BASIN

Beowulf Analysis Symbolic INterface

A multi-user environment for parallel data analysis and visualization

Enrico Vesperini

Doug Jones

Dave Goldberg

Steve McMillan (PI)

and the BASIN team

Department of Physics

Drexel University

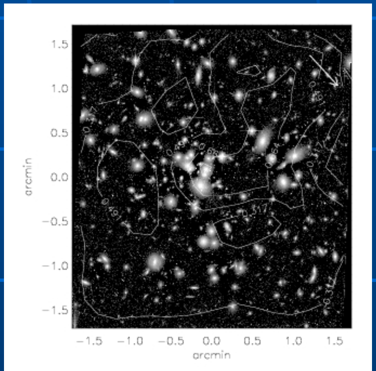
SciPy2007

August 16,2007

Who we are…

Theoretical and observational astrophysicists working with large (10^2-10^3 GB) datasets from simulations and observations:

- stellar dynamics, N-body simulations of dense stellar systems, globular star clusters;
- gravitational lensing, dark matter distribution
- observational extra-galactic astronomy, Sloan Digital Sky Survey-SDSS.



What we use:
Beowulf Clusters

- Beowulf clusters: attractive (low-cost) parallel systems
- Several packages exist for serial data analysis (e.g., in observational astronomy, IRAF, MIDAS).
- For parallel data analysis each group tends to (re-) develop its own set of tools:
  - Expertise in parallel tools/algorithms.
  - Large time investment.

What is it?

- Integrated suite of tools for parallel data analysis and visualization.
- Multi-user environment for interactive data analysis and visualization.

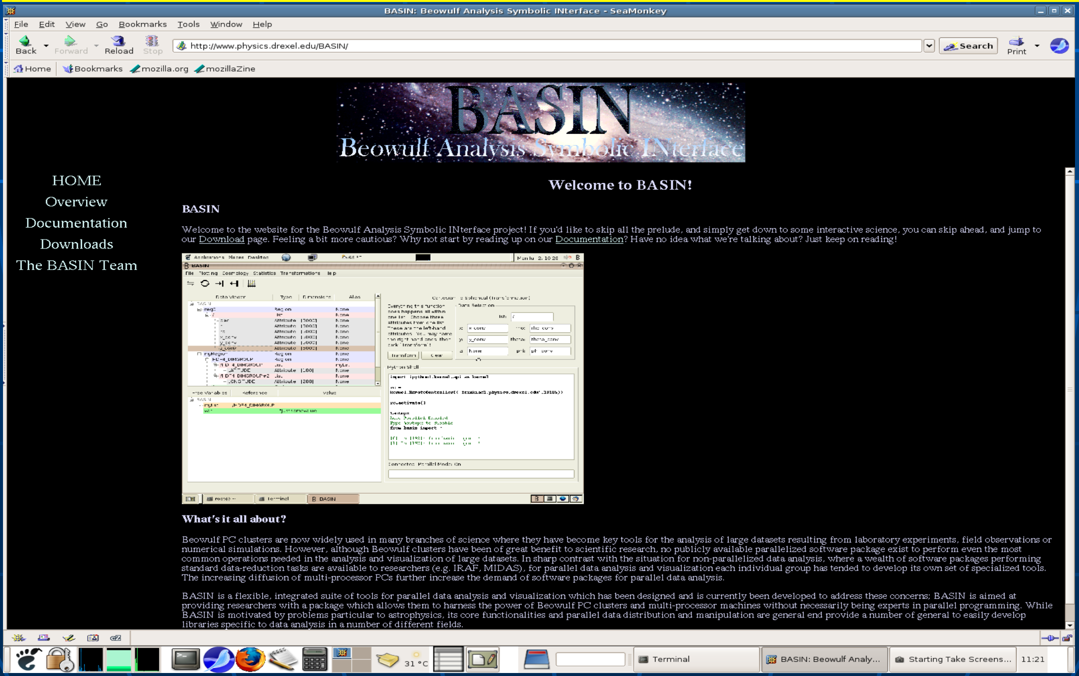
Why?

- Easy and transparent parallel data analysis.
- Avoid redundant development of functions commonly used in data analysis.

Where?

BASIN is freely (GPL) available at

http://www.physics.drexel.edu/BASIN



Welcome to BASIN!

BASIN

Welcome to the website for the Beowulf Analysis Symbolic INterface project! If you'd like to skip all the prelude, and simply get down to some interactive science, you can skip ahead, and jump to our Download page. Feeling a bit more cautious? Why not start by reading up on our Documentation? Have no idea what we're talking about? Just keep on reading!

What's it all about?

Beowulf PC clusters are now widely used in many branches of science where they have become key tools for the analysis of large datasets resulting from laboratory experiments, field observations or numerical simulations. However, although Beowulf clusters have been of great benefit to scientific research, no publicly available parallelized software package exist to perform even the most common operations needed in the analysis and visualization of large datasets. In sharp contrast with the situation for non-parallelized data analysis, where a wealth of software packages performing standard data-reduction tasks are available to researchers (e.g. IRAF, MIDAS), for parallel data analysis and visualization each individual group has tended to develop its own set of specialized tools. The increasing diffusion of multi-processor PCs further increase the demand of software packages for parallel data analysis.

BASIN is a flexible, integrated suite of tools for parallel data analysis and visualization which has been designed and is currently been developed to address these concerns; BASIN is aimed at providing researchers with a package which allows them to harness the power of Beowulf PC clusters and multi-processor machines without necessarily being experts in parallel programming. While BASIN is motivated by problems particular to astrophysics, its core functionalities and parallel data distribution and manipulation are general end provide a number of general to easily develop libraries specific to data analysis in a number of different fields.

BASIN DEMO

1) Start the computational engine on a remote parallel server.

2) Connect a local client to the computational engine.

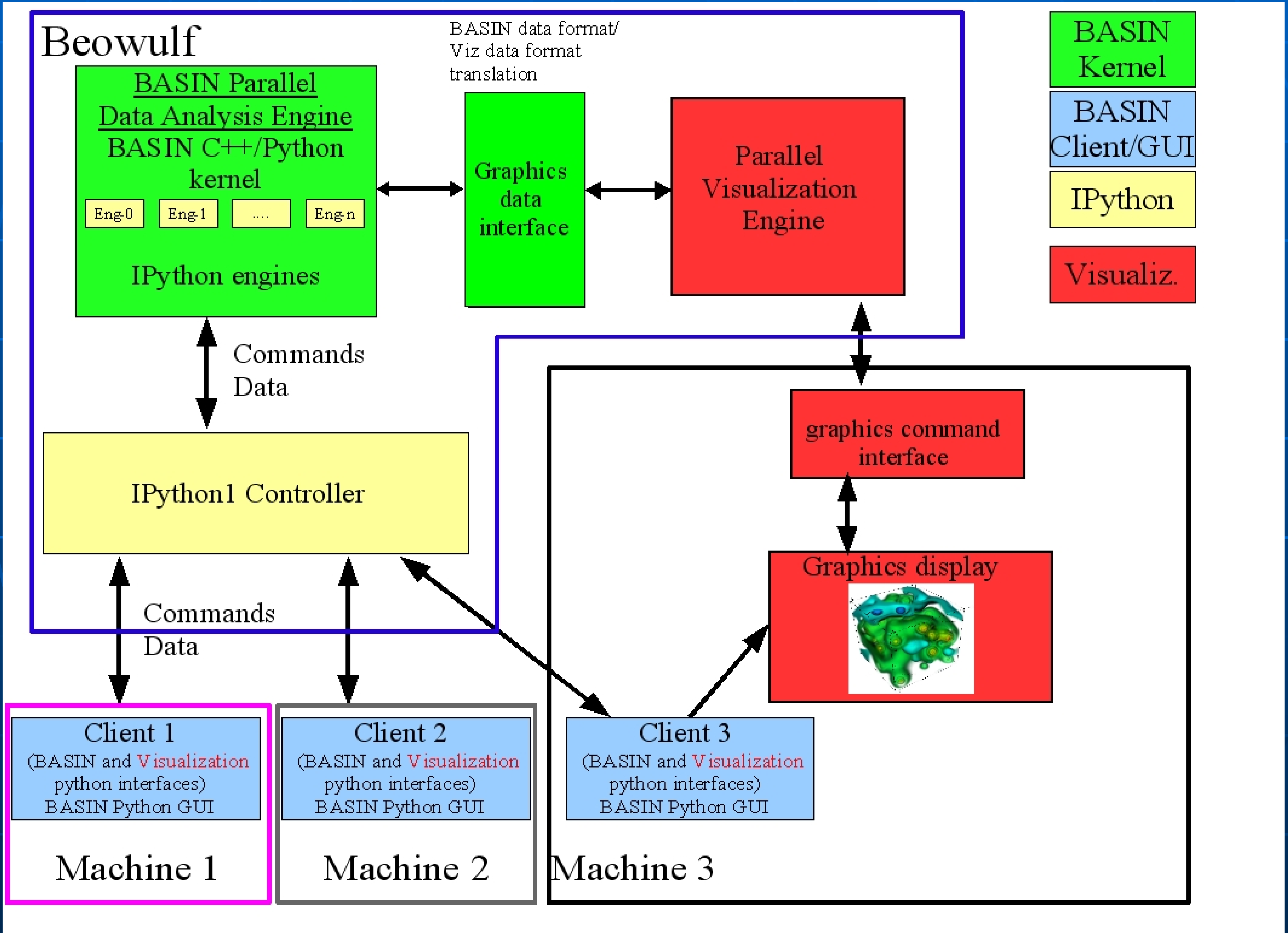
3) Read file distribute data.

4) Parallel calculation of new attributes.

