parameters for these images are as follows: zooming factor equals to 1 and the azimuth and elevation of the camera are set to: (1) Az:0–El: 0; (2) Az:90–El: 0; (3) Az:0–El: 90; (4) Az:45–El: 45. The default images are always created even if the azimuth, elevation, and zooming camera are assigned with user-defined values. They can be avoided with a nodefault option, prescribed by the user.

VisIVOViewer can be also used to produce images in a given sequence of azimuth elevation and zooming that can be externally mounted to produce a movie. In addition, VisIVO Viewer allows visualization with Splotch (Dolag et al. 2008) (see § 3.3.2).

3.3.1. Data Point Views

VisIVO Viewer uses any three columns of a VBT to create a 3D coordinate system where data points can be drawn. If the 3D coordinate system is not specified in the command options, it is created with columns having names starting with X, Y and Z or Ra, De, and Mag. Visualization of many points can give scientifically interesting images, e.g., consider looking for inner filaments and clusters, using an adequate opacity factor (Fig. 3).

If the axis scale values of the selected coordinate system are very different, the produced images could be very small and sometimes could even degenerate into a single point. To avoid

\begin{footnotesize}
\begin{itemize}
\item VSI Server Web Sites: http://visivo.port.ac.uk/visivoweb/, http://visivoserver.oact.inaf.it, http://palantir7.oats.inaf.it/visivoweb/.
\item At http://www.vtk.org/.
\item The Mesa 3D Graphics Library, http://www.mesa3d.org/.
\end{itemize}
\end{footnotesize}
this effect the scale option can be employed to always scale the coordinate axes into a cubic region.

Points can be colored using a look-up (Lut) table among some predefined available look-up tables and using a logarithmic scale. A geometrical form (or glyph) can be associated with each data point in the image: e.g., spheres, cones, cylinders, and cubes can be used to represent each data point. This feature produces good and very meaningful images when few elements must be displayed. Moreover, the radius and the height of glyph elements can be scaled with two columns of the VBT. On the other hand, glyph visualization is not allowed with an excessively large number of elements. In addition, the rendering process could take a very long time and very often the effect is a totally cluttered cubic region, making the images difficult to interpret.

VisIVO Viewer, with Lut and glyphs, can display up to six properties on the same image simultaneously, a very powerful feature for meaningful visualization and effective data exploration.

### 3.3.2. Splotch Views

Splotch is a ray tracer to visualize SPH simulations, which we have recently customized to read data from a VBT. Splotch is designed to deal with pointlike data, optimizing the ray-tracing calculation by ordering the particles as a function of their depth, defined as a function of one of the coordinates or other associated parameters. Realistic three-dimensional renderings are reached through a composition of the final color in each pixel by properly calculating emission and absorption of individual volume elements. Our customized version is included in the current VisIVO Server distribution. Many options used to visualize Data Points can be used with Splotch (Fig. 4). The description of Splotch can be found in Dolag et al. (2008). Currently we are working on optimizing rendering times with Splotch by following a hybrid parallelization approach exploiting multicore CPUs in conjunction with the functionality offered by modern underlying GPUs.

### 3.3.3. Volume Views

VisIVO Viewer includes algorithms for direct volume rendering and isosurfacing (Fig. 5, top and bottom) for regular 3D meshes created from a point distribution operation (see § 3.2.6). The underlying rendering functionality is founded on the Visualization Toolkit. However for large-scale data sets the standard isosurfacing algorithm execution is slow. Further, as the resulting surface models generally contain very large numbers of tiny triangles, the overall computational performance can be degraded considerably, e.g., in generating sequences of renderings for producing movies. We have previously implemented optimized rendering algorithms in VisIVO Desktop and we are currently porting these into VisIVO viewer. The isosurfacing algorithm is optimized significantly in several ways: multiresolution, min-max blocks, point caching, and finally multithreading. Multiresolution allows us to generate lightweight preview models. Min-max blocks precompute minimum and maximum values for data blocks, thus allowing us to discard irrelevant data blocks (that is, with values below or above the isosurface threshold) during rendering very quickly. Finally, point caching ensures that no extra computations are performed for polygonal model vertices. The direct volume rendering algorithm employs similar optimizations by using multiresolution data sampling, adaptive pixel sampling, and render caching, thus allowing fast algorithm execution with high rendering quality. Coloring is implemented through user-prescribed look up tables which are selected from a library of several predefined look-up tables. VisIVO Viewer also provides a computationally efficient 3D slicer for visualization of orthogonal cross sections cutting through regular 3D meshes (Fig. 5, middle).

