VisIVOWeb: A WWW Environment for Large-Scale Astrophysical Visualization

A. Costa, U. Becciani, P. Massimino, M. Krokos, G. Caniglia, C. Gheimer, A. Grillo, and F. Vitello

ABSTRACT. This article presents a newly developed Web portal called VisIVOWeb that aims to provide the astrophysical community with powerful visualization tools for large-scale data sets in the context of Web 2.0. VisIVOWeb can effectively handle modern numerical simulations and real-world observations. Our open-source software is based on established visualization toolkits offering high-quality rendering algorithms. The underlying data management is discussed with the supported visualization interfaces and movie-making functionality. We introduce VisIVOWeb Network, a robust network of customized Web portals for visual discovery, and VisIVOWeb Connect, a lightweight and efficient solution for seamlessly connecting to existing astrophysical archives. A significant effort has been devoted for ensuring interoperability with existing tools by adhering to IVOA standards. We conclude with a summary of our work and a discussion on future developments.

Online material: color figures

1. INTRODUCTION

The primary goal of scientific visualization is to create images and animations to aid scientists in understanding complex data sets: e.g., large-scale multidimensional data sets obtained either from numerical simulations or real-world astrophysical observations. The typical sizes of modern astrophysical data sets are increasing steadily, due to continuous advances in computational power and instrumentation. As a result such data sets often require storage in a distributed way: e.g., the Millennium-II simulation (Springel 2005), which produced nearly 20 Tbyte. Present and next-generation sky surveys are also of constantly increasing sizes and complexity: e.g., the Sloan Digital Sky Survey (Abazajian 2009), which offers a catalog of 15.7 Tbyte. Moreover, a significant effort has been devoted in handling highly complex numerical simulations in the context of the Virtual Observatory (Padovani 2004; Allam 2005): e.g., the Italian Theoretical Virtual Observatory (Costa 2008) and the German Astrophysical Virtual Observatory (Boylan-Kolchin 2009), offering dark matter halo catalogs and merger trees from the Millennium-II.

Modern visualization can aid astrophysicists in gaining good insights into highly complex data sets, through rapid and intuitive discovery of familiar patterns and correlations between properties, without involving CPU-intensive analysis codes. As a result, suitably constructed visualization tools can be instrumental for future astrophysical advances, e.g., to allow comparing two or more simulations or even appropriately correlating simulations and observational data sets. As an example, consider the massive halo-formation process or interaction among galaxies seen in the evolution of a simulation and similar observed objects. This article presents a World Wide Web (WWW) environment for visualization of large-scale astrophysical data sets that can be installed easily in data centers hosting the data sets, offering astrophysicists an array of advanced visualization and exploration interfaces, as well as VisIVOWeb Connect, a lightweight and efficient solution for seamlessly connecting to existing astrophysical archives. The way users can gain access to the system and import their data sets is described in §§ 3 and 4, respectively. Section 5 describes the underlying data model required to support VisIVOWeb functionality and outlines a typical operational scenario. The visualization algorithms supported, together with the relevant visualization interfaces, are presented in §§ 6 and 7. We discuss the creation of scientific movies in § 8, focusing on a
specific example for demonstration. We then outline VisIVOWeb Network and VisIVOWeb Connect, together with our recent dissemination activity. Finally, we conclude with a summary of our work, including pointers to some future developments.

2. RELATED WORKS

VisIVOWeb is a customized WWW interface wrapped around VisIVOServer (Becciani 2010) that can be used through common Internet browsers. We have implemented VisIVOServer previously through a comprehensive collection of modules for processing and visualization of astrophysical data sets. The software is open-source, written in C++. The underlying visualization engine is founded on the functionality of the Visualization Toolkit. As VisIVOWeb is wrapped around VISO Server, it offers fast rendering of customized 3D views of large-scale astrophysical data sets in a WWW environment that provides a unique mix of data management and multidimensional visualization tools.

