Interactive Visualization for the UNICORE Grid Environment

Piotr Bała[2,1], Krzysztof Benedyczak[2], Aleksander Nowiński[1], Krzysztof S. Nowiński[1], and Jarosław Wypychowski[1]

[1] Warsaw University
Interdisciplinary Center for Mathematical and Computational Modelling, know@icm.edu.pl
[2] Faculty of Mathematics and Computer Science
Nicolaus Copernicus University, bala@mat.uni.torun.pl

Abstract. A description of interactive visualization plugins for the UNICORE grid system is provided. The plugins allows for attaching to running UNICORE jobs, downloading current state of computation and visualization of the results. The plugin set contains a general purpose 2D visualization system Sapphire and an example plugin MolDyAna for 3D visualization of the progress of molecular dynamics jobs. The Sapphire subsystem contains user configurable tools for simple plots of different types and 2D data plotting tools, together with an extensive GUI. MolDyAna provides both standard methods of animation and graphing and several advanced postprocessing and visualization techniques

1 Introduction

Recent advances in the grid technology created number of tools which provide user's interface to distributed resources [1]. Computational grids allowed users to use variety of geographically distributed resources and to select and aggregation of them across multiple organizations for solving large scale computational and data intensive problems. In order to attract users, grid interfaces must be easy to install, simple to use and able to work with the different operating system and hardware platforms. The important feature required by the users is ability to perform scientific visualization.

Recent grid activity has been significantly oriented towards utilization of the computational resources for the data processing and number crunching rather than visualization. This approach was also influenced by the fact that most of computational intensive applications have no dedicated user interfaces. Some expensive commercial graphical tools are available but they are treated as separate applications run locally on the users workstation and can hardly be integrated with the grid environment. Using these applications the user can prepare input or perform advanced postprocessing. The results are stored in the files used as input for another job to be run on the grid. The grid middleware is used to transfer input and output files and to submit and control job.

In order to provide user with integrated environment attempts to integrate advanced visualization with the grid environment have been performed. Most of them are limited by the existing technology.

There are two most important barriers inhibiting creation Of effective grid visualization tools: lack of the visualization capabilities in mostly script based grid tools and the fact that the visualization software is build using traditional programming languages and libraries such as Tcl, Gtk or GL. In effect, existing visualization software is difficult to integrate with standard grid environment built in Java.

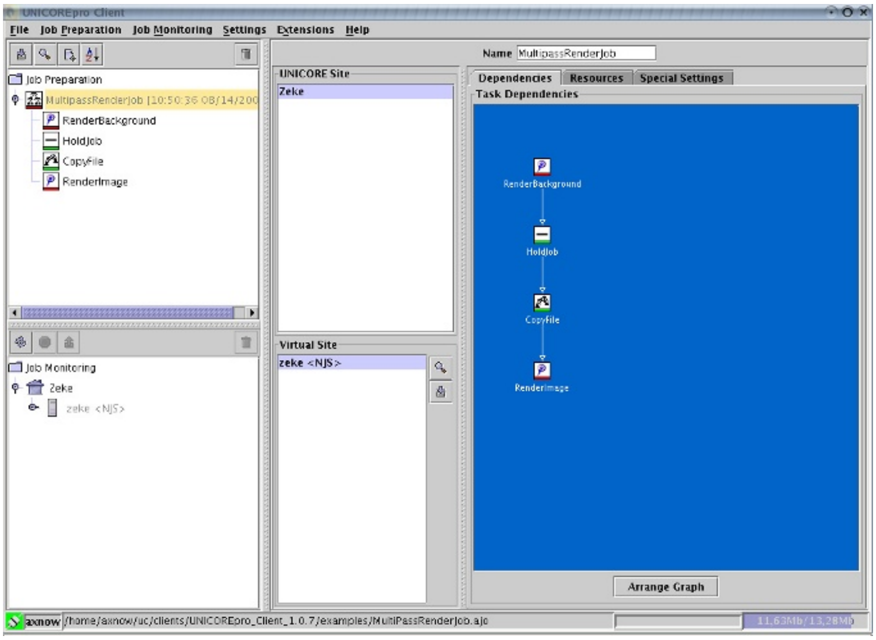
Different approach is presented by the UNICORE [2] middleware. Compared to other tools UNICORE provides access to the various types of resources, has advanced user interface, is flexible and allows for easy integration with external applications. Well-developed user interface, which can be easily extended by the user is another advantage. In particular, both generic Java visualization libraries can be used for visual analysis of job progress and specialized visual applications can be integrated with the help of plugin technology into the UNICORE platform.

