

SciServer



Bringing analysis close to (your) data

Gerard Lemson
Manuchehr Taghizadeh-Popp
Johns Hopkins University

sciserver.org



JOHNS HOPKINS
UNIVERSITY



idies

Motivation

- ▶ Big part of science is about data.
(data collection, cleaning, analysis, publishing, mirroring, etc.)
- ▶ BIG DATA: can't download to laptop for analysis (100 TB+)

 Bring analysis close to data.

SciServer: Data infrastructure system with 4 goals.

Give scientists web tools providing...

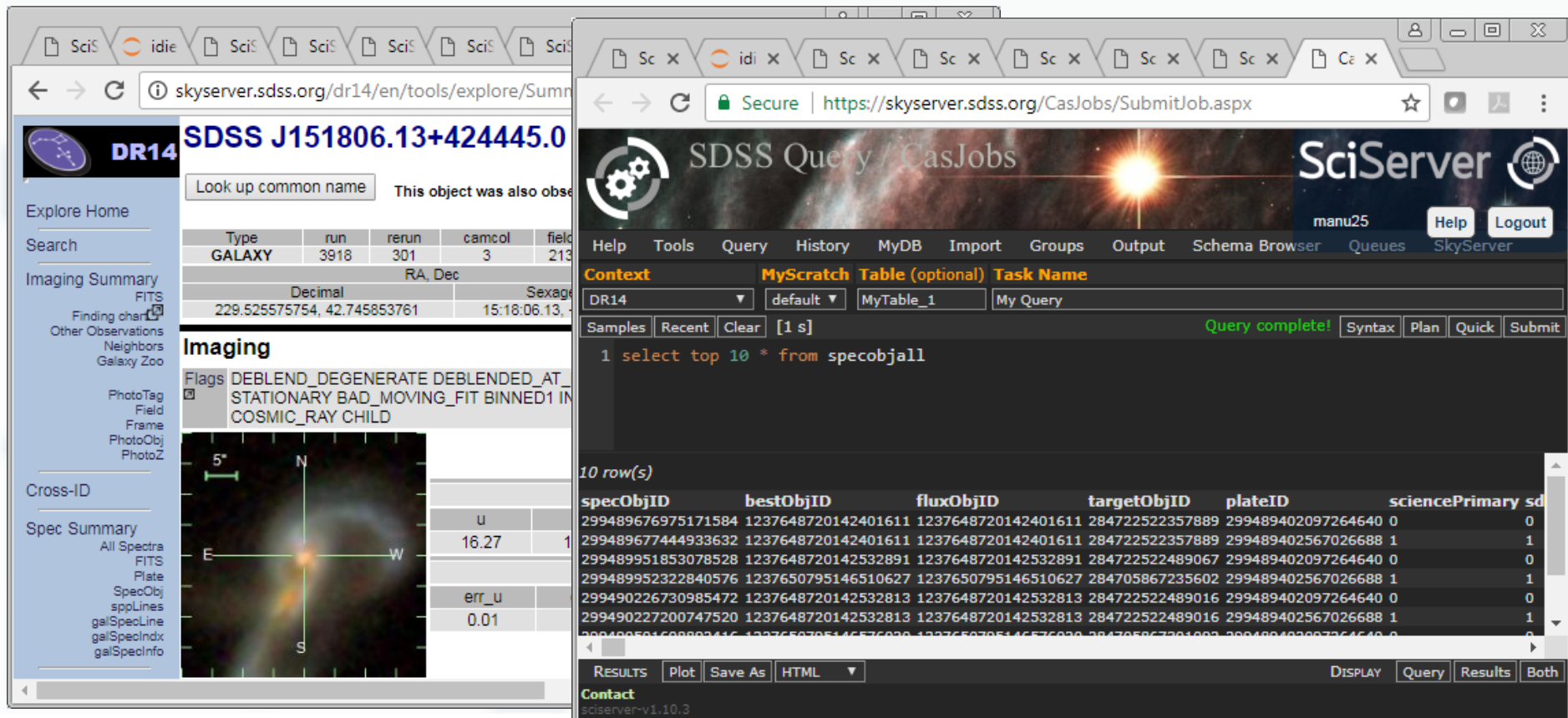
- 1) ...hosting of huge public/private datasets.
- 2) ...data-intensive computing for everyone.
- 3) ...personal data storage space.
- 4) ...capability for sharing data within a team.

Based at Johns Hopkins University.

History: analysis close to data, RDB+SQL

Early 2000s: websites exposing SDSS database.

- ▶ **SkyServer**: for exploring sky objects.
- ▶ **CasJobs**: asynch SQL queries, personal database storage.