To the best of our knowledge, there are no other comparable environments available in the Virtual Observatory (Padovani 2004; Allam 2005). The rendering process is typically computationally intensive, often requiring millions of floating-point and integer operations for generating single 3D views (Crocket 1997). This is the reason for developing a fully C++ implementation, rather than optimizing other existing Java-based environments: e.g., TOPCAT (Taylor 2005). Compared with TOPCAT our rendering capability is significantly enhanced as we can handle large-scale multidimensional data sets. The visualization functionality provided by VisIVOWeb is also very different compared with a plotting tool such as VPlot (Kale 2004) and Aladin (Bonnarel 2000). The latter is simply a software sky atlas allowing the user to visualize digitized astronomical images and to also superimpose entries from astronomical catalogs or databases. Among the visualization algorithms provided by VisIVOWeb for handling pointlike data sets, Splotch (Jin 2010) deserves particular attention. A customized ray tracer is employed for effectively creating meaningful 3D views of modern cosmological simulations. The intensive underlying calculation is optimized by preprocessing the relevant particles, and recent developments exploit GPUs for hardware acceleration. The rendering is achieved through composition of the final color in each pixel by calculating emission and absorption of individual volume elements. Splotch is released under GNU license and is integrated in VisIVOServer fully. We focus on visualization interfaces within VisIVOServer in § 7. The next section outlines the typical user scenario for gaining access to our system.

3. SYSTEM ACCESS

VisIVOWeb is supported fully by commonly used Internet browsers; e.g., Internet Explorer, Firefox, Chrome, Safari and Opera. JavaScript, Java, pop-up windows, and cookies must be enabled.

The system can be accessed using either of the following modes: authorized or anonymous. To employ an authorized account, users are required to complete a Web form; their system account is then activated following approval (typically, in a couple of hours). On the other hand, anonymous accounts do not require approval, they are simply created through the home page of the system, and they are activated immediately. These accounts employ randomly generated user names and empty passwords. Assuming there is sufficient amount of disk space available, there is no limit on the max number of anonymous accounts employed by the system.

All users (no matter if anonymous or authorized access) are provided with their own individual staging area within the system for uploading and managing their data sets and, more importantly, the visualizations produced from these data sets: e.g., images and movies. For anonymous accounts, although access to full functionality for processing and visualization is provided, they are valid for no more than four days since last user access. At the end of this period all user-uploaded data sets and created visualizations are removed automatically. Authorized accounts are valid for two months since last user access. Prior to expiration a warning e-mail is sent, then a simple login into the system by the user can extend the validity of an account automatically. The typical scenario for importing data sets into the system is outlined in § 4.

4. IMPORTING DATA SETS

Before any analysis and/or visualization is performed, user data sets must be imported into VisIVOWeb. The system currently supports most commonly employed astrophysical data formats: e.g., ASCII and Catalog of Suspected Variables (CSV) tables, VOTables (Ochsenbein 2009), FITS tables, GADGET-2 (Springel 2005), and raw binary and raw grid. Furthermore, other data formats are supported that target either specific applications used by a sizable scientific community or those of particular scientific interest. Consequently, the system supports the outcomes of the code FLY (Becciani 2007) and TVO XML, a VOTable that can be used to describe the content of generic binary files. This descriptor is currently under revision by the Theory Interest Group within the International Virtual Observatory Alliance (IVOA) framework.\footnote{See http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/IvoaTheory.}

Several interfaces are provided for importing, depending upon the data format of the input data sets. VisIVOWeb can handle point clouds and volumetric data sets; the latter typically

\footnote{See http://sourceforge.net/projects/visivoserver/.
10 See http://sourceforge.net/projects/visivoserver/.
11 See http://www.vtk.org/.}
Data sets imported into VisIVOWeb are always converted into an internal data representation called a VisIVOBinaryTable (VBT). A VBT can describe both point data sets and volume data sets; for detailed information on its structure and header, the reader is referred to Becciani (2010). The importing of a data set may produce one or more VBTs; e.g., a GADGET-2 snapshot can generate six VBTs, corresponding to different particle types associated with gas, halo, disk, bulge, stars, and boundary. 

Creating database objects.—This depends on the size of the data file to be processed and on the network bandwidth. 

