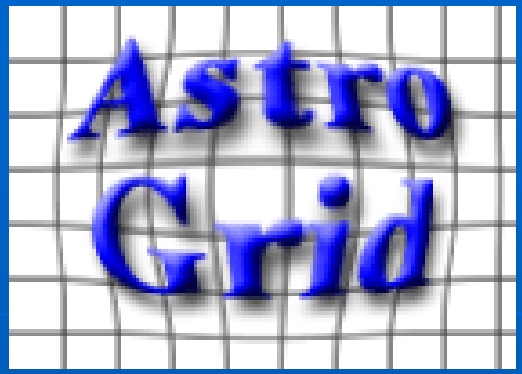


VisIVO: Interoperability Towards Grid

INAF–Catania Astrophysical Observatory

A. Costa[1], M. Comparato[1], U. Becciani[1]

C. Gheller[2]

[1] INAF – Catania Astrophysical Observatory

[2] CINECA - Italy

VisIVO

Visualization Interface for the Virtual Observatory

VisIVO is a visualization package developed in collaboration between INAF (Catania Astrophysical Observatory) and CINECA (the largest Italian academic supercomputing center) with the specific object of supporting visualization and analysis of astrophysical data. The package is written in C++ .

VisIVO: Italian Contribution to the VO-TECH – DS6: Data Exploration (Data Mining + Visualization)

The **VOTech** project aims at completing all technical preparatory work necessary for the construction of the European Virtual Observatory (Euro-VO).

VisIVO

Visualization Interface for the Virtual Observatory

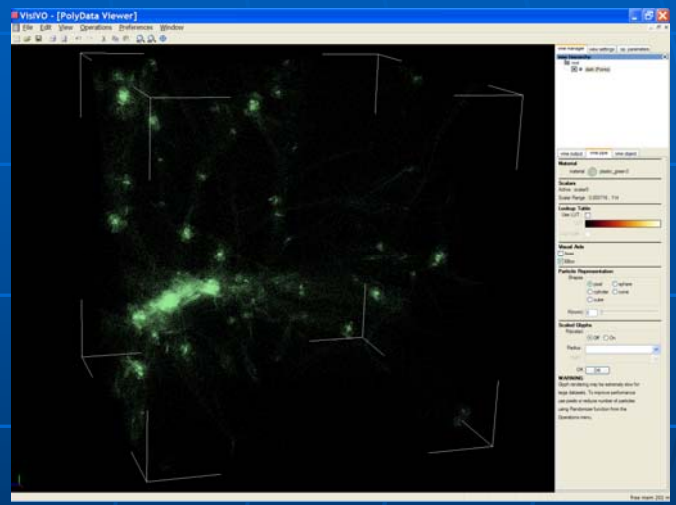
VisIVO can be used both as a stand-alone application, that acts on local files, and as an interface to the Virtual Observatory framework, from which it can retrieve the data.
NEW 0.9 release is now available:

http://visivo.cineca.it/

http://visivo.oact.inaf.it/

FLY

- FLY is a parallel tree code that runs, with a very high resolution, N-Body simulations of the Large Scale Structure of the Universe. It runs on multi processors Unix-like systems where a full implementation of the MPI-2 communication standard is available and on IBM SP systems using LAPI. The code is designed to minimize data replication among processors and to maximize the number of bodies which can be used.



FLY has been developed by:

U. Becciani, V.Antonuccio-Delogu

(INAF Astrophysical Observatory of Catania)

http://www.oact.inaf.it/fly/

VisIVO - Astrogrid - FLY

Astrogrid is used to run a Fly job in grid environment managing its input and output files

The testing platform is the EPCC high performance computing system

- Running Fly through Astrogrid :Problems
  - In the Astrogrid env. FLY cannot be executed multiple times from the same user
  - It was not designed as a command line application.
  - FLY must be recompiled each time you change some simulation parameters. (number of bodies, processors)