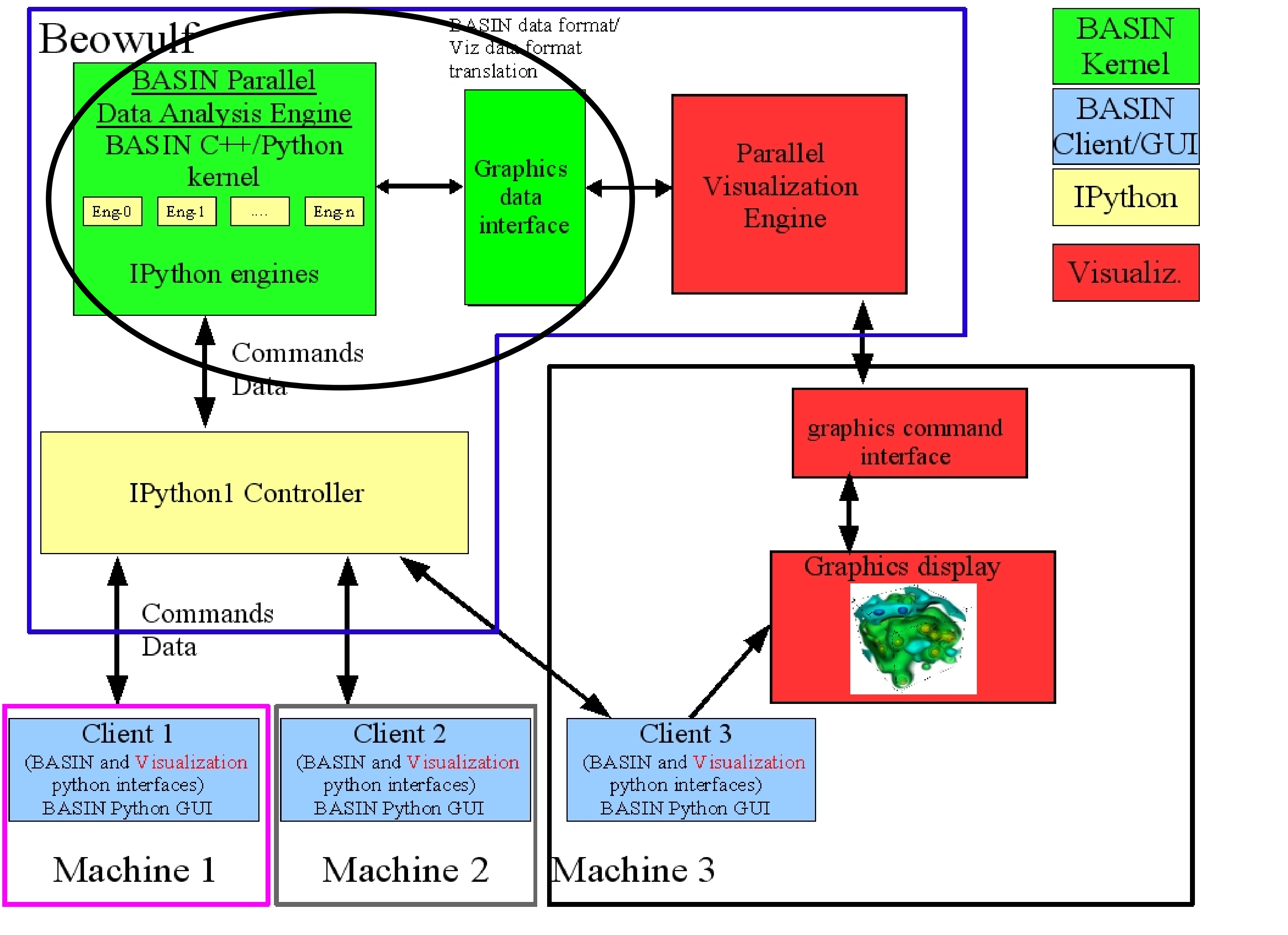
5) Visualize distributed data (using VisIt-LLNL)

6) Transfer data on the local machine and plot with one of the standard Python plotting packages (Matplotlib).

BASIN architecture



BASIN architecture



Beowulf

BASIN Parallel Data Analysis Engine

BASIN C++/Python kernel

Eng-0 Eng-1 .... Eng-n

IPython engines

BASIN data format/ Viz data format translation

Graphics data interface

Parallel Visualization Engine

Commands Data

IPython1 Controller

Commands Data

graphics command interface

Graphics display

Client 1 (BASIN and Visualization python interfaces) BASIN Python GUI

Machine 1

Client 2 (BASIN and Visualization python interfaces) BASIN Python GUI

Machine 2

Client 3 (BASIN and Visualization python interfaces) BASIN Python GUI

Machine 3

🟩 BASIN Kernel

🟦 BASIN Client/GUI

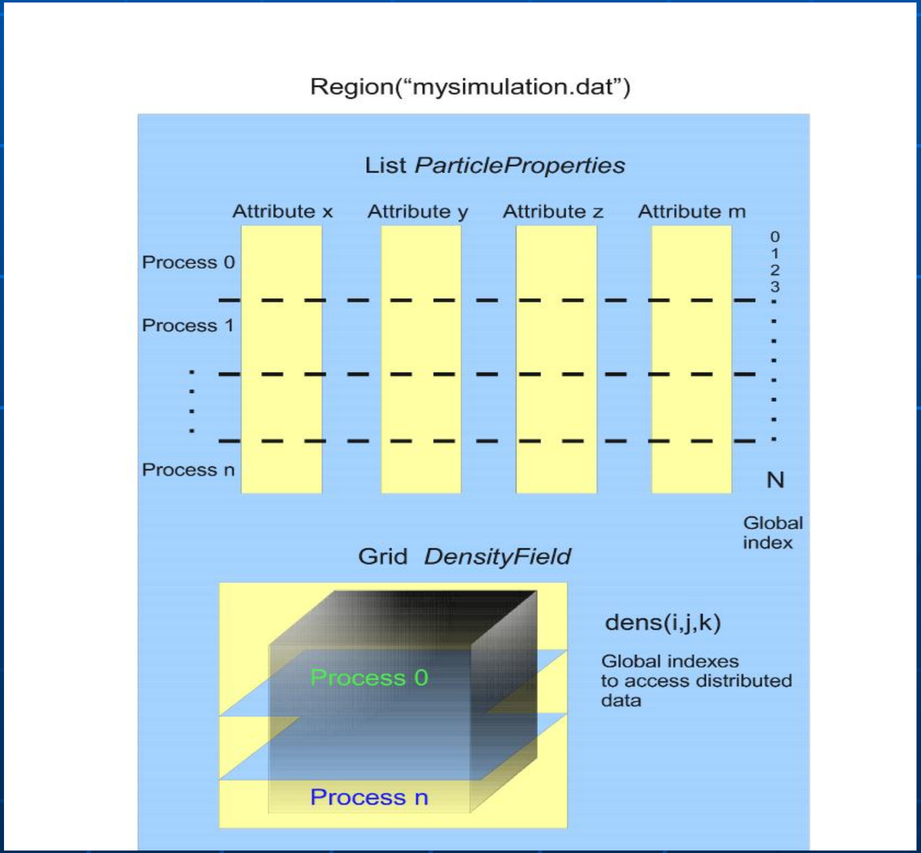
🟨 IPython

🟥 Visualiz.

Data Analysis Engine

BASIN kernel : classes and functions for data distribution and parallel data operations.

(C++/MPI)



Objects visible to the user:

- ***Region*** defined by the data file read
- ***Data*** :
  - ***Grid***
  - ***List***
- ***Attribute***

Data distribution and parallel operations

Attribute class

- Hide the complexity of data distribution, retrieval and parallel operations.
- Tools to ease the parallel data distribution and management on a distributed memory machine.
- Shared-memory view of data in a distributed memory machine.
- Arbitrary n-dimensional arrays of primitive datatypes or user-defined structures and objects.
- Parallel math (elementwise and reduction) and logical operations on distributed data.
- Each process can locate its own block of data.

Data distribution and parallel operations

Attribute class

- Hide the complexity of data distribution, retrieval and parallel operations.
- Tools to ease the parallel data distribution and management on a distributed memory machine.
- Shared-memory view of data in a distributed memory machine.
- Arbitrary n-dimensional arrays of primitive datatypes or user-defined structures and objects.
- Parallel math (elementwise and reduction) and logical operations on distributed data.
- Each process can locate its own block of data.

Data distribution and parallel operations

Attribute class

- Hide the complexity of data distribution, retrieval and parallel operations.
- Tools to ease the parallel data distribution and management on a distributed memory machine.
- Shared-memory view of data in a distributed memory machine.
- Arbitrary n-dimensional arrays of primitive datatypes or user-defined structures and objects.
- Parallel math (elementwise and reduction) and logical operations on distributed data.
- Each process can locate its own block of data.

Data distribution and parallel operations

Attribute class

- Hide the complexity of data distribution, retrieval and parallel operations.
- Tools to ease the parallel data distribution and management on a distributed memory machine.
- Shared-memory view of data in a distributed memory machine.
- Arbitrary n-dimensional arrays of primitive datatypes or user-defined structures and objects.
- Parallel math (elementwise and reduction) and logical operations on distributed data.
- Each process can locate its own block of data.

Data distribution and parallel operations

Attribute class

- Hide the complexity of data distribution, retrieval and parallel operations.
- Tools to ease the parallel data distribution and management on a distributed memory machine.
- Shared-memory view of data in a distributed memory machine.
- Arbitrary n-dimensional arrays of primitive datatypes or user-defined structures and objects.
- Parallel math (elementwise and reduction) and logical operations on distributed data.
- Each process can locate its own block of data.

Data distribution and parallel operations

Attribute class

- Hide the complexity of data distribution, retrieval and parallel operations.
- Tools to ease the parallel data distribution and management on a distributed memory machine.
- Shared-memory view of data in a distributed memory machine.
- Arbitrary n-dimensional arrays of primitive datatypes or user-defined structures and objects.
- Parallel math (elementwise and reduction) and logical operations on distributed data.
- Each process can locate its own block of data.