Converting.—This depends on the size of the data file and on the data type and number of tables that it contains. This operation is required for conversion of data sets into the internal data representation as described in § 4.1.

Creating database objects.—This is typically a very short period of time that is required for creating all the necessary data structures and metadata related to importing. Section 5 describes the details of the database objects.

The graph in Figure 1 shows the scaling of overall importing times versus data set sizes. Our tests were performed on a dual-core AMD Opteron processor 280 with 4 Gbyte RAM. Several binary data sets were imported with their sizes varying from 380 Mbyte up to 12 Gbyte. The linear scaling observed suggests that handling large data sets is effectively restricted only by the underlying file-system limitations.

5. DATA MANAGEMENT

The underlying data model is implemented through a relational database. This allows users a private staging area within the system for managing their data sets and the images and movies produced from such data sets. Typically, a number of operations are employed prior to any analysis and/or visualization, e.g., for extracting interesting features. This section describes the developed database objects (§§ 5.1 and 5.2) and the relevant interface (§ 5.3) and outlines a representative number of available operations (§ 5.4).

5.1. Metadata Objects

The metadata (or attributes) associated with uploaded data sets are encapsulated by customized VisIVOFile objects. These objects are inherited from VisIVOMetadata objects, that encapsulate attributes such as owner or creation time, adding new attributes such as the file format employed (e.g., CSV or FITS tables). Moreover, the objects VisIVOTable and VisIVOVolume are defined by inheriting from VisIVOMetadata for representing VBTs (see § 4.1). VisIVOTable objects introduce attributes such as endianness, number of fields, type, and number of elements. VisIVOVolume objects introduce attributes such, number and size of cells for each dimension (Fig. 2). As an example, consider importing into our system GADGET-2 data sets. A VisIVOFile object will be generated by the original file and importing may produce six (one for each particle type) VisIVO-Table objects.

5.2. Operation Objects

An operation in VisIVOWeb can be described as a many-to-many relation connecting VisIVOMetadata objects. We can simplify this relation by introducing a VisIVOOperation object to treat a many-to-many relation as two one-to-many relations (Fig. 2). In other words, many VisIVOMetadata objects can possibly generate several VisIVOMetadata objects by using a single VisIVOOperation. We also define the objects VisIVOOpInList/VisIVOOpOutList as aggregations of VisIVOMetadata objects. As an example, consider merging VBTs representing point clouds. VisIVOOpInList is the aggregation of the original VisIVOMetadata objects corresponding to the relevant VBTs, and VisIVOOpOutList will contain a new VisIVOMetadata: i.e., the result of the operation.

5.3. Operational Scenario

We provide our system with a graphical Web interface that allows data to be easily stored and accessed. Data are organized in a treelike structure (Fig. 3). In this way a user is allowed to select one or more data sets and to use them as members of an operation or as elements of a visualization pipeline. Different objects in the data model (such as files, volumes, and tables) have different icons. This interface allows us to manage object metadata and to select an individual object or a whole branch to be deleted.
5.4. Operations

A single-point VBT is required by the following operations:

Randomization.—A new point VBT is produced by randomly selecting particles according to user-prescribed percentage values.

Math.—Several mathematical operations for combining columns of VBTs. A suitable drop-down menu is provided to assist users in inputting any required algebraic expression.

Select rows.—This is a multidimensional selector. The user can impose constraints to specify min / max values for particular fields of a VBT. The result of the operation is a new point VBT matching the prescribed criteria.

Append.—A new VBT is created from a number of VBTs with an identical number of columns.

Extraction.—A new VBT is created by identifying a region of interest using a box/sphere. The user must supply suitable columns to be used as Cartesian coordinates, and also dimensions and placement of the extracting box/sphere.

Point distribution.—A point VBT is transformed into a volumetric VBT. Each point is distributed to nearby voxels using one of the following algorithms: cloud-in-cell, nearest grid point, or triangular-shaped cloud (Hockney 1981).

A point VBT and a volume VBT are required by the following operation:

Grid2Point.—For distributing the values of a volumetric VBT within a point VBT in the same domain using the cloud-in-cell, nearest grid point, or triangular-shaped cloud algorithm.