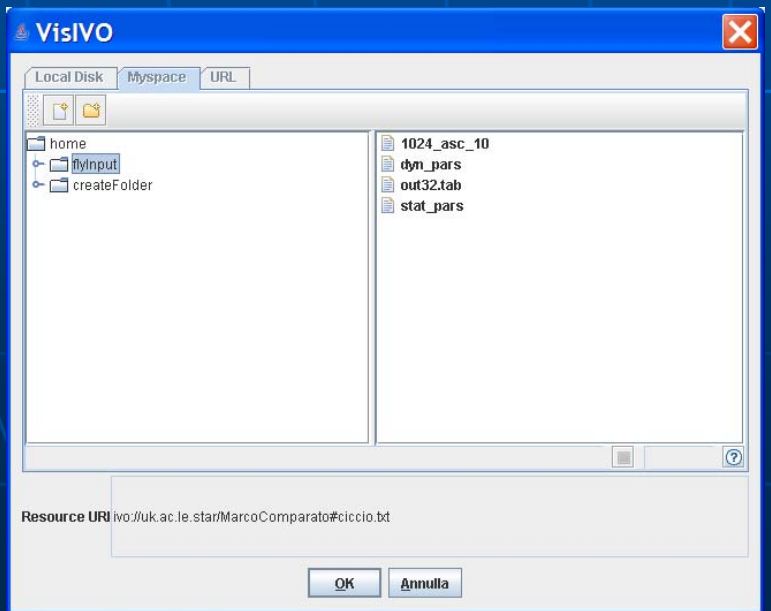
VisIVO - Astrogrid - FLY

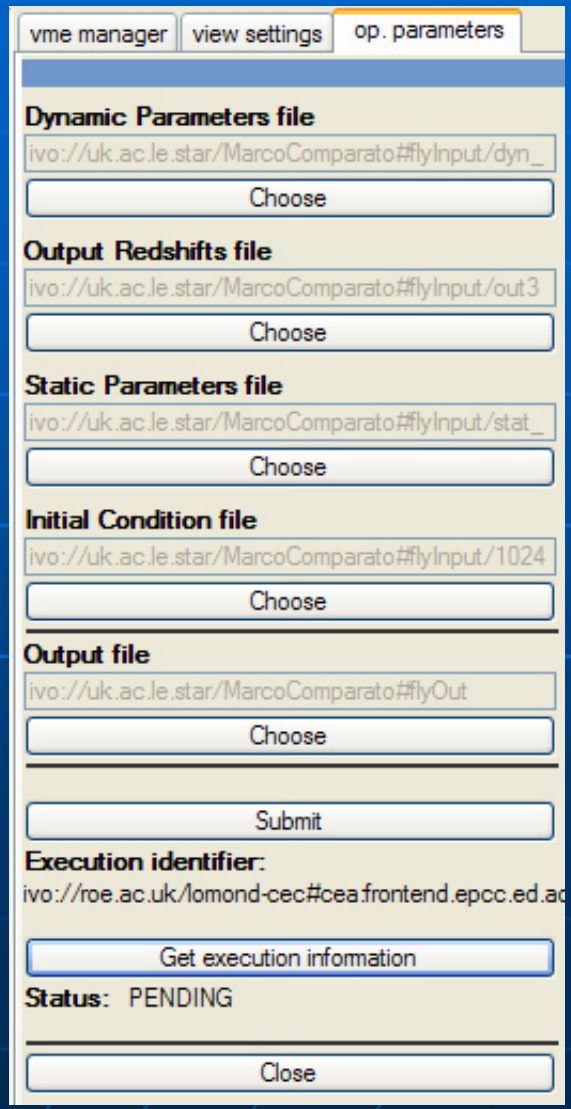
- Solution:
  - We have developed a wrapper shell script that makes possible to run simultaneously multiple instances of FLY and makes transparent to the end user the building processes needed after changes to the simulation parameters.
  - We have wrapped the FLY shell script with a Common Execution Connector to connect it to the AstroGrid infrastructure and registered it to make it visible to the Virtual Observatory

    FLY is now visible to AstroGrid's web portal, desktop UI and workflow system

VisIVO - Astrogrid - FLY

Implemented an interface in VisIVO to make FLY runnable from our visualization software, with the ability to select files stored in MySpace as input and output of the simulation.