The screenshot shows two browser windows from the SkyServer website. The left window displays the details for SDSS J151806.13+424445.0, including a table of object properties and an imaging summary. The right window shows the CasJobs interface where a SQL query was executed, returning a table of object IDs and flux measurements.

SDSS J151806.13+424445.0

Type	run	rerun	camcol	field
GALAXY	3918	301	3	213

RA, Dec: 229.525575754, 42.745853761

Imaging Summary: DEBLEND_DEGENERATE, DEBLENDED_AT, STATIONARY, BAD_MOVING_FIT, BINNED1 IN COSMIC_RAY CHILD

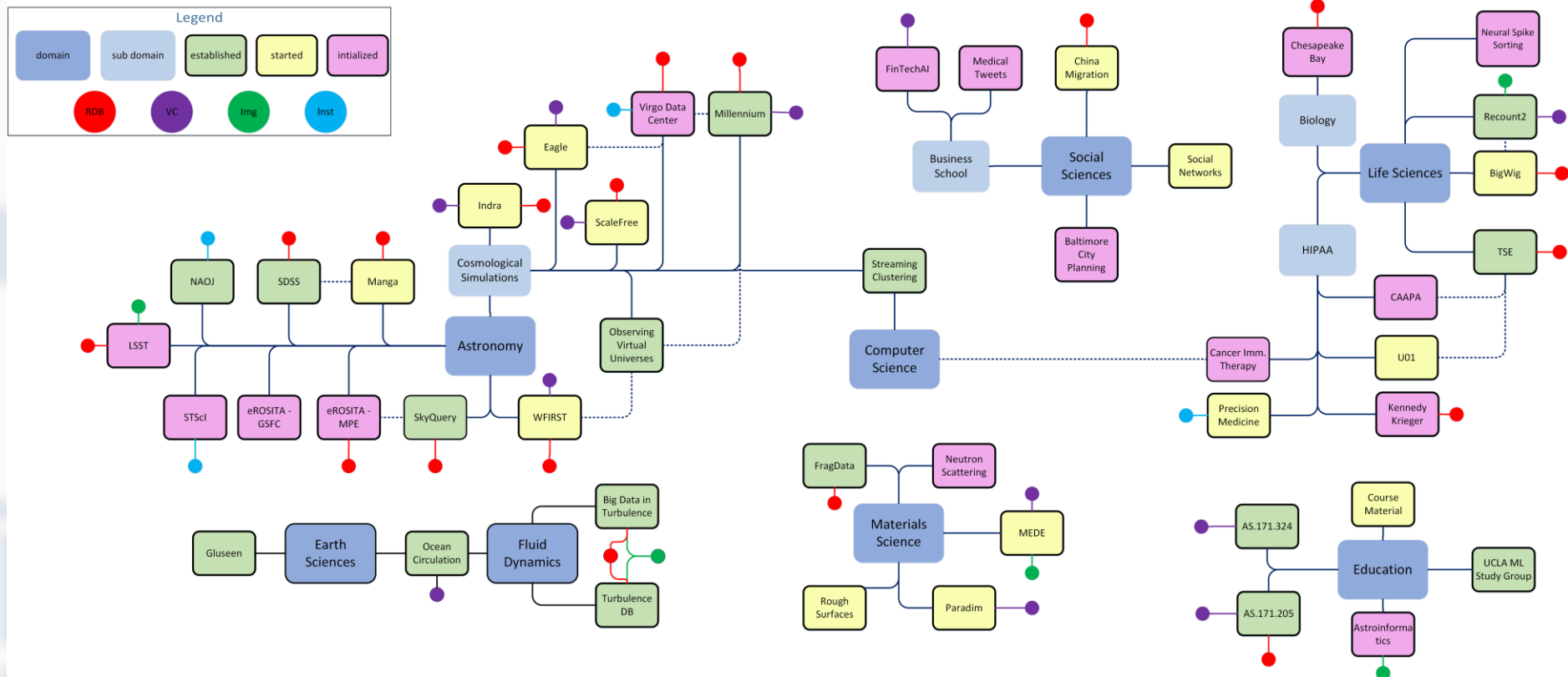
10 row(s)