The following operations are for volume VBTs only:

Coarse volumes.—This reduces a VBT using a plane for selecting a region of interest according to user-prescribed percentage values.

Subvolumes.—This operation extracts rectangular regions of interest from prescribed VBTs.

Several new operations are currently planned to be implemented.13

---

13 To obtain the most recent developments, the reader is referred to source forge.net: http://sourceforge.net/projects/visivoweb/.
6. VISUALIZATION

The standard visualization views in VisIVOWeb employ orthographic projections. The focal point is fixed in the center of the box to be observed, and users can prescribe the camera azimuth and elevation. The system supports several algorithms based on the Visualization Toolkit\textsuperscript{14} and Splotch (Jin 2010). The former is suitable for point clouds and volumetric data sets while the latter is a customized ray tracer suitable for particle-based data sets. To decouple our visualization engine from the hosting computer infrastructure a default installation of VisIVOWeb employs the Mesa libraries,\textsuperscript{15} an open-source implementation of OpenGL. Nevertheless, a customized installation of the system is feasible so as to exploit fully the hardware capability of the underlying hosting server.

6.1. Point Clouds

Point clouds can be visualized using either VTK-based algorithms or Splotch, provided that suitable columns are selected from the input VBT for a Cartesian system of coordinates to be used for rendering. The entire computational box of an $N$-body simulation ($z = 0.1$) is displayed at the top of Figure 4; its size is 70 Mpc hr$^{-1}$. The image at the bottom illustrates fine details in a sub-box from the same simulation. By using predefined color tables, scalar values can be mapped to colors so as to modulate VisIVOWeb-rendered views based on values of specific properties of individual points in a cloud. Apart from colors, user-defined shapes (e.g., cylinders, cones, or spheres) can also be attached to particular points so that geometrical characteristics (e.g., height or radius) can be modulated according to values of specific point properties. Splotch also operates by mapping scalar data (a column of the input VBT) to colors by indexing though a set of predefined color palettes. A summary of mappings of VBT fields to visible elements in the VisIVOWeb views is shown in Figure 2.

\textsuperscript{14} See http://www.vtk.org/.
\textsuperscript{15} See http://www.mesa3d.org/.
is shown in Table 1 for VTK-based and Splotch rendering, respectively.

6.2. Volumes

Volumetric data sets can be visualized by using either volume rendering algorithms or isosurfacing. Figure 5 illustrates the rendering of a major merger galaxy cluster simulated with high spatial resolution using the Enzo 1.5 cosmological adaptive mesh refinement (AMR) code. The upper image is a temperature profile visualized using a volume rendering technique. The lower image is the density profile visualized using isosurface technique. The box size is \(187 \text{ Mpc hr}^{-1}\) and voxel number is \(544^3\). The isosurfaces in the bottom image correspond to prescribed constant values attained by the relevant scalar function. Inspecting the rendering results for a number of different constant values reveals the inner structure complexity of the underlying data sets.

7. VISUALIZATION INTERFACES

The visualization interfaces provided within VisIVOWeb are controlled by a widget interactively for setting viewing parameters: e.g., zoom in/out factors, azimuth, and elevation (Fig. 6). This widget is implemented as a Java applet controlling an interactive box within the user’s browser. The user can interact with the widget by using either the computer mouse or the buttons built into the widget. Once the viewing parameters are fixed the user can render via the display button.

7.1. Opacity Function and Color Maps

An interactive editor is provided (Fig. 7) that allows users to create customized color maps and opacity functions for visualization. On the left-hand side of the interface the opacity function
function and color map are displayed. Using the computer mouse a user can edit this opacity function, not only by inserting additional nodes, but also by updating existing nodes or deleting them by simply dragging them out of the window. The overall opacity function is constructed by linearly interpolating all nodes. The user can also adjust each node’s color through the color selector, hue-saturation-brightness editor, or red-green-blue editor; the corresponding color map is updated in real time. Color maps and opacity functions can be saved for later use in system templates.