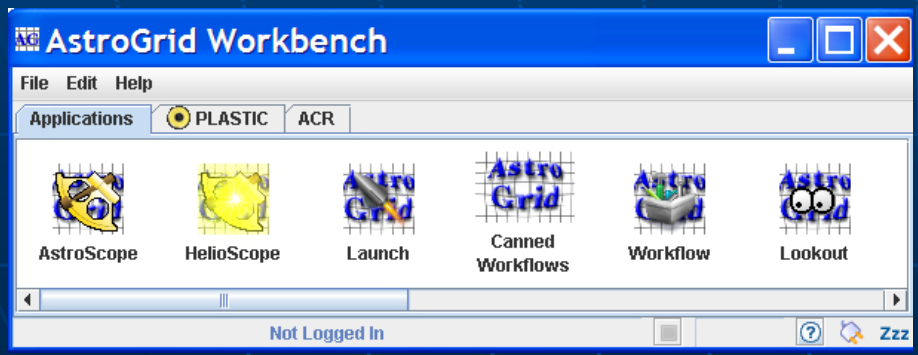


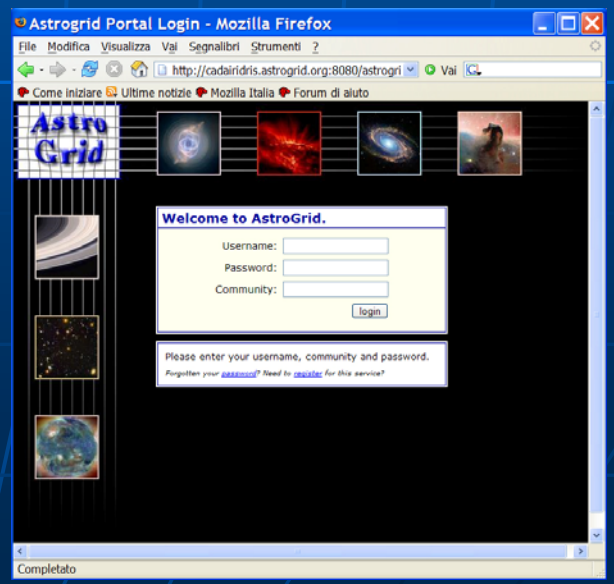


Interaction

FLY

ACR





Lessons learned

- CEA makes it easy to publish algorithms on HPC resources to the grid
  - Setting up the CEA server takes an hour or two
  - Hard part is writing the wrapper script that the CEA server invokes
- Invoking remote CEA applications from clients is straightforward using the AR
  - Using the AR from Java programs is trivial
  - More involved from C++, but do-able

Grid Experiences within INFN Production Grid (EGEE compliant)

The interface to the grid

- Apache Tomcat 5.5.9
- Axis 1.2
- JSSE (Java Secure Socket Extension)
- GILDA User Interface Plug & Play

(fully compatible with LCG 2.4.0 and gLite 1.1)

HOP Group finder

HOP It is an algorithm for finding groups of particles based on the one developed and coded by Daniel Eisenstein & Piet Hut, Institute for Advanced Study, Princeton, NJ

HOP was distributed as RPM Package and installed in the Worker Nodes Elements

http://cmb.as.arizona.edu/~eisenste/hop/hop.html

Upload your credentials

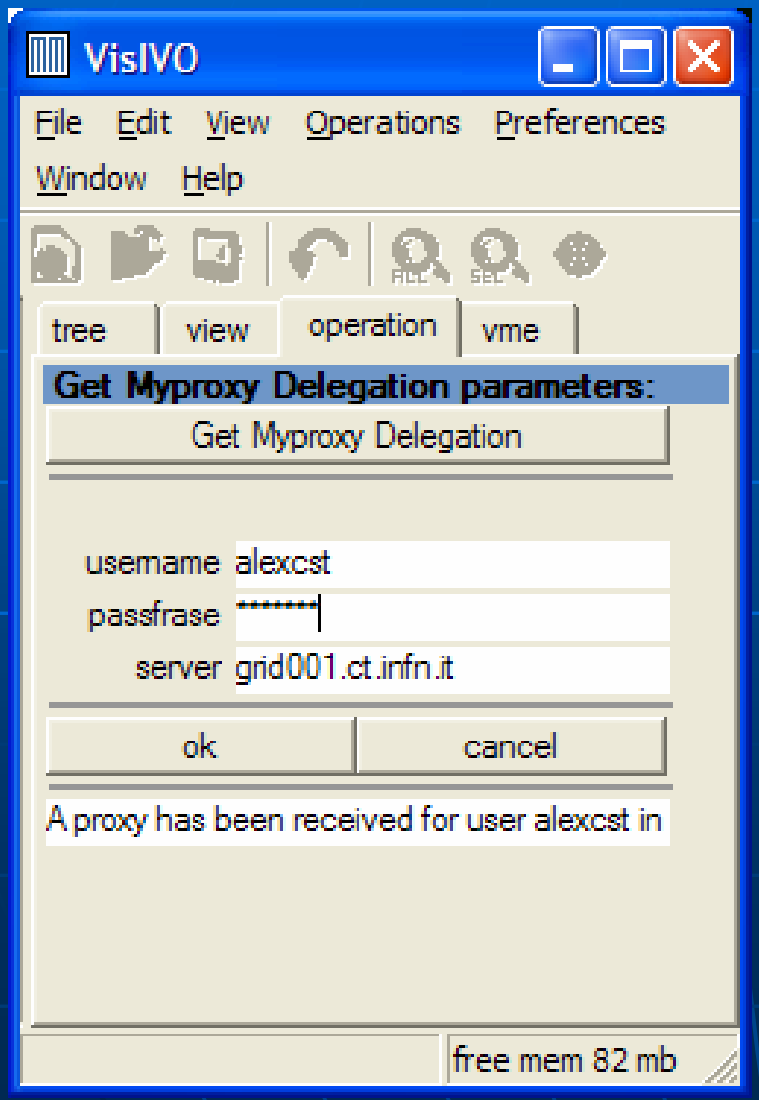
First of all you need a digital X509 certificate issued by INFN Certification Authority, or most simply a digital certificate issued by GILDA (Grid Infn Laboratory for Dissemination Activities); you can follow the link https://gilda.ct.infn.it/users.html for the instructions to request and obtain a digital certificate and a User Interface account.