Data distribution and parallel operations

Attribute class

```
reg0 = Region ("/home/vesperin/starclust.dat")
m = reg0.get_attribute ("mass")
x = reg0.get_attribute ("x")
y = reg0.get_attribute ("y")
z = reg0.get_attribute ("z")
print m[10]
m[120]=12.3
total_Mass=sum(m)
logm=log10(m)
Hm=where(m>5,m,0)
logr=log10(sqrt(x*x+y*y+z*z))
```

Center of Mass

- C/ MPI

```
……m, x…..


MPI_Scatter(&m[0], n_loc, MPI_DOUBLE,
 &m_loc[0],n_loc,MPI_DOUBLE, 0,
MPI_COMM_WORLD);
MPI_Scatter(&m[0], n_loc, MPI_DOUBLE,
 &m_loc[0],n_loc,MPI_DOUBLE, 0,
MPI_COMM_WORLD);



for(int i = 0; i < n_loc; i++)
{
x_sum_loc += x_loc[i] * m_loc[i];
m_sum_loc += m_loc[i];
}

MPI_Reduce(&x_sum_loc, &x_sum, 1,
MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);
MPI_Reduce(&m_sum_loc, &m_sum, 1,
MPI_DOUBLE, MPI_SUM, 0,MPI_COMM_WORLD);

    if(myRank == 0) {
      com[0] = x_sum / m_sum;
      }
```

- BASIN(C++/Python)

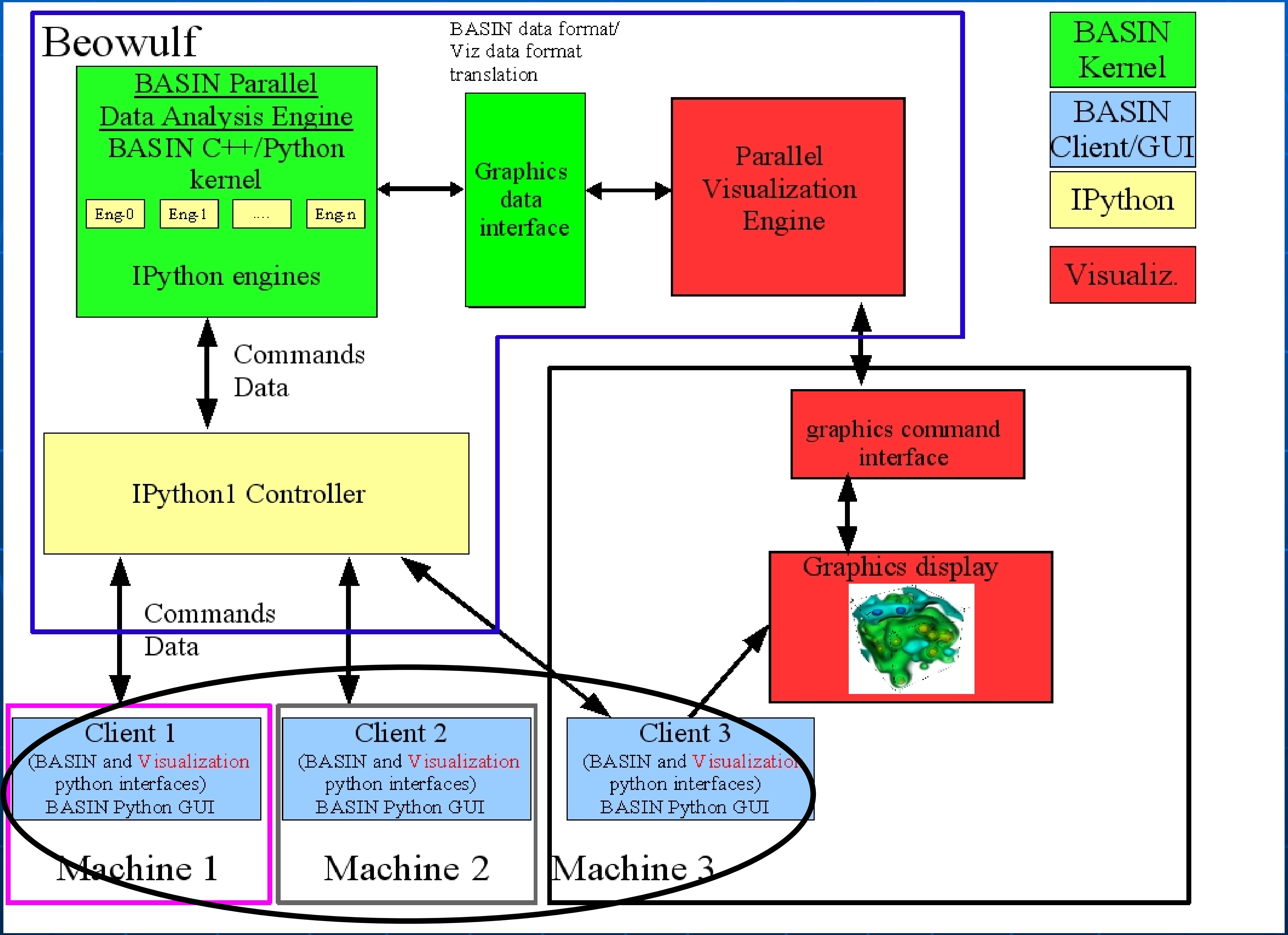
```
com=sum(m*x)/sum(m)
```

Data Analysis Engine

BASIN kernel : Scientific Packages

- *Cosmology*
- *Stellar dynamics*
- *Statistics*
- *FFT* (FFTW, http://www.fftw.org/)
- *Coordinate transformations*

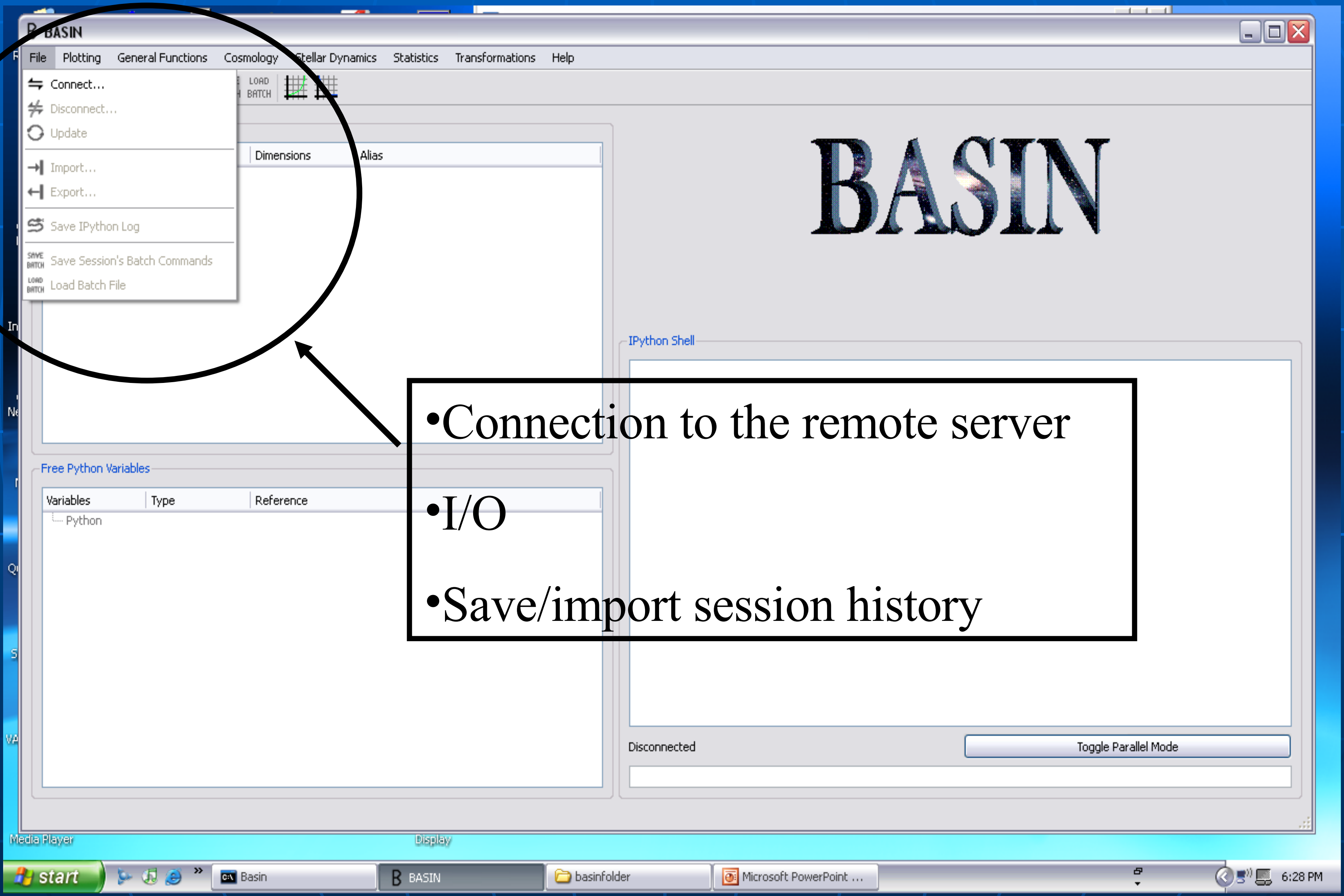
BASIN architecture


Beowulf
BASIN Parallel Data Analysis Engine
BASIN C++/Python kernel
Eng-0
Eng-1
....
Eng-n
IPython engines
BASIN data format/ Viz data format translation
Graphics data interface
Parallel Visualization Engine
Commands Data
IPython1 Controller
Commands Data
graphics command interface
Graphics display
Client 1 (BASIN and Visualization python interfaces) BASIN Python GUI
Client 2 (BASIN and Visualization python interfaces) BASIN Python GUI
Client 3 (BASIN and Visualization python interfaces) BASIN Python GUI
Machine 1
Machine 2
Machine 3
BASIN Kernel
BASIN Client/GUI
IPython
Visualiz.