7.2. Interfaces

The graphical user interface for VTK-based rendering is shown in Figure 8. The user can select VBT columns to construct a Cartesian system of coordinates for rendering. The scale option is employed for transforming coordinates so that rendering occurs within a cubical box if required. Changes in the opacity value are typically used for revealing and/or masking a data set’s inner structures. Finally, using the lookup table option a VBT column can be selected to be mapped into a color palette. A set of predefined color tables are accessible by default. The user can also apply shapes (e.g., spheres, cylinders, or cones) during rendering of individual points, modulating their geometric characteristics by employing a specific property. The graphical user interface for Splotch rendering is illustrated in Figure 9. First, the VBT columns for constructing the Cartesian system of coordinates for rendering must be chosen. Subsequently, several scalar values need to be selected by the user to be associated to intensity and color, smoothing length and smoothing-length factor (Monaghan 1988), gas gray absorption, and brightness, respectively.

8. SCIENTIFIC MOVIES

Scientific movies are useful not only to scientists for presenting and communicating their research results, but also to museums and science centers for introducing the general public into complex scientific concepts. The creation of a movie represents a significant challenge for the underlying computational resources as often hundreds or thousands of high-quality images must be produced. For example, creating a flythrough lasting 5 minutes within a 16 million particle cosmological simulation requires 38,200 s in a dual-core AMD Opteron processor 280 with 4 Gbyte of RAM memory.

| X | Y | Z | Color | Intensity | Radius | Height |
|---|---|---|-------|-----------|--------|--------|
| VTK-based | * | * | * | * | ... | * | * |
| Splotch | * | * | * | * | * | ... | ... |

8.1. Temporal Parallelism

To alleviate the high computational demands a temporal parallelism is adopted for decomposing the overall computation. The fundamental unit of computation is a complete frame and an individual processor is assigned with a number of frames to be rendered. Our system is fully integrated with OpenPBS\textsuperscript{16} and LSF.\textsuperscript{17} A number of frames from scientific movies created on a cluster of 54 CPU cores using the portal at INAF-Catania are shown in Figure 5.

8.2. Operational Scenario

Movies are currently created in VisIVOWeb by using several interfaces. First, users can create flythroughs by moving a

\textsuperscript{16}See http://www.openpbs.org/.
\textsuperscript{17}See http://www.platform.com/.
camera along a motion path that is prescribed within the domain of a data set.

Second, flythroughs can be produced by intermediate snapshots specified as camera positions/orientations; the system generates a movie with a camera path containing the specified positions/orientations.

Third, movies can be created by interpolating several steps of a time evolution. The user can browse a cosmological time evolution and choose two or more coherent data sets. The system will then produce the necessary number of intermediate data sets by calculating particle positions and applying boundary conditions as necessary. This approach can be very useful, e.g., in revealing galaxy formation or large-scale structures such as galaxy clusters.

8.3. A Movie from a 512 Million Particle Simulation

One of the most challenging tasks in performing large-scale cosmological simulations is monitoring the outcomes, so that if anomalies are observed the simulation run is corrected appropriately. Typically, such monitoring requires visualization of several snapshots at different redshift, each being tens of gigabytes.

We recently performed a large-scale structure of the universe simulation (100 Mpc h$^{-1}$) containing 800$^3$ particles. The run was performed on the COMETA Consortium Grid$^{18}$ using FLY (Becciani 2007). The simulation was carried out over three months using 250 CPU cores; each data snapshot was 12 Gbyte.

We employed VisIVOWeb to extract 8 million particles and to identify a subregion (15 Mpc hr$^{-1}$) by using the operations described in § 5.4. We then interpolated the original snapshots in the time domain, producing a large number of intermediate frames that have been used in creating a movie showing the fluid time evolution. Overall, 1500 intermediate data sets were created requiring a CPU time of two months. The final movie employs a frame rate of 10 fps.$^{19}$ The whole data archive handled for the movie creation was in excess of 18 Tbyte.