2 Visualization Capabilities in the UNICORE Grid Environment

The details of the UNICORE can be found elsewhere [3] and we will summarize here its most important features and recent achievements and extensions.
UNICORE is uniform interface to the computer resources which allows users to prepare, submit and control application specific jobs and file transfers (see. Fig 2). Jobs to be run on the same system can be grouped in job-groups. The user specifies target system for job group as well as resource requirements for CPU time, number of CPUs amount of memory and disk space for job. Jobs can have complicated structure with various job subgroups which can be run at different systems. Subgroup jobs and file transfers can be ordered with user-defined dependencies. This includes conditional execution and loops with control which can be used as environment variables in the user's jobs. The user input is mapped to the target system specific commands and options by the UNICORE infrastructure.

The UNICORE client provides the plugin mechanism, which became very attractive and efficient method for integration of applications with grid middleware. Plugin is written in Java and is loaded to the UNICORE client during start, or on request. Once it is available in the client, in addition to the standard futures such as preparation of the script job, user gets access to the menu, which allows preparation application specific jobs.

Dedicated plugins can be used for a database access or postprocessing. Since plugins are written in Java, they can be easily integrated with external applications or Java applets. Number of plugins have been developed to provide users with interface to the most popular applications. The visualization has been however limited to the traditional textual GUI.



Name MultipassRenderJob

MultipassRenderJob [10:50:36 08/14/200

axnow /home/axnow/uc/clients/UNICOREpro_Client_1.0.7/examples/MultiPassRenderJob.ajo

Fig. 1. UNICORE Client

During last two years in ICM a couple of plugins for UNICORE client has been developed providing advanced visualization capabilities. Those plugins show that there is large potential area of integration visualization services into the grid technology. Plugins typically provide support for building custom application tasks, but so called ’outcome panel’ provides ability to include GUI for data analysis and visualization. Effectively it gives to user opportunity to run custom job, acquire results and visualize it in one application. We have found the opportunity of early result examination very important and comfortable to the user.

The advanced visualization capabilities were added to the UNICORE Client in the different ways. The first approach was to extend existing plugins and build in some visualization routines dedicated to the particular type of application. This method is straightforward, but in log scale is time consuming because requires redesign of the code for each changes in the application. This method was used by us to provide graphic capabilities to Gaussian plugin.

The another approach is to develop dedicated visualization plugin which can be used along with other plugins to construct jobs with complicated workflow. Such plugin can be used in various application areas and is not limited to the particular application. One should note that application specific data formats can be easily translated to the visualization suitable one with a simple command or script task which can be included in the UNICORE workflow.

In both cases the UNICORE Client must be integrated with the application, which provides visualization capabilities. Since plugins are written in Java, the easiest way to create uniform system is to use Java visualization. Although Java provides reasonable graphical extensions, we had to develop dedicated visualization system, which fulfills user requirements in the area of the molecular biology and quantum chemistry.

3 Job Progress Watching with New Tools

And how does it work in a real world? A number of UNICORE plugins have been developed in ICM during EUROGRID project [4]. Typical approach for plugin programming is represented by a Gaussian Plugin - plugin for quantum chemistry code. This plugin contains GUI for custom job preparation and a visualization panel.

Typically, a Gaussian output consists of optimized molecule geometry, some simple numerical data like energy values and a series of 3D fields containing data on electron density, potential, etc. on the rectangular grid. The visualization component allows the user to view slices of result data as color maps and isoline maps with atoms of analyzed molecule giving an opportunity to examine the results of computation in progress.

The access to data produced by a current (running) job is provided by so-called "Filter plugin". It is able to find the job directory on the target system, and then filter job output and send it back to the user allowing thus to analyze computation results in the middle of processing. This opportunity is very helpful, especially if it is known that program may loose right path of computation without general failure as it is often the case of molecular dynamics computations. Knowing intermediate results allow user to make decision about aborting or continuing computation — in general lots of impatient people find this a very nice feature. Filter plugin analyzes job output files using regular expression mechanism and retrieves interesting part of computation results. Then visualization panel can read the data and present it to the scientist in graphical form providing immediate feedback, and allowing in-time result control. At this point UNICORE does not support direct connection to the job required by advanced visualization systems (like VisIt [5]), and this kind of job connection can be done only in very limited way, but we do find it potentially large area of future development.