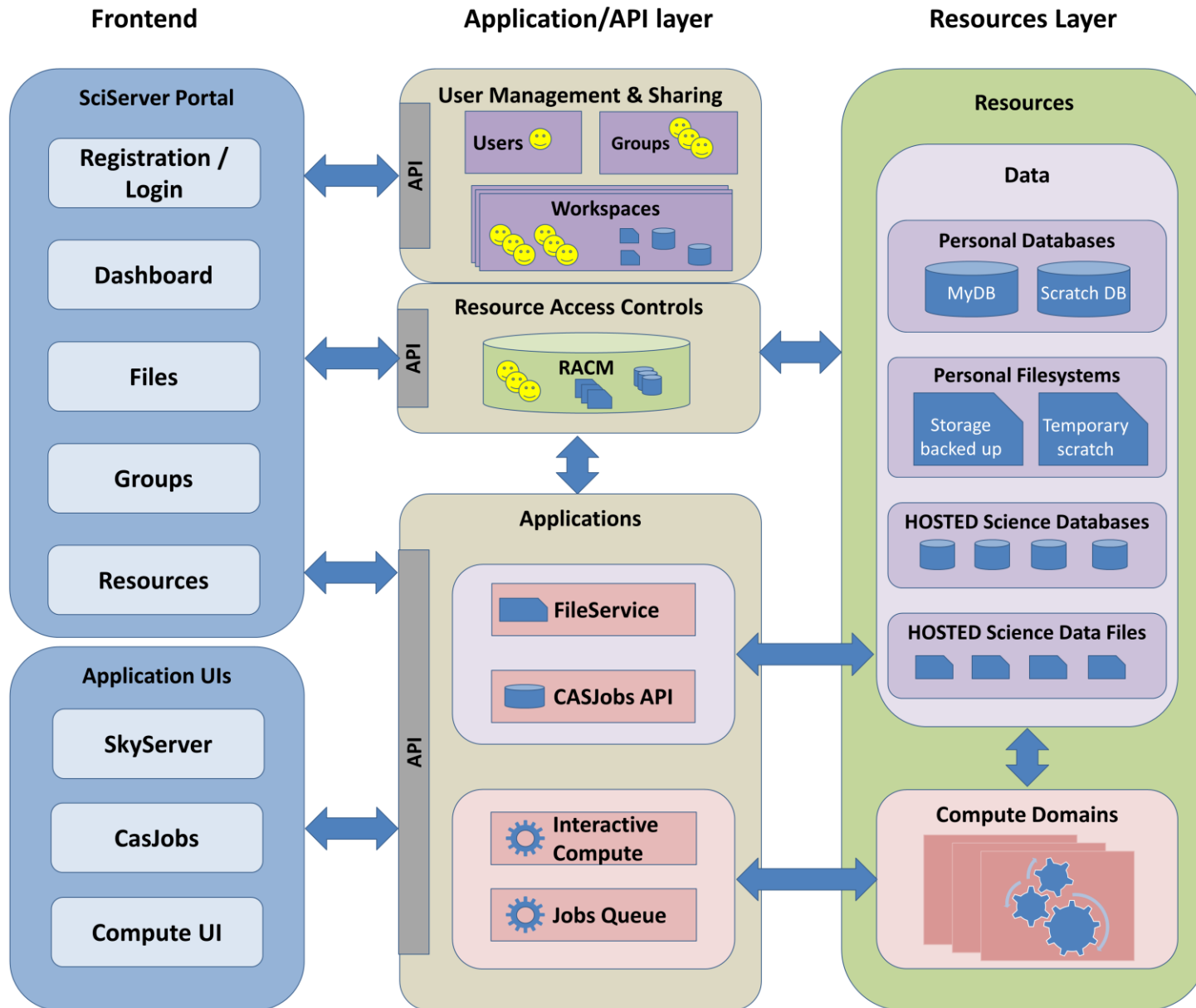
specObjID	bestObjID	fluxObjID	targetObjID	plateID	sciencePrimary	sd
299489676975171584	1237648720142401611	1237648720142401611	284722522357889	299489402097264640	0	0
299489677444933632	1237648720142401611	1237648720142401611	284722522357889	299489402567026688	1	1
299489951853078528	1237648720142532891	1237648720142532891	284722522489067	299489402097264640	0	0
299489952322840576	1237650795146510627	1237650795146510627	284705867235602	299489402567026688	1	1
299490226730985472	1237648720142532813	1237648720142532813	284722522489016	299489402097264640	0	0
299490227200747520	1237648720142532813	1237648720142532813	284722522489016	299489402567026688	1	1

What's New

- Data-analysis capability with Jupyter Notebooks.
 - python, R(Rstudio), Matlab, Julia, ...
 - terminal: conda/pip, git, gcc, ...
- Creation of teams and sharing private resources.
 - use in class room: course ware
 - discussed in GWS1
- Expansion to all sciences:**
Genomics, Oceanography, Material Science, Turbulence, Humanities, Health, ...

Supported science projects