9. VISIVOWEB CONNECT AND NETWORK

VisIVOWeb Connect is a lightweight solution that allows seamless connection of VisIVOWeb to any astrophysical archive by adopting an http-based approach. Using this technology users can view and manage data sets from any Internet archive directly within VisIVOWeb. The system can handle any data archive and it allows automatic uploading of data sets into a user-specified VisIVOWeb portal.

$^{18}$ See http://www.pi2s2.it/.
$^{19}$ See http://astrct.oact.inaf.it/visivo/cosmo/.

Fig. 6.—Interactive widget employed for adjusting rendering settings for a cubical subsample in an $N$-body cosmological simulation ($z = 0.1$). See the electronic edition of the PASP for a color version of this figure.
To exploit the VisIVOWeb Connect functionality the relevant data sets must be encoded using a supported data format and be accessible through a supported data transfer protocol. Assuming that these conditions are observed, integrating VisIVOWeb Connect with any Internet archive is straightforward, simply requiring a URL containing locations/data types of the data sets to be uploaded.

Recent developments begun the deployment of a network of VisIVOWeb portals. At the time of writing, active nodes exist at INAF-Catania, INAF-Trieste, and University of Portsmouth. The creation of a network of VisIVOWeb portals allows the user to choose the closest portal in order to minimize uploading times. Furthermore, it allows users to potentially customize different portals with tools specific to particular archives.

10. CURRENT STATUS AND FUTURE WORK

The visualization functionality supported by VisIVOWeb is rich, and we are currently in the process of developing several demonstrators to be used for formal evaluation in the context of specific astrophysical communities using both numerical simulations\(^\text{20}\) and real-world observations\(^\text{21}\). The initial feedback is very encouraging and we are working on fixing software bugs prior to our next major release, but also pursuing several enhancements. First, several new operations\(^\text{22}\) are planned to be implemented, to expand significantly the collection of operations presented in § 5.4. Second, we are working on upgrading the core visualization functionality to the latest version of the Visualization Toolkit, and the recently released version of Splotch (Jin 2010). The latter work is of particular importance, as it offers the exciting possibility for exploiting hybrid

---

\(^{20}\) See http://astrct.oact.inaf.it/visivo/cosmo/.

\(^{21}\) See http://research.icg.port.ac.uk/.

\(^{22}\) The reader is referred to sourceforge.net (http://sourceforge.net/projects/visivoweb/) for the newest developments.
massively parallel architectures containing large numbers of multicore CPUs and CUDA-enabled GPUs. To exploit high-performance computing, we have already integrated within our system the functionality to allow authenticated users to submit movie creation jobs using the EGEE grid resources, based on the computational capability provided by the grid operated by the COMETA consortium.

Finally, depending upon the user feedback obtained during the formal evaluation activity and within the scope of specific astrophysical communities, we are planning to incorporate new functionality, e.g., for rendering streamlines and AMR data sets efficiently or designing highly customized opacity functions. Such algorithms either already exist in the underlying visualization libraries (but they are simply not enabled) or they can be developed readily. This work will also dictate the direction of development for customizing future VisIVOWeb portals employing VisIVOWeb Connect. At the time of writing, this technology is used for connecting our system seamlessly with the

Fig. 9.—Graphical user interface employed in VisIVOWeb for Splotch rendering. See the electronic edition of the PASP for a color version of this figure.
Italian Theoretical Virtual Observatory Web archive (Costa 2008) and the UK mirror of the Sloan Digital Sky Server (data release 7) CasJobs service (Caniglia 2009).

There are a number of user guides for VisIVOWeb available to download from the nodes operating currently (INAF-Catania, INAF-Trieste, and University of Portsmouth. Each node also contains streaming videos of online tutorials; the relevant slides are available at the VisIVO portal. The underlying visualization technology documents are accessible at the subversion repository. The functionality of our system has been presented in the VO-Day events run recently across Italy, organized in the framework of Euro VO Astronomical Infrastructure for Data Access and IVOA.