BASIN Python User Interface and GUI

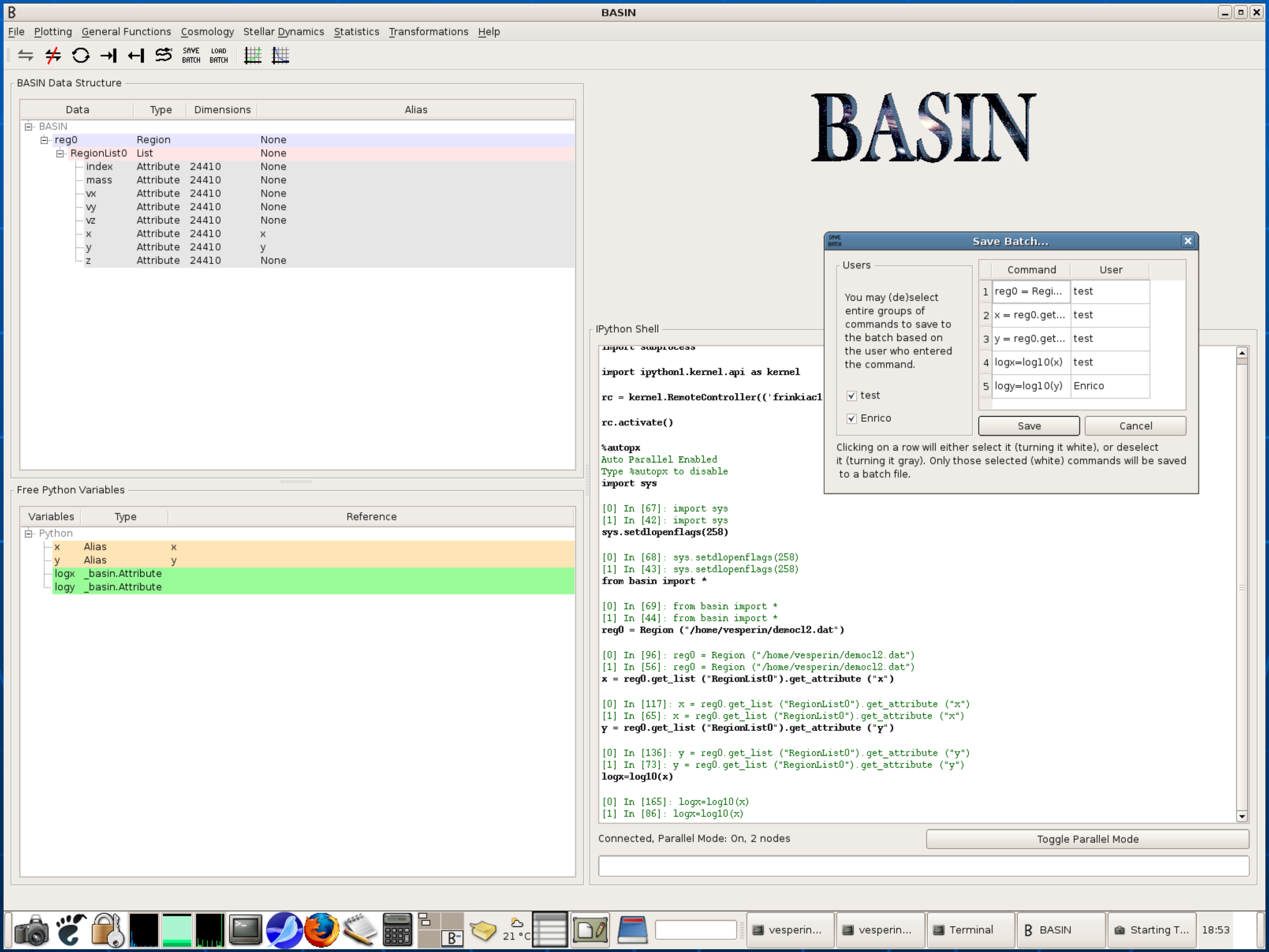
- BASIN Python interface created with Boost Python.
- A remote Python client can invoke BASIN commands to be executed by the Data Analysis Engine.
- Multiple distributed clients can connect to the same BASIN Data Analysis Engine and share the same data (*`IPython/IPython1`* F.Perez, B.Granger `http://ipython.scipy.org/moin/IPython1`)

BASIN Python User Interface and GUI



- Connection to the remote server
- I/O
- Save/import session history

BASIN Python User Interface and GUI



BASIN Data Structure

Data	Type	Dimensions	Alias
BASIN			
reg0	Region		None
RegionList0	List		None
index	Attribute	24410	None
mass	Attribute	24410	None
vx	Attribute	24410	None
vy	Attribute	24410	None
vz	Attribute	24410	None
x	Attribute	24410	x
y	Attribute	24410	y
z	Attribute	24410	None

Free Python Variables

Variables	Type	Reference
Python		
x	Alias	x
y	Alias	y
logx	_basin.Attribute	
logy	_basin.Attribute	

IPython Shell

```
import subprocess

import ipython1.kernel.api as kernel

rc = kernel.RemoteController(('frinkiac1

rc.activate()

%autopx
Auto Parallel Enabled
Type %autopx to disable
import sys

[0] In [67]: import sys
[1] In [42]: import sys
sys.setdlopenflags(258)

[0] In [68]: sys.setdlopenflags(258)
[1] In [43]: sys.setdlopenflags(258)
from basin import *

[0] In [69]: from basin import *
[1] In [44]: from basin import *
reg0 = Region ("/home/vesperin/democl2.dat")

[0] In [96]: reg0 = Region ("/home/vesperin/democl2.dat")
[1] In [56]: reg0 = Region ("/home/vesperin/democl2.dat")
x = reg0.get_list ("RegionList0").get_attribute ("x")

[0] In [117]: x = reg0.get_list ("RegionList0").get_attribute ("x")
[1] In [65]: x = reg0.get_list ("RegionList0").get_attribute ("x")
y = reg0.get_list ("RegionList0").get_attribute ("y")

[0] In [136]: y = reg0.get_list ("RegionList0").get_attribute ("y")
[1] In [73]: y = reg0.get_list ("RegionList0").get_attribute ("y")
logx=log10(x)

[0] In [165]: logx=log10(x)
[1] In [86]: logx=log10(x)
```

Connected, Parallel Mode: On, 2 nodes

Toggle Parallel Mode

Save Batch...

Users

You may (de)select entire groups of commands to save to the batch based on the user who entered the command.

☑ test

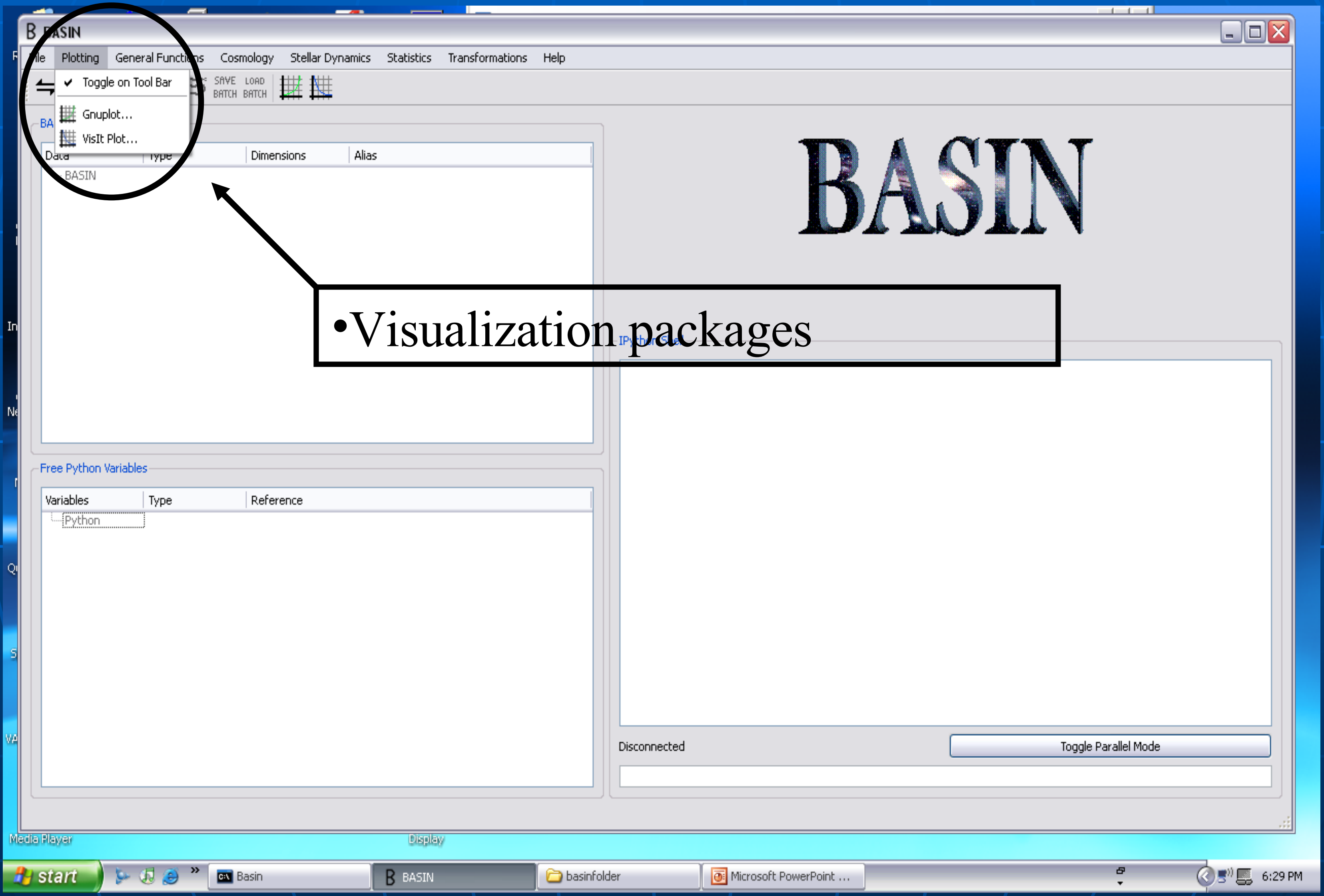
☑ Enrico

	Command	User
1	reg0 = Regi...	test
2	x = reg0.get...	test
3	y = reg0.get...	test
4	logx=log10(x)	test
5	logy=log10(y)	Enrico