Then you have to upload credentials to a myproxy-server from a User Interface everywhere in the world.

The command line is :

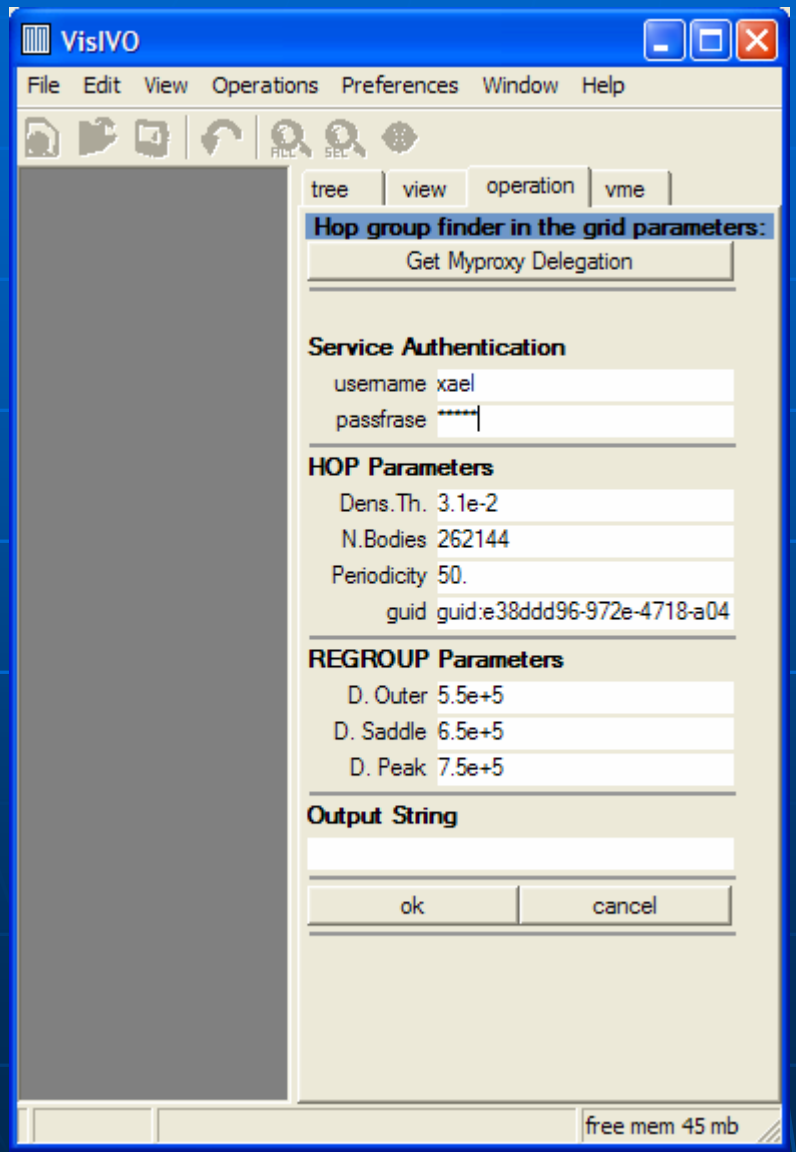
$ myproxy-init -s “myproxy server” -l “myproxy username” (you have to provide a myproxy passphrase).

Get Myproxy Delegation



- The OK button will invoke the web service and will allow you to retrieve credentials from the myproxy opening your session in the grid environment.