![Fig. 2.—VisIVO Filter Timings. Top: randomization operations for a Gadget simulation containing 3 billion elements. Bottom: select row and point property operations for different numbers of elements.](image-url)
3.3.4. VisIVO Viewer Performance

VisIVO Viewer implements a range of rendering algorithms, e.g., for points, isosurfaces, and volumes, supporting high-performance visualization. For example, the speed improvement of our optimized isosurfacing compared to standard VTK isosurfacing is normally in the range of 30–100 times faster (even without multiresolution). This performance gives users the ability to select an optimal isosurface threshold rapidly, an impossible task with the standard implementation. The rendering performance for standard astrophysical data sets (i.e., data sets typically fitting into the local PC’s memory) is very satisfactory for our purposes; e.g., typically no more than a few seconds are required to generate renderings. Nevertheless rendering using the Splotch algorithm could involve substantial computational costs; e.g., several minutes or even longer times depending upon the values assigned to the Splotch rendering parameters. Our experimental results indicate that the value chosen for smoothing length has a strong influence on overall rendering times. Typically the user needs to do some experimentation to strike a balance when choosing an appropriate smoothing length. Too large values may result in prohibitively excessive computational costs. On the other hand, too small values may result in flat renderings, i.e., without foglike effects.

Fig. 3.—VisIVO Viewer Data Points Views: a cosmological N-body simulation containing 4 million elements; the simulation elements are now colored according to their mass density using opacity 0.3; glyph representation of 57 elements obtained from a Hipparcos catalog query, M30 object with radius 40”, optical band. See the electronic edition of the PASP for a color version of this figure.

Fig. 4.—Seventy million elements of a cosmological N-body simulation visualized with Splotch. Careful inspection reveals elliptical galaxies, dwarf ellipticals, merging galaxies, and also filaments and voids. See the electronic edition of the PASP for a color version of this figure.
4. VisIVO ON THE GRID

The development of the VisIVO family visualization tools was initiated within the EU-funded project VO-TECH. The research collaborations with the Cometa Consortium in Catania, Italy, and the University of Portsmouth, UK, produced VisIVO Server. To include the possibility of handling very large-scale computational tasks, we have recently implemented a grid version of VisIVO Server. Our current implementation exploits the infrastructure of the Cometa Consortium with main grid nodes located in the Sicilian cities of Catania, Messina, and Palermo. The underlying hardware is based on IBM BladeCenter technology, each containing up to 14 IBM LS21 blades interconnected with a low latency Infiniband-4X network. Further, each blade is equipped with 2 AMD Opteron 2218 rev. F dual-core processors with a clock rate of 2.6 GHz, allowing native execution of x86 instructions in 32 and 64 bits. Overall the currently employed grid infrastructure consists of over 2,500 CPU cores (Becciani 2007). The grid version of VisIVO Server is a very powerful environment offering data analysis and exploration of very large-scale astrophysical data sets. For example, during a lengthy run for generating a highly complex numerical simulation, the system allows users to analyze and visually explore the simulation; in other words, it provides a way to monitor the simulation as it happens. A forthcoming article will discuss in detail our experiences with VisIVO Server in grid environments.

5. VisIVO WEB

VisIVO Web is a recently developed World Wide Web portal providing VisIVO Server services to the scientific community. Our purpose was to offer prospective users an intuitive and easy-to-use graphical environment for accessing the full functionality of VisIVO Server. This section gives an overview of VisIVO Web; a detailed presentation of its functionality will be given in a forthcoming publication (Fig. 6).