Save | Cancel

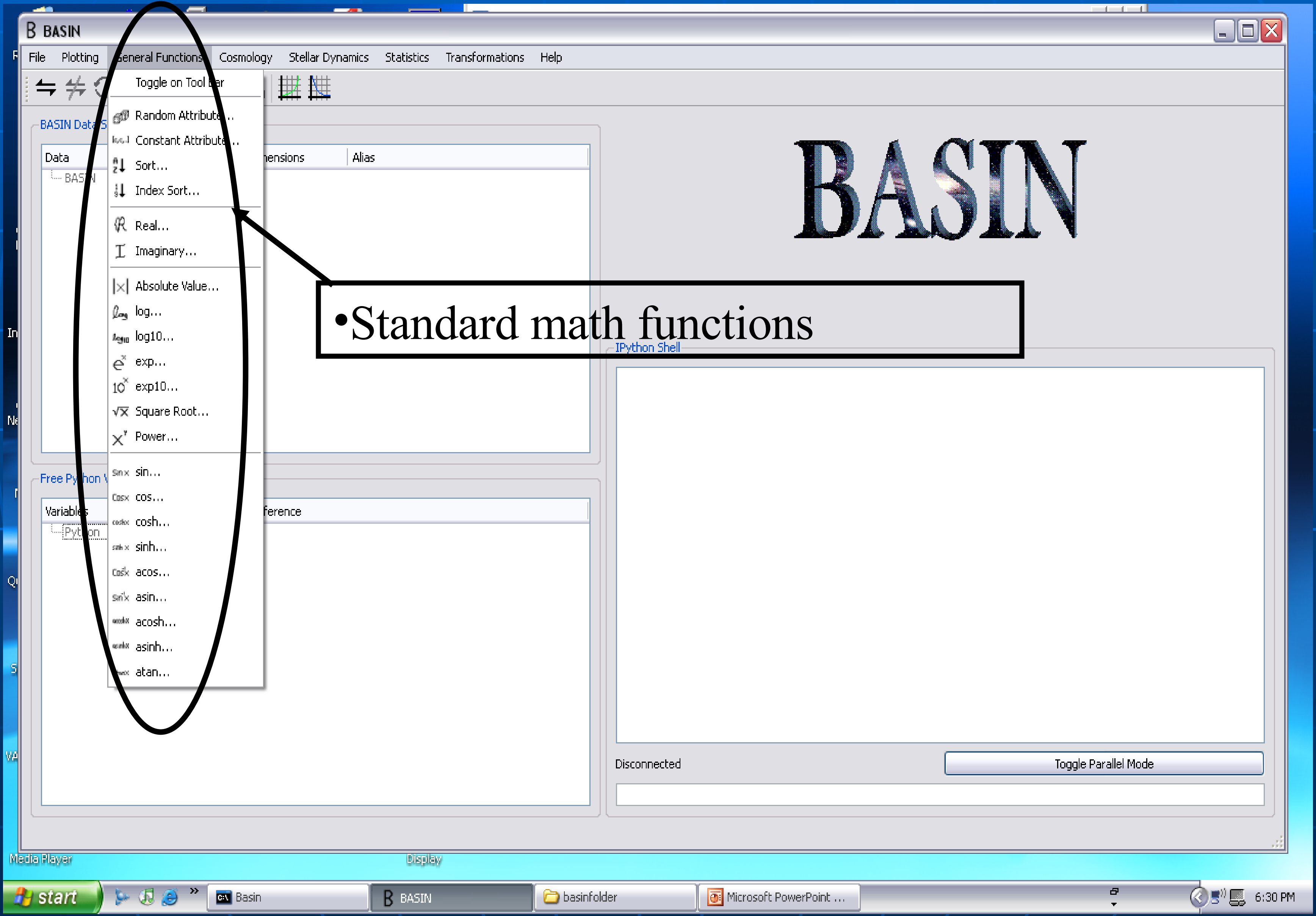
Clicking on a row will either select it (turning it white), or deselect it (turning it gray). Only those selected (white) commands will be saved to a batch file.