The final component, that is the visualization itself, can be provided by a general visualization plugin, allowing user to choose among several generic visualization techniques. The so called Vis plugin, and its descendant — Sapphire plugin provide this functionality. Any file in recognized format may be downloaded from execution space while job has been finished, and then presented in custom way in client's window. The sapphire plugin is a sample application of visualization components that can be used in a standalone visualization application. The result is fully featured 2D data viewer, allowing visualize and analyze data quickly in single user environment.

In the case of grid molecular dynamics simulations a specialized plugin using Java3D technology can be used for extensive visual analysis of the MD results. The MolDyAna plugin provides standard animation and graphing functions as well as advanced data filtering and visualization methods revealing deeper features of a molecular dynamics trajectory.

4 Generic Tools for 1D and 2D Data Visualization – The Sapphire Library

Custom components for data visualization are grouped into a special library called ”Sapphire”. It provides easily extensible and quite powerful toolset for building 2D data presentation together with an extended GUI for graph customization, allowing easy modifications for all properties of plots.

Sapphire provides all important data visualization methods in 2D environment. It includes line, bar and area plots, mesh plots, colormaps, isoline maps, vector fields and glyph plots. Some specialized features like barb plots for meteorological applications are also included. All those types of visualization can be mixed in any way on the single graph, using more then one axis for each coordinate. Java2D PostScript and off-screen image rendering capabilities ensure very high publication quality of hardcopies and image exports.

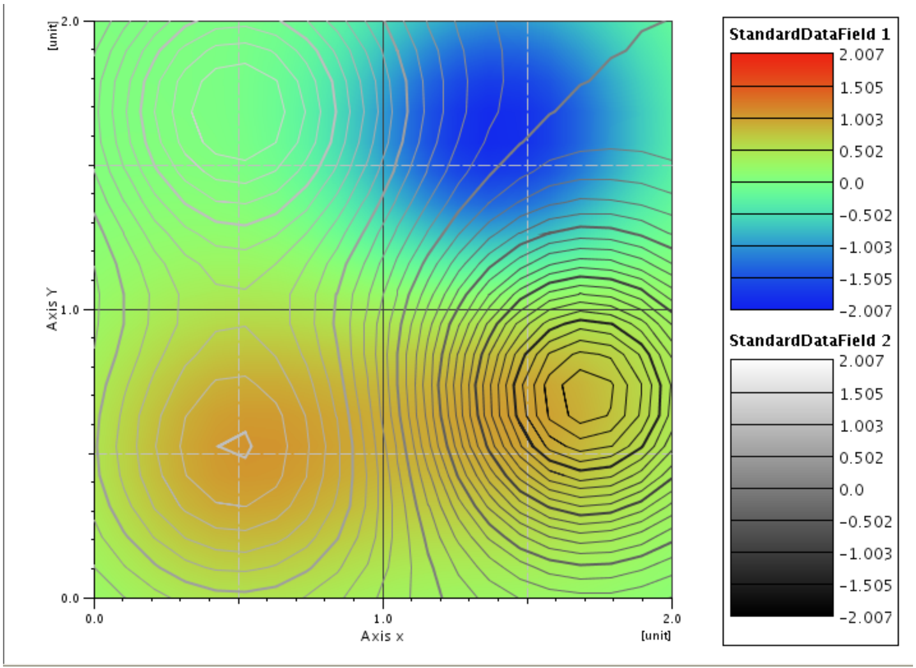


Fig. 2. Example of 2D Sapphire plot of electrostatic field in vicinity of a molecule

The Sapphire plot system is organized in an object tree that can be browsed and modified by adding new graph components (datasets, plots, axes etc.) from a library with the help of a GUI component built on top of standard Java tree GUI component.