Users can upload and manage their data sets (by using registered or anonymous access), e.g., by using interactive widgets to construct customized renderings, or storing data analysis and visualization results for future reference. The data sets are managed internally through a relational database for preserving any metadata and maintaining data consistency. Both remote and local data sets can be uploaded, i.e., residing on a remote server or locally on a user’s PC, respectively. For remote files, the user must specify the relevant URL and optionally a username and password for authentication. Depending upon the size of the data sets under consideration, remote uploads could last a long period. To resolve this situation, VisIVO Web allows an offline mode so that users can issue upload commands and then simply close their current session—a follow-up e-mail message

---

16 The VOTECH project, http://eurovotec.org.
Fig. 6.—VisIVO Web Operational Scenario. Top figure shows uploading of data sets from a user’s PC or from a server remotely. Middle figures demonstrate the application of VisIVO Filter operations. Bottom figure illustrates the final display using customized color tables and interactive widgets for setting VisIVO Viewer camera positions. A detailed description of the VisIVO Web operational scenario can be found in the user guide, available at the VisIVO Web site. See the electronic edition of the PASP for a color version of this figure.
typically gives notification when the uploading operation is completed.

Once data are uploaded, a sequence of simple actions is required to rapidly obtain meaningful visualizations. Typically, various VisIVO Filter operations are performed, and VisIVO Web automatically displays all applicable VisIVO Filter operations allowing for graphical input of the relevant parameters. Finally, the VisIVO Server Viewer is employed for display. A check box located on the righthand side of any processed data set is used in conjunction with the view button to create user-prescribed VisIVO Viewer views. VisIVO Web is currently implemented in Web sites hosted by the University of Portsmouth, the INAF Astrophysical Observatory of Catania, and Astronomical Observatory of Trieste, Italy.\(^7\)

6. INSTALLATION NOTES

VisIVO Server is an open-source software package available to the scientific community at SourceForge, a Web site that provides free hosting to open source software development projects.\(^8\) VisIVO Server requires at a minimum the following libraries: gcc 4.3, cmake 2.6, xerces 2.8, cfitsio 3.1, curl 7.19.2, vtk 5.2, and mesa 7.0.3. These libraries must be installed on the underlying server and their compilation can be done either dynamically or statically. Standard installation of VisIVO Server does not require an X Server, so that no further restrictions are imposed (e.g., firewalls do not typically allow X Windows to be opened remotely).

7. SUMMARY

VisIVO is an integrated suite of tools and services specifically designed for the Virtual Observatory. This article focused on VisIVO Server, a new platform for astrophysical visualization of large-scale data sets that can be easily installed on any computing server. VisIVO Server is an open-source collection of data processing and visualization modules allowing fast rendering of 3D views. VisIVO Server is founded on the functionality of VisIVO Desktop, our previously developed standalone astrophysical visualization application for standard PCs. The defining characteristic of VisIVO Server is that no restrictions are imposed on dimensionality of data sets.

We described the constituent parts of VisIVO Server in detail, namely VisIVO Importer, VisIVO Filter, and VisIVO Viewer. VisIVO Importer converts user prescribed data sets into a highly efficient internal data format employed by VisIVO Filters and VisIVO Viewer. VisIVO Filters are a collection of several processing modules for constructing customized data tables from VisIVO Importer. Finally, VisIVO Viewer creates 3D views of astrophysical data sets, currently supporting points, volume isosurfaces, and ray-tracing using a customized algorithm called Splotch.

We discussed different visualization scenarios and demonstrated example renderings. Users can obtain meaningful visualizations rapidly while preserving full and intuitive control of the relevant visualization parameters. We briefly described a computational grid implementation of VisIVO Server and also discussed VisIVO Web—a custom-designed web portal supporting services based on the VisIVO Server functionality. The VisIVO Server software is distributed under a GPL license for noncommercial use only. It is an open source project and can be simply downloaded from the SourceForge code repository.

VisIVO Server and VisIVO Desktop can exchange data using the same internal data format (VBT). They can be closely integrated, but are also complementary and independent of each other. For example, VisIVO Server can create a preview of a data set or an appropriately defined subsampled version, then VisIVO Desktop can be employed for visualization and subsequent interaction.