BASIN Python User Interface and GUI



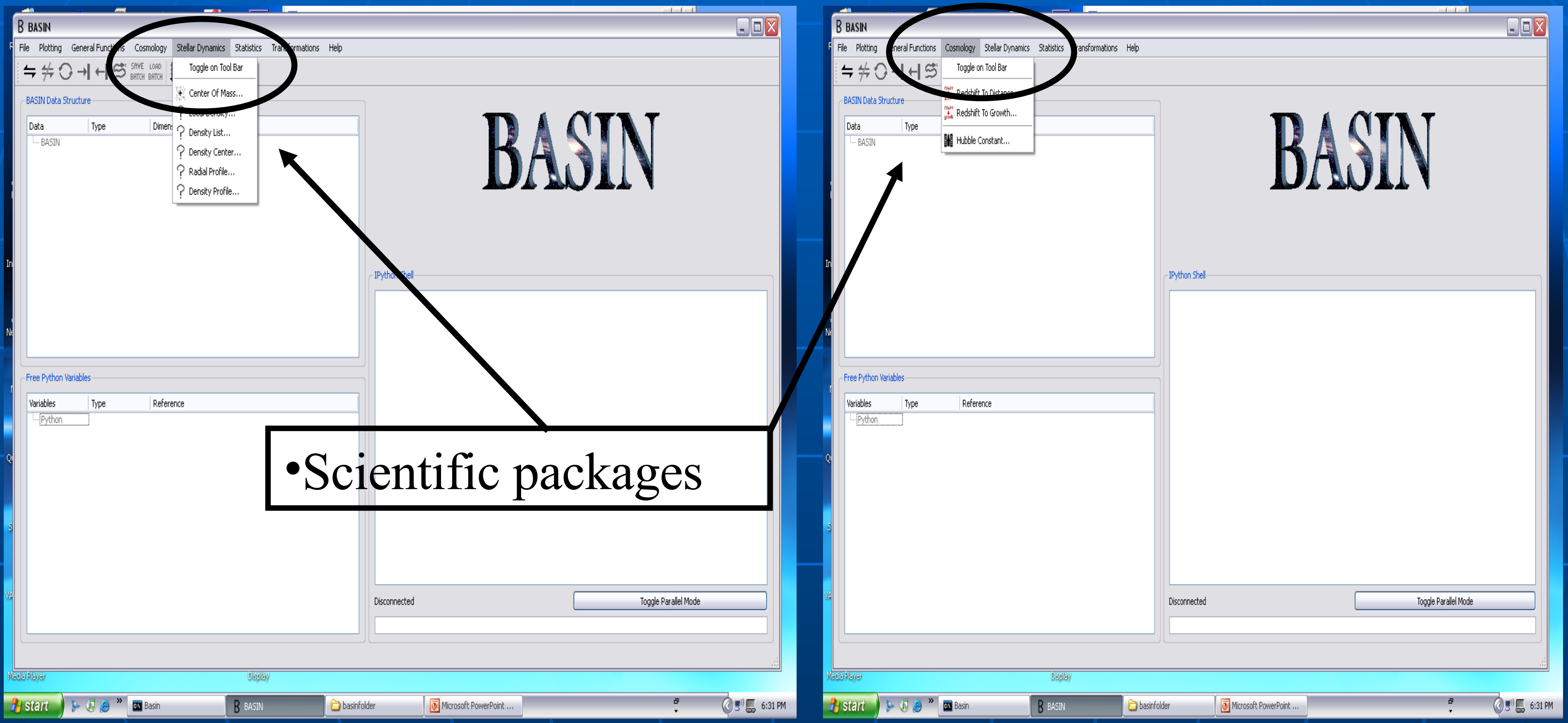
- Visualization packages

BASIN Python User Interface and GUI



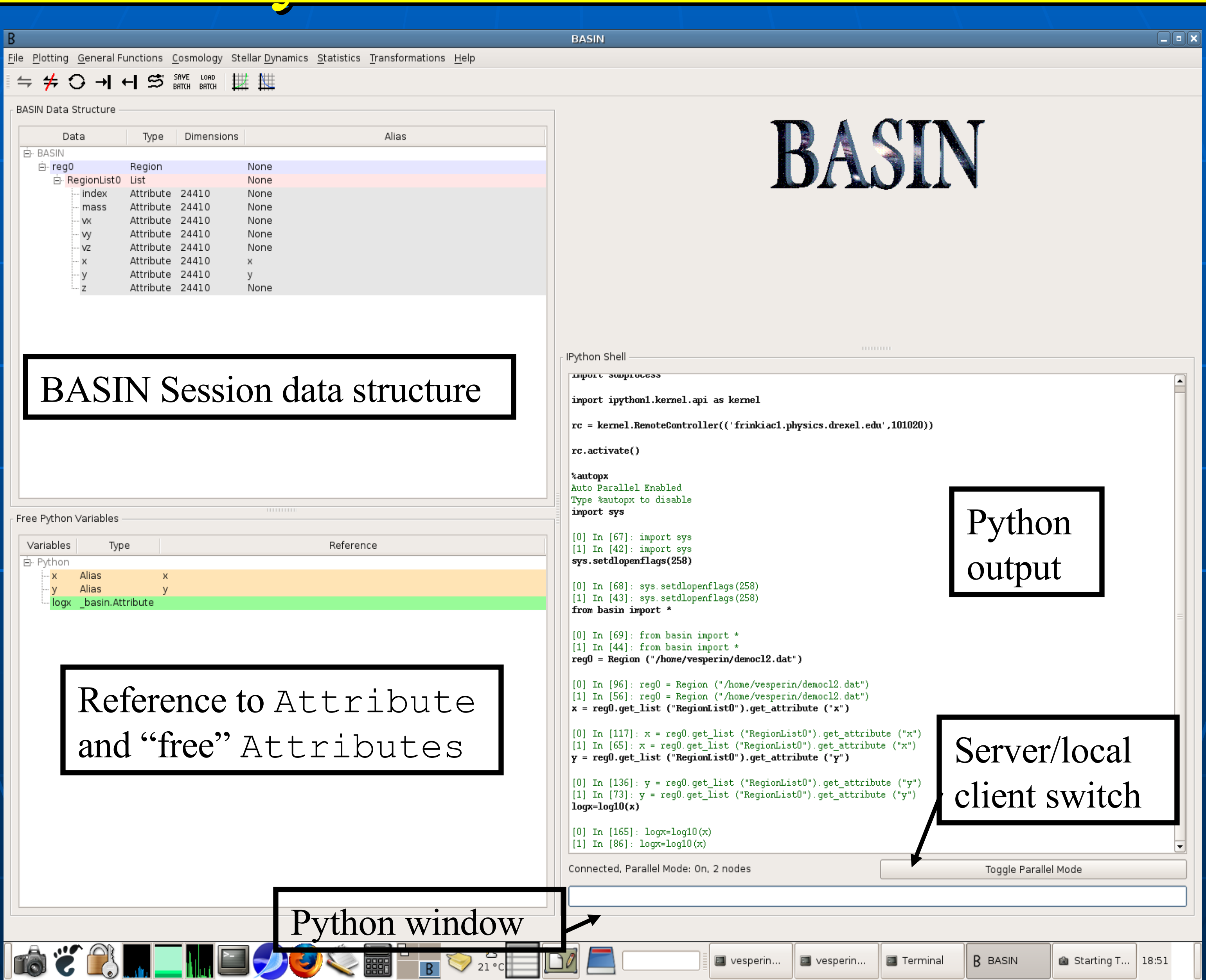
•Standard math functions

BASIN Python User Interface and GUI



- Scientific packages

BASIN Python User Interface and GUI


BASIN
File Plotting General Functions Cosmology Stellar Dynamics Statistics Transformations Help
SAVE BATCH
LOAD BATCH
BASIN
vesperin...
vesperin...
Terminal
BASIN
Starting T...
21 °C
18:51

BASIN Data Structure

Data	Type	Dimensions	Alias
BASIN			
reg0	Region		None
RegionList0	List		None
index	Attribute	24410	None
mass	Attribute	24410	None
vx	Attribute	24410	None
vy	Attribute	24410	None
vz	Attribute	24410	None
x	Attribute	24410	x
y	Attribute	24410	y
z	Attribute	24410	None

BASIN Session data structure

Free Python Variables

Variables	Type	Reference
Python		
x	Alias	x
y	Alias	y
logx	_basin.Attribute	

Reference to `Attribute` and “free” `Attributes`

IPython Shell

```
import subprocess

import ipython1.kernel.api as kernel

rc = kernel.RemoteController(('frinkiac1.physics.drexel.edu',101020))

rc.activate()

%autopx
Auto Parallel Enabled
Type %autopx to disable
import sys

[0] In [67]: import sys
[1] In [42]: import sys
sys.setdlopenflags(258)

[0] In [68]: sys.setdlopenflags(258)
[1] In [43]: sys.setdlopenflags(258)
from basin import *

[0] In [69]: from basin import *
[1] In [44]: from basin import *
reg0 = Region ("/home/vesperin/democl2.dat")

[0] In [96]: reg0 = Region ("/home/vesperin/democl2.dat")
[1] In [56]: reg0 = Region ("/home/vesperin/democl2.dat")
x = reg0.get_list ("RegionList0").get_attribute ("x")

[0] In [117]: x = reg0.get_list ("RegionList0").get_attribute ("x")
[1] In [65]: x = reg0.get_list ("RegionList0").get_attribute ("x")
y = reg0.get_list ("RegionList0").get_attribute ("y")

[0] In [136]: y = reg0.get_list ("RegionList0").get_attribute ("y")
[1] In [73]: y = reg0.get_list ("RegionList0").get_attribute ("y")
logx=log10(x)

[0] In [165]: logx=log10(x)
[1] In [86]: logx=log10(x)
```