Each graph component has its own set of properties controllable by 'property sheets', that are feature of java basically intended to be used in IDE packages. Property sheet for object shows object properties values and allows editing them in customizable way. For example, property editor for line renderer (object responsible for rendering data as a poly line) allows user to modify both line properties (color, thickness, style etc.) and rendering method (standard polyline or staircase). What is important in this method, all newly added components do not require to provide custom GUI. For a programmer wishing to add own components into library it is enough to follow java code guidelines and mark through special method, which object properties shall be available for user.

5 Molecular Dynamics Analyzer (MolDyAna) and Its Use in Examination of MD Trajectories

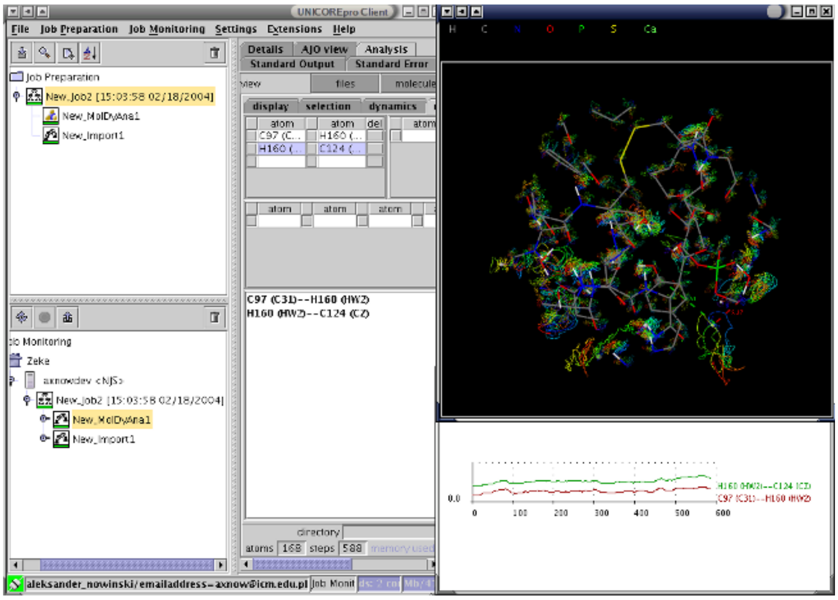
MolDyAna is a toolkit for interactive graphical analysis of the molecular dynamics trajectories build as a Java standalone application or as UNICORE plugin with an extensive use of Java3D technology. It can accept ASCII trajectory files in raw .xyz format and in formats used by several standard MD packages (GROMOS, AMBER etc.).

MolDyAna can display molecule geometry using standard methods of presentations (lines, sticks, ball-and-stick and space fill modes). Different parts of the system can be assigned to up to 4 classes with each class shown in its own mode (e.g. reacting atoms as large spheres in space fill mode, active center displayed as ball-and stick with its neighborhood shown as sticks and the rest of the system with lines only presentation. Tube/ribbon plot of protein backbone will also be available. MolDyAna allows the user to browse trajectory frames or display animation of the trajectory.

The user can pick interatomic distances, angles and dihedral angles and analyze the graphs of these measurements versus time. An arbitrary number of such graphs can be plotted.

MolDyAna provides in addition several non-standard interactive visual analysis techniques that can be used to obtain better insight into the MD trajectory.

The user can pick an atom or a set of atoms and redisplay the paths of these atoms in the whole simulated time. Moments of particularly fast movements of the atoms or correlated movements in various parts of the system can be pinpointed much easier this way than by the analysis of distance/angle/dihedral graphs versus time. The problem of identification of the time axis has been solved by using a colormap to encode the time, plotting the atom positions at the beginning of simulation in blue and corresponding position at the end of simulation in red with interpolated color values in between.


UNICOREpro Client
File Job Preparation Job Monitoring Settings Extensions Help
Details AJO view Analysis
Standard Output Standard Error
view files molecule
display selection dynamics
Job Preparation
New_Job2 [15:03:58 02/18/2004]
New_MolDyAna1
New_Import1
Job Monitoring
Zeke
New_Job2 [15:03:58 02/18/2004]
New_MolDyAna1
New_Import1
H C N O P S Ca
H160 (HW2)--C124 (CZ)
C97 (C31)--H160 (HW2)
0.0
0 100 200 300 400 500 600
directory
atoms 168 steps 588
aleksander_nowinski/emailaddress=axnow@icm.edu.pl

C97 (C31)--H160 (HW2)
H160 (HW2)--C124 (CZ)

Fig. 3. MolDyAna plugin example

In addition to standard plots of distances or angles versus time MolDyAna offers an abstract 3D plot of three user chosen measurements again color coded with the simulated time. The complicated 3D trajectory can often reveal e.g. correlations between various geometric features and changes of such correlations in time. For example, a correlation emerging at some moment may indicate creation of some sort of a bond between observed subsystems.