Currently we are working toward an integrated visualization functionality so that VisIVO Server and VisIVO Desktop are wrapped around a unified visualization kernel. Our vision is to maintain VisIVO Server for high-performance computing (e.g., grid environments, large underlying memory of 100 GB or more) and VisIVO Desktop to support interactive front-end interfaces providing an advanced GUI and exploiting the full graphical capabilities of modern PCs (e.g., fast multiprocessor architectures, emerging GPUs). Toward this aim, we will need to define a full communication protocol between the server and desktop versions to allow applications for each to be developed independently.

The authors would like to acknowledge the support of Dr. Roberto Munzone for his significant contribution during the development of the Gadget reader, and Dr. P. Manzato and Dr. M. Molinaro from the INAF Astronomical Observatory of Trieste for several useful discussions during the course of this project. Our special thanks to F. Vitello from the Cometa Consortium for developing parts of the VisIVO website. Finally, we would like to thank Professor B. Nichol from the Institute of Cosmology and Gravitation, University of Portsmouth for his support during the development of VisIVO Server. This work has made use of results produced by the PI2S2 Project managed by the Consorzio COMETA, a project cofunded by the Italian Ministero dell’Istruzione, Universita’ e Ricerca (MIUR) within the Piano Operativo Nazionale Ricerca Scientifica, Sviluppo Tecnologico, Alta Formazione (PON 2000–2006). More information is available at http://www.pi2s2.it and http://www.consorzio-cometa.it. The work reported in this paper was partially supported by the HPC-EUROPA++ (project number 211437), under the EC Coordination and Support Action Research Infrastructure Programme in FP7 and by the University of Portsmouth Promising Researcher’s award.

\(^7\) VisIVO Server Web Sites: http://visivo.port.ac.uk/visivoweb/, http://visivoserver.oact.inaf.it, http://palantri7.oats.inaf.it/visivoweb/.

\(^8\) At http://sourceforge.net/.
REFERENCES

Becciani, U., Comparato, M., & Antonuccio Delogu, V. 2007, Comp. Phys. Comms., 176, 211
Becciani, U. 2005, New Grid Infrastructure in Sicily, Computational Grids for Italian Astrophysics: Status and Perspectives, ed. Benacchio, L., & Pasian, F., (Rome: Polimetrica International Scientific, 2007, 179
———. 2009, Mem. Soc. Astron. Italiana Suppl., 13,10
Bonnarel, F., et al. 2000, A&AS, 143, 33
Costa, A., Manzato, P., Becciani, U., Comparato, M., Costa, V., Gasparo, F., Gheller, C., Grillo, A., et al. 2008, PASP, 120, 933
Comparato, M., Becciani, U., Costa, A., Larsson, B., Garilli, B., Gheller, C., & Taylor, J. 2007, PASP, 119, 858, 898
Dolag, K., Reinecke, M., Gheller, C., & Imboden, S. 2008, New J. Phys., 10, 125006
Hockney, R. W., Eastwood, J. W., & Buchert, T. 1981, Computer Simulation using Particles (New York: McGraw Hill)
Ivezic, Z., Axelrod, T., & Brandt, W. N., et al. 2008, Serbian Astron. J., 176, 1
York, D. G., Adelman, J., & Anderson, J. E., et al. 2000, AJ, 120, 1579
Mohr, J., Adams, D., & Barkhouse, W., et al. 2008, Proc. SPIE, 7016, L70160
Price, D. J. 2007, PASA, 24, 159
Schilizzi, R., Dewdney, P., & Lazio, J. 2008, Proc. SPIE, 7012
Springel, V. 2005, MNRAS, 364, 1105
The Flexible Image Transport System (FITS). http://fits.gsfc.nasa.gov/
The LOw Frequency ARray 2009, http://www.lofar.org/
The Sloan Digital Sky Survey 2008, http://www.sdss.org/
The Tool for OPerations on Catalogues And Tables (TOPOCAT). 2009 http://www.starlink.ac.uk/topcat/
Viceconti, M., Astolfi, L., & Leardini, A., et al. 2004, IEEE Proc. Conf. on Information Visualisation (IV’04), 15