Connected, Parallel Mode: On, 2 nodes

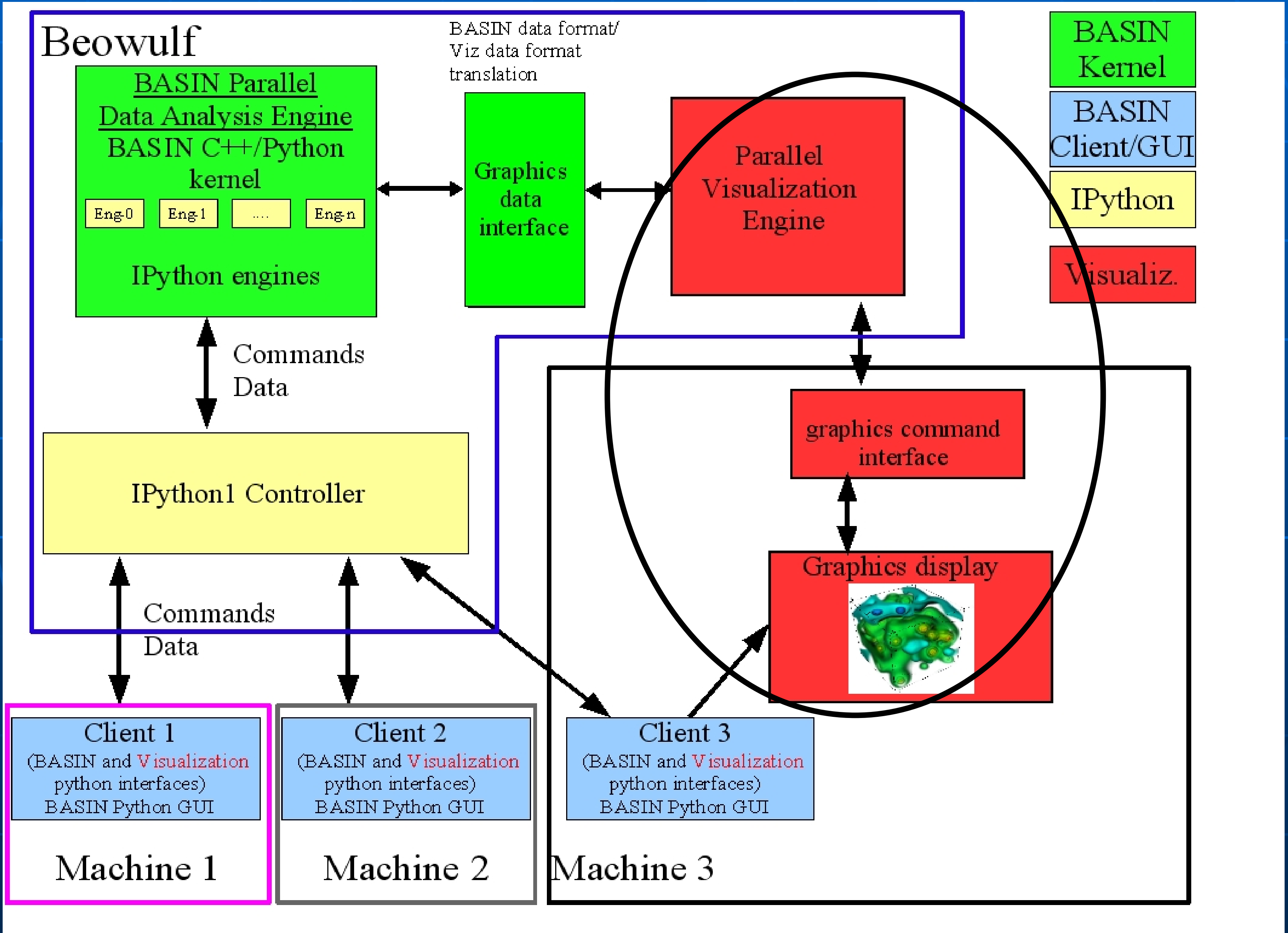
Toggle Parallel Mode

Python output

Server/local client switch

Python window

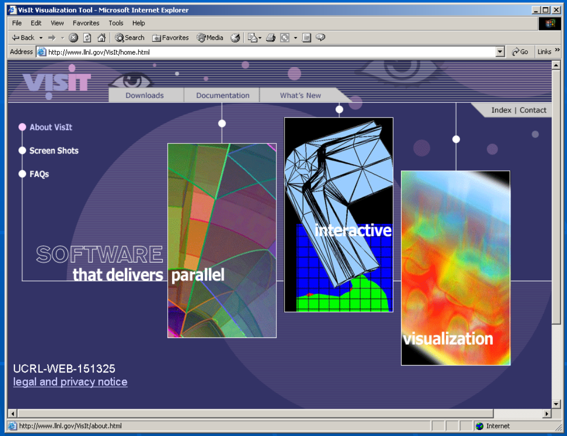
BASIN architecture



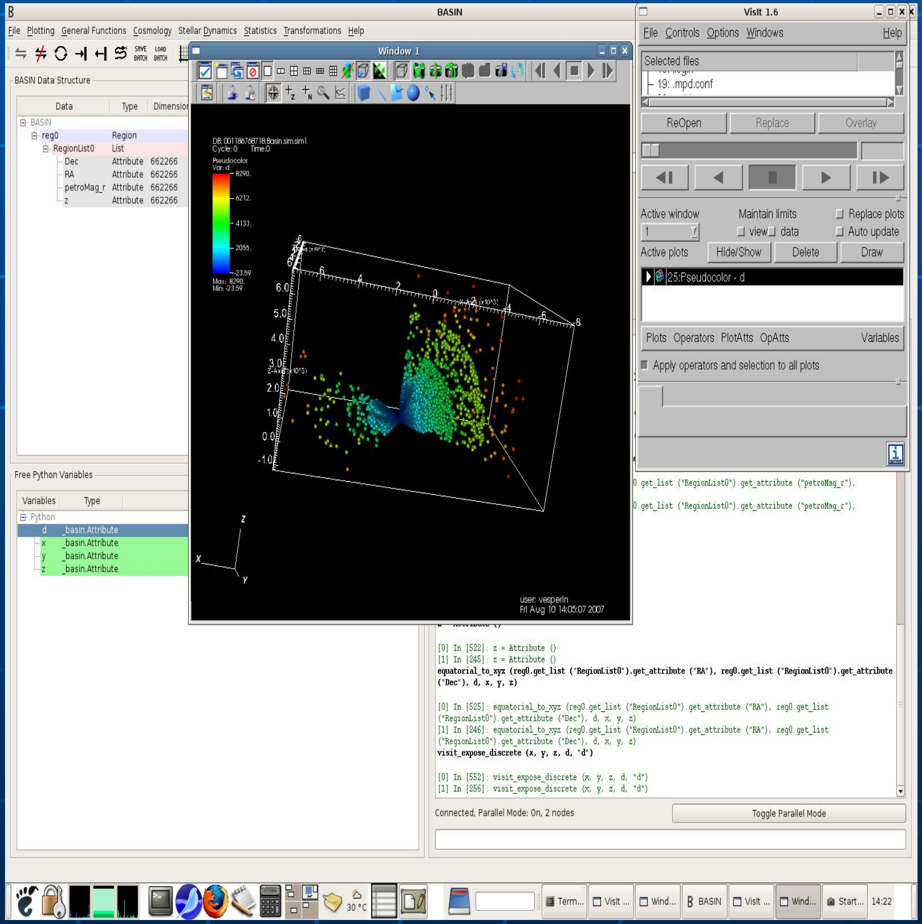
Visualization

VisIt

developed at LLNL
http://www.llnl.gov/visit



- Visualization of large distributed datasets (structured and unstructured meshes)
- Parallel visualization engine
- Available in BASIN (in collaboration with Brad Whitlock at LLNL)



Data	Type	Dimension
BASIN		
reg0	Region	
RegionList0	List	
Dec	Attribute	662266
RA	Attribute	662266
petroMag_r	Attribute	662266
z	Attribute	662266

Variables	Type
Python	
d	_basin.Attribute
x	_basin.Attribute
y	_basin.Attribute
z	_basin.Attribute

DB: 001186768718.basin.sim.sim1
Cycle: 0 Time:0

user: vesperin
Fri Aug 10 14:05:07 2007

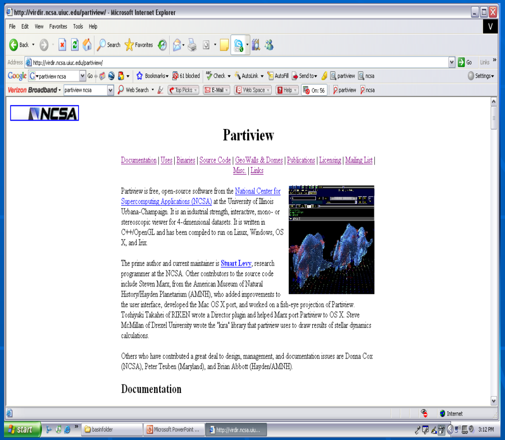
```
0.get_list ("RegionList0").get_attribute ("petroMag_r"),

0.get_list ("RegionList0").get_attribute ("petroMag_r"),

[0] In [522]: z = Attribute ()
[1] In [245]: z = Attribute ()
equatorial_to_xyz (reg0.get_list ("RegionList0").get_attribute ("RA"), reg0.get_list ("RegionList0").get_attribute
("Dec"), d, x, y, z)

[0] In [525]: equatorial_to_xyz (reg0.get_list ("RegionList0").get_attribute ("RA"), reg0.get_list
("RegionList0").get_attribute ("Dec"), d, x, y, z)
[1] In [246]: equatorial_to_xyz (reg0.get_list ("RegionList0").get_attribute ("RA"), reg0.get_list
("RegionList0").get_attribute ("Dec"), d, x, y, z)
visit_expose_discrete (x, y, z, d, "d")

[0] In [552]: visit_expose_discrete (x, y, z, d, "d")
[1] In [256]: visit_expose_discrete (x, y, z, d, "d")
```

Connected, Parallel Mode: On, 2 nodes

Visualization

Partiview

Developed at NCSA
http://virdir.ncsa.uiuc.edu/partiview/


http://virdir.ncsa.uiuc.edu/partiview/ - Microsoft Internet Explorer
NCSA
Internet
basinfolder
Microsoft PowerPoint ...
http://virdir.ncsa.uiu...
3:12 PM

Partiview

Documentation | Uses | Binaries | Source Code | GeoWalls & Domes | Publications | Licensing | Mailing List | Misc. | Links

Partiview is free, open-source software from the National Center for Supercomputing Applications (NCSA) at the University of Illinois Urbana-Champaign. It is an industrial strength, interactive, mono- or stereoscopic viewer for 4-dimensional datasets. It is written in C++/OpenGL and has been compiled to run on Linux, Windows, OS X, and Irix.

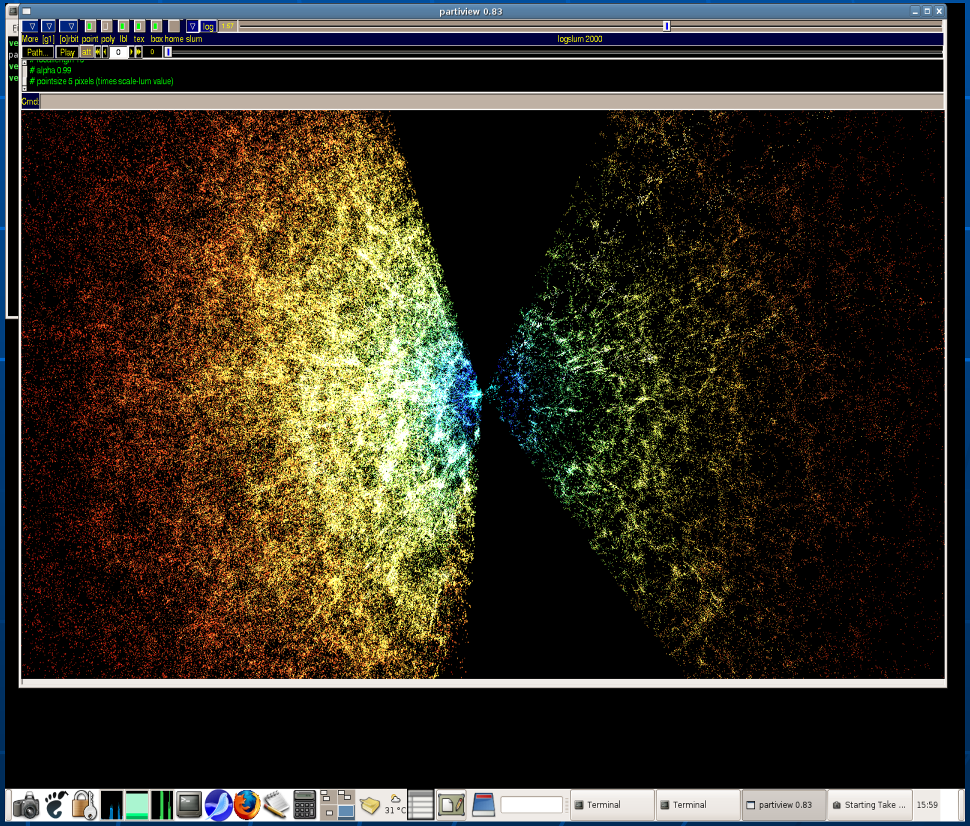
The prime author and current maintainer is Stuart Levy, research programmer at the NCSA. Other contributors to the source code include Steven Marx, from the American Museum of Natural History/Hayden Planetarium (AMNH), who added improvements to the user interface, developed the Mac OS X port, and worked on a fish-eye projection of Partiview. Toshiyuki Takahei of RIKEN wrote a Director plugin and helped Marx port Partiview to OS X. Steve McMillan of Drexel University wrote the "kira" library that partiview uses to draw results of stellar dynamics calculations.

Others who have contributed a great deal to design, management, and documentation issues are Donna Cox (NCSA), Peter Teuben (Maryland), and Brian Abbott (Hayden/AMNH).