Fast vibrations of relatively rigid subsystems or movement of low inertia fragments often visually dominate MD trajectories . To hide such fast modes and to present some generalized picture of a MD trajectory, MolDyAna includes a possibility of Sawitzky-Golay filtering [6], [7].

In order to obtain for a function $f(t)$ its smoothed approximation value $g(t_0)$ one computes best approximation of $f(t)$ on the interval $< t_0 - \Delta t, t_0 + \Delta t >$ by a quadratic polynomial $p(t) = a_{t_0}(t - t_0)^2 + b_{t_0}(t - t_0) + c_{t_0}$ and takes $g(t_0) = p(t0) = c_{t_0}$. (MolDyAna computes a Gaussian weighted approximation here). In addition to good quality of trajectory smoothing without excessive deformation of molecule geometry one obtains physically meaningful values of b_{t_0} (average velocity component) and a_{t_0} (average acceleration component) that can be used for further visual analysis.

The fast vibrational models are neglected by the smoothing procedure but are still present in the picture: the statistics of difference $f(t)_g(t)$ on $< t_0 - \Delta t, t_0 + \Delta t >$ representing the neglected vibrations is visualized as principal moments ellipsoid.

The MolDyAna panel can be used both in a stand alone application and (with a suitable wrapper) in an UNICORE plugin. In the later case, some two-way communication between a grid MD job and MolDyAna is provided. To avoid excessive transfer of MD trajectories that can easily reach hundreds of gigabytes the user can start a MolDyAna plugin session by selecting only interesting parts of the trajectory (e.g. given time interval, reaction site atoms or protein backbone only) and generate a suitable script for the filter plugin. Selected parts of the trajectory file will be then transferred to MolDyAna for visual analysis.

6 Conclusions

The UNICORE software was used as framework providing uniform access to the number of resources and applications important for life sciences such as computational chemistry and molecular biology. This includes seamless access to the computational resources. The UNICORE is flexible framework for application specific interfaces development. Examples presented here demonstrates how visualization capabilities can be included in the UNICORE middleware. In results user obtains uniform and flexible environment for scientific simulations. Developed extensions allows to use grid middleware in the scientific discovery process which require significant visualization capabilities both during pre- and post-processing. Modular architecture based on the plugin concept extended now with visualization plugins opens number of possible applications in the different areas.

One should note, that UNICORE middleware can be used also together with other grid middleware, especially it can be used as job preparation, submission and control tool for Globus.

Acknowledgements. This work is supported by European Commission under IST grants EUROGRID (no. 20247) and GRIP (no. 32257).

References

1. C. Kesselman I. Foster, editor. *The Grid: Blueprint for a Future Computing Infrastructure.* Morgan Kaufman Publishers, USA, 1999.
2. UNICORE. Unicore Forum. http://www.unicore.org.
3. P. Bała, B. Lesyng and D. Erwin EUROGRID - European Computational Grid Testbed. *J. Parallel and Distrib. Comp.* 63(5):590–596, 2003
4. J. Pytliński, Ł. Skorwider, P. Bała, M. Nazaruk, V. Alessandrini, D. Girou, G. Grasseau, D. Erwin, D. Mallmann, J. MacLaren, J. Brooke. BioGRID - An European grid for molecular biology. *Proceedings 11th IEEE International Symposium on Distributed Computig*, IEEE Comput. Soc. 2002, p. 412
5. Eickermann, Thomas; Frings, Wolfgang VISIT - a Visualization Interface Toolkit Version 1.0 Interner Bericht FZJ-ZAM-IB-2000-16, Dezember 2000
6. Savitzky A., and Golay, M.J.E. 1964, Analytical Chemistry, vol. 36, pp. 1627–1639.
7. Hamming, R.W. 1983, Digital Filters, 2nd ed. (Englewood Cliffs, NJ: Prentice-Hall).