11. SUMMARY

Modern visualization can aid astrophysicists in gaining good insights of highly complex data sets, generated either from real-world observations or numerical simulations, by providing suitable tools for rapid and intuitive visual discovery. We have introduced an open-source WWW environment called VisIVOWeb for visualization of large-scale astrophysical data sets offering astrophysicists an array of advanced visualization tools that can be exploited effectively using common Internet browsers. The system offers full data management functionality so that users can upload and manage their data sets, e.g., by using interactive widgets to construct customized renderings or generating meaningful astrophysical animations or even storing their results (movies or otherwise) for future reference. To the best of our knowledge, there are no other comparable open-source environments available to astrophysicists in the Virtual Observatory.

We reviewed relevant work and introduced the way users can gain access to the system for importing their data sets. We discussed the underlying model for data management, focusing on the developed database objects, the relevant interface and a number of currently available operations. We also presented the visualization algorithms supported by the system, founded on the Visualization Toolkit and Splotch, together with the relevant visualization interfaces, including an interactive editor for constructing customized opacity functions. The operational scenario for generating scientific movies from large-scale data sets was then introduced discussing a specific example for demonstration based on a large-scale simulation containing 512 million particles. The whole data handled by VisIVOWeb for this movie creation was in excess of 18 Tbyte. We then introduced VisIVOWeb Connect/Network for networking VisIVOWeb portals and connecting them seamlessly to existing astrophysical archives. Finally, we concluded with a description of the current status and pointers to a number of future developments.

The authors would like to thank Marco Comparato for his contribution in realizing the first version of VisIVOWeb. We would also like to thank Patrizia Manzato and Marco Molinaro for their intensive debugging activity during initial development of VisIVOWeb. Original data sets were provided by the IRA-CINECA Simulated Cluster Archive. This work has made use of results produced by the PI2S2 project managed by the Consorzio COMETA, co-funded by the Italian Ministero dell’Istruzione, Universita e Ricerca under Piano Operativo Nazionale Ricerca Scientifica, Sviluppo Tecnologico, Alta Formazione (PON 2000-2006). Finally, we thank Luigia Santagati for proof-reading.

REFERENCES
Abazajian, K., et al. 2009, ApJS, 182, 2, 543
Allam, S. S., et al. 2005, BAAS, 37, 1200
Becciani, U., & Antonuccio-Delogu, V., et al. 2007, Comput. Phys. Commun., 176, 3, 211
Becciani, U., & Costa, A., et al. 2010, PASP, 122, 887, 119
Boylan-Kolchin, M., Springel, V., White, S. D. M, Jenkins, A., & Lemson, G. 2009, MNRAS, 398, 3, 1150
Bonmarel, F., et al. 2000, A&A, 143, 33
Caniglia, G., Krokos, M., Becciani, U., Gheller, C., Nichol, R., Comparato, M., Costa, A., Grillo, A., et al. 2009, Second Int. Conf. Visualization (Piscataway: IEEE), 10
Costa, A., & Manzato, P., et al. 2008, PASP, 120, 870, 933
Crocket, T. W. 1997, Parallel Comput., 23, 7, 819
Hockney, R. W., & Eastwood, J. W. 1981, Computer Simulation Using Particles (New York: McGraw-Hill)
Jin, Z., Krokos, M., Rivi, M., Gheller, C., Dolag, K., & Reinecke, M. 2010, preprint (arXiv:1004.1302)
Kale, S., Vijayaraman, T. M., Kembhavi, A., Krishnan, P. R., Navelkar, A., Hegde, H., Kulkarni, P., & Balaji, K. D. 2004, ASP Conf. Ser. 314, CONF. NAME HERE (San Francisco: ASP), 350
Monaghan, J. J. 1988, Comput. Phys. Commun., 48, 88
Ochsenbein, F., & Williams, R. 2009, IVOA Recommendation 30 November 2009, http://ivoa.net/Documents/VOTable/
Padovani, P., et al. 2004, 424, A&A, 545
———. 2005, MNRAS, 364, 1105
Springel, V., et al. 2005, Nature, 435, 629
Taylor, M. B. 2005, in ASP Conf. Ser. 347, Astronomical Data Analysis Software and Systems XIV (San Francisco: ASP), 29