Documentation

- Visualization of 4D particle datasets

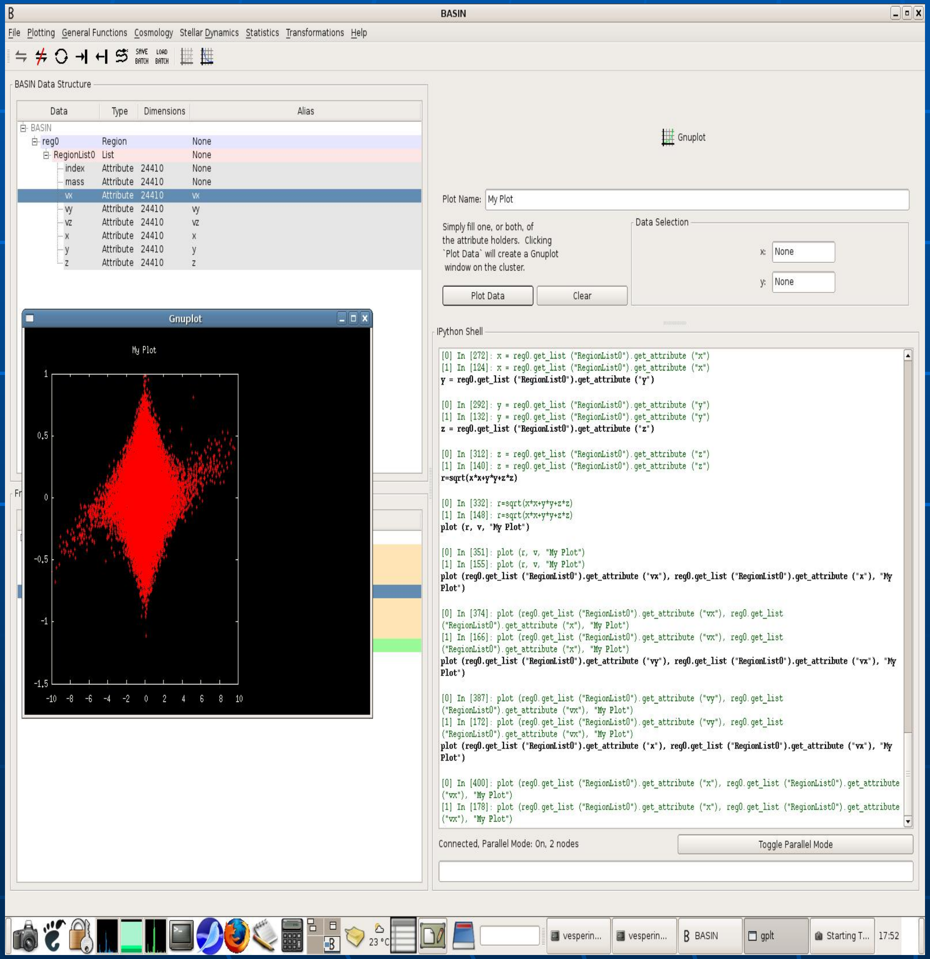
- BASIN/Partiview interface: work in progress (in collaboration with Stuart Levy, Matt Hall at NCSA)



```
# alpha 0.99
# pointsize 5 pixels (times scale-lum value)
```

Visualization

Gnuplot



Data	Type	Dimensions	Alias
BASIN			
reg0	Region		None
RegionList0	List		None
index	Attribute	24410	None
mass	Attribute	24410	None
vx	Attribute	24410	vx
vy	Attribute	24410	vy
vz	Attribute	24410	vz
x	Attribute	24410	x
y	Attribute	24410	y
z	Attribute	24410	z

Plot Name: My Plot

Simply fill one, or both, of the attribute holders. Clicking `Plot Data` will create a Gnuplot window on the cluster.

x: None

y: None

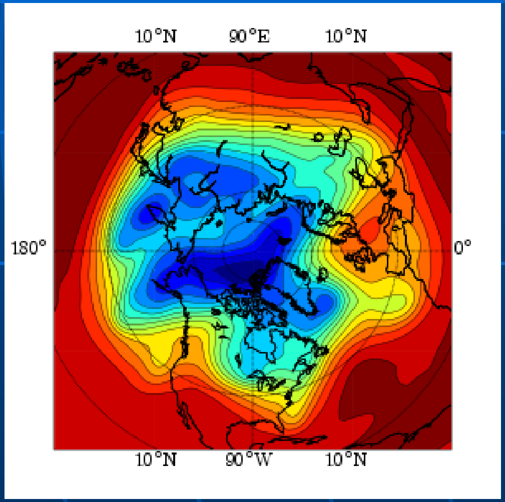
```
[0] In [272]: x = reg0.get_list ("RegionList0").get_attribute ("x")
[1] In [124]: x = reg0.get_list ("RegionList0").get_attribute ("x")
y = reg0.get_list ("RegionList0").get_attribute ("y")

[0] In [292]: y = reg0.get_list ("RegionList0").get_attribute ("y")
[1] In [132]: y = reg0.get_list ("RegionList0").get_attribute ("y")
z = reg0.get_list ("RegionList0").get_attribute ("z")

[0] In [312]: z = reg0.get_list ("RegionList0").get_attribute ("z")
[1] In [140]: z = reg0.get_list ("RegionList0").get_attribute ("z")
r=sqrt(x*x+y*y+z*z)

[0] In [332]: r=sqrt(x*x+y*y+z*z)
[1] In [148]: r=sqrt(x*x+y*y+z*z)
plot (r, v, "My Plot")

[0] In [351]: plot (r, v, "My Plot")
[1] In [155]: plot (r, v, "My Plot")
plot (reg0.get_list ("RegionList0").get_attribute ("vx"), reg0.get_list ("RegionList0").get_attribute ("x"), "My
Plot")

[0] In [374]: plot (reg0.get_list ("RegionList0").get_attribute ("vx"), reg0.get_list
("RegionList0").get_attribute ("x"), "My Plot")
[1] In [166]: plot (reg0.get_list ("RegionList0").get_attribute ("vx"), reg0.get_list
("RegionList0").get_attribute ("x"), "My Plot")
plot (reg0.get_list ("RegionList0").get_attribute ("vy"), reg0.get_list ("RegionList0").get_attribute ("vx"), "My
Plot")

[0] In [387]: plot (reg0.get_list ("RegionList0").get_attribute ("vy"), reg0.get_list
("RegionList0").get_attribute ("vx"), "My Plot")
[1] In [172]: plot (reg0.get_list ("RegionList0").get_attribute ("vy"), reg0.get_list
("RegionList0").get_attribute ("vx"), "My Plot")
plot (reg0.get_list ("RegionList0").get_attribute ("x"), reg0.get_list ("RegionList0").get_attribute ("vx"), "My
Plot")

[0] In [400]: plot (reg0.get_list ("RegionList0").get_attribute ("x"), reg0.get_list ("RegionList0").get_attribute
("vx"), "My Plot")
[1] In [178]: plot (reg0.get_list ("RegionList0").get_attribute ("x"), reg0.get_list ("RegionList0").get_attribute
("vx"), "My Plot")
```

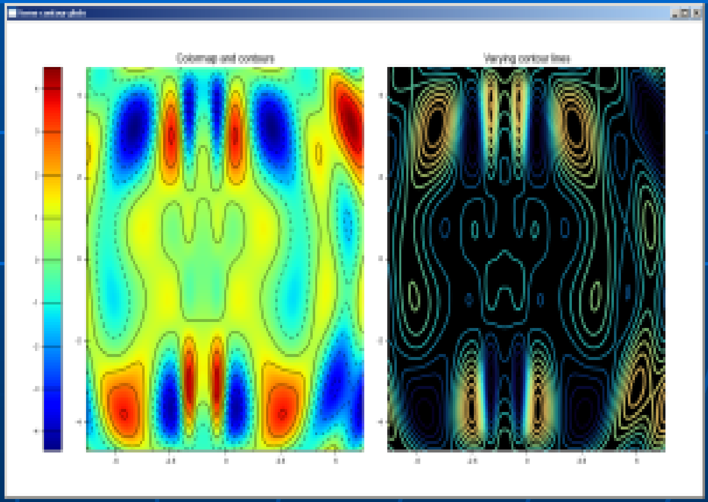
Connected, Parallel Mode: On, 2 nodes

- Simple plotting package based on the Gnuplot API.
- Only for development purposes.
- Available in BASIN

Visualization

Data transferred to the client machine: all the Python plotting packages (e.g. Matplotlib, Gnuplot, Chaco, etc.)





Summary

Goals:

Ease access to parallel data analysis

Avoid redundant development

Interactive and multi-user parallel data analysis

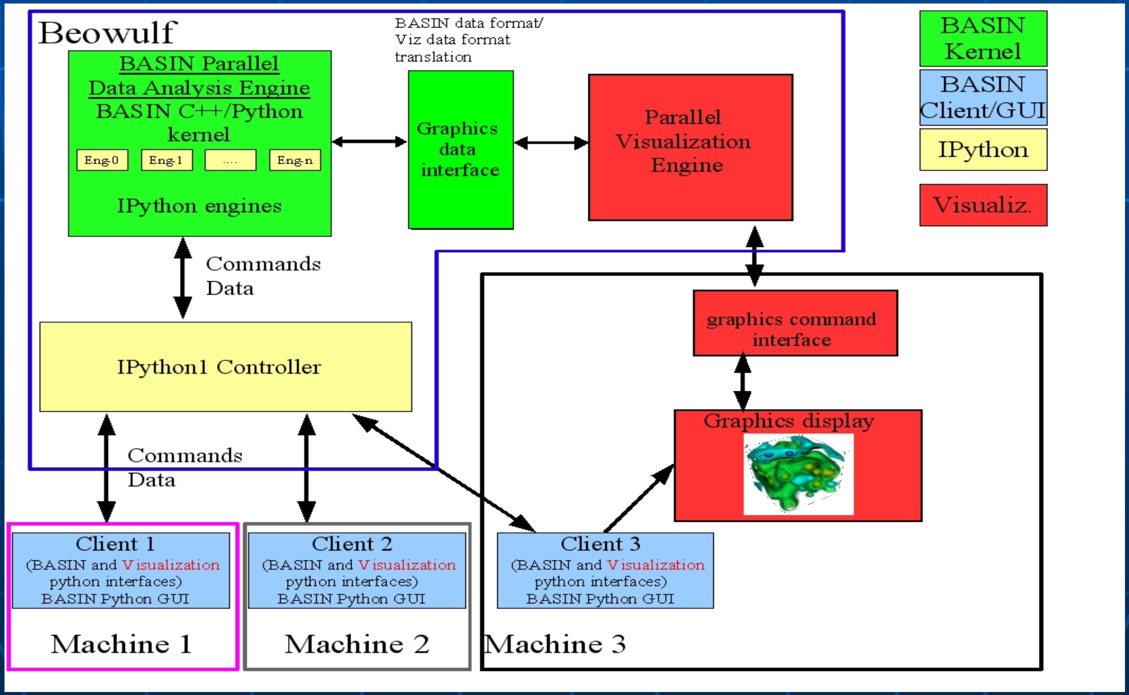
Summary

What we have:

Kernel for parallel data management and operations (C++/Python)

Scientific packages

Interface to a few existing visualization packages



Summary

What to look for next:

Increase science scope beyond astrophysics

Extend visualization options

Two-way communication with visualization packages

Improve ease of use and installation