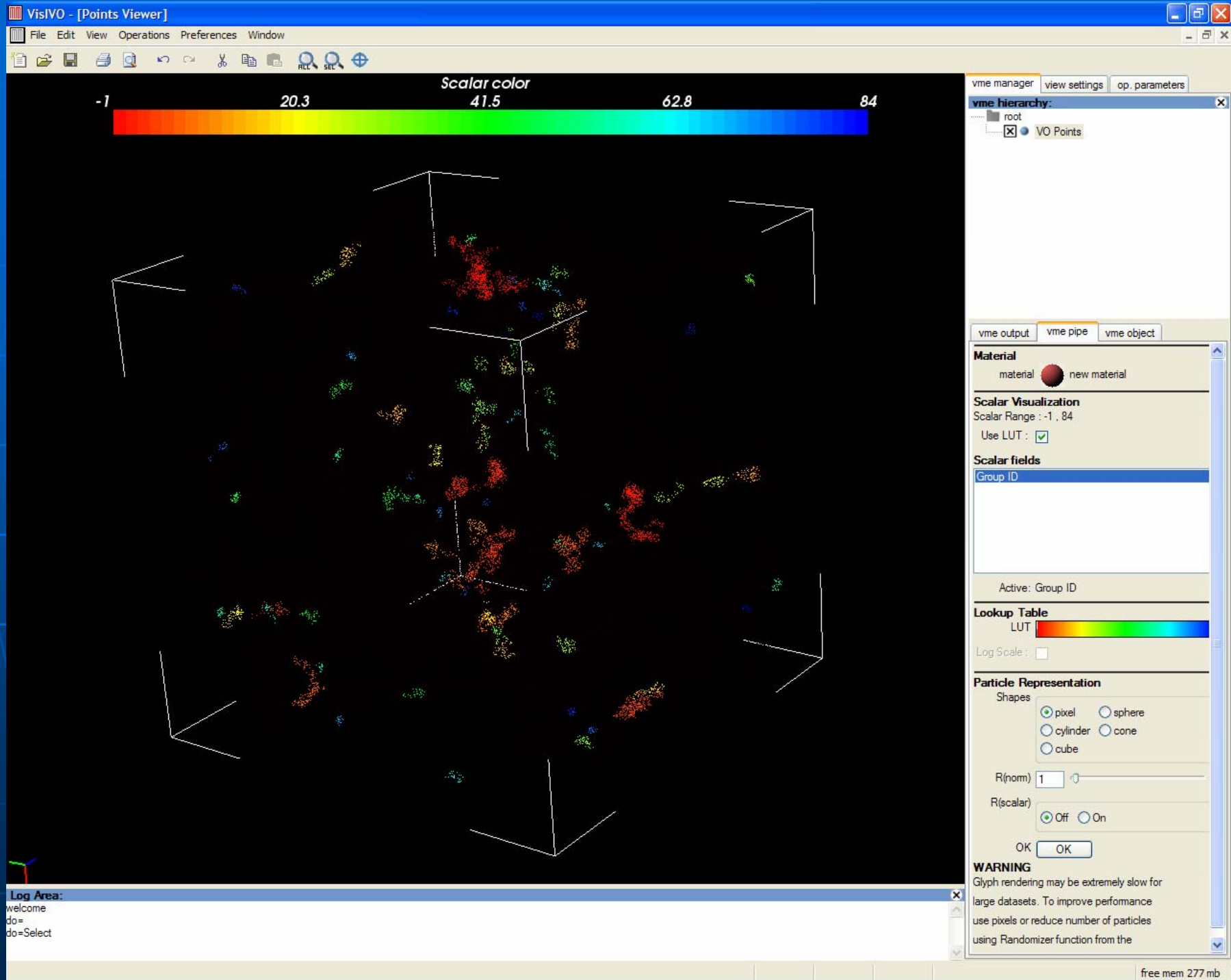
HOP-Regroup



- The OK button will invoke the web service and will run a HOP-REGROUP session:
- the web server will create the ad hoc jdl file reporting all the user specifications: “guid” and

Output Results as VoTable

- Point Coordinates
- Density values (from Hop Algorithm)
- Group ID


VisIVO - [Points Viewer]
File Edit View Operations Preferences Window
Scalar color
-1
20.3
41.5
62.8
84
vme manager
view settings
op. parameters
vme hierarchy:
root
VO Points
vme output
vme pipe
vme object
Material
material
new material
Scalar Visualization
Scalar Range : -1 , 84
Use LUT :
Scalar fields
Group ID
Active: Group ID
Lookup Table
LUT
Log Scale :
Particle Representation
Shapes
pixel
sphere
cylinder
cone
cube
R(nom)
1
R(scalar)
Off
On
OK
OK
WARNING
Glyph rendering may be extremely slow for
large datasets. To improve performance
use pixels or reduce number of particles
using Randomizer function from the
Log Area:
welcome
do=
do=Select
free mem 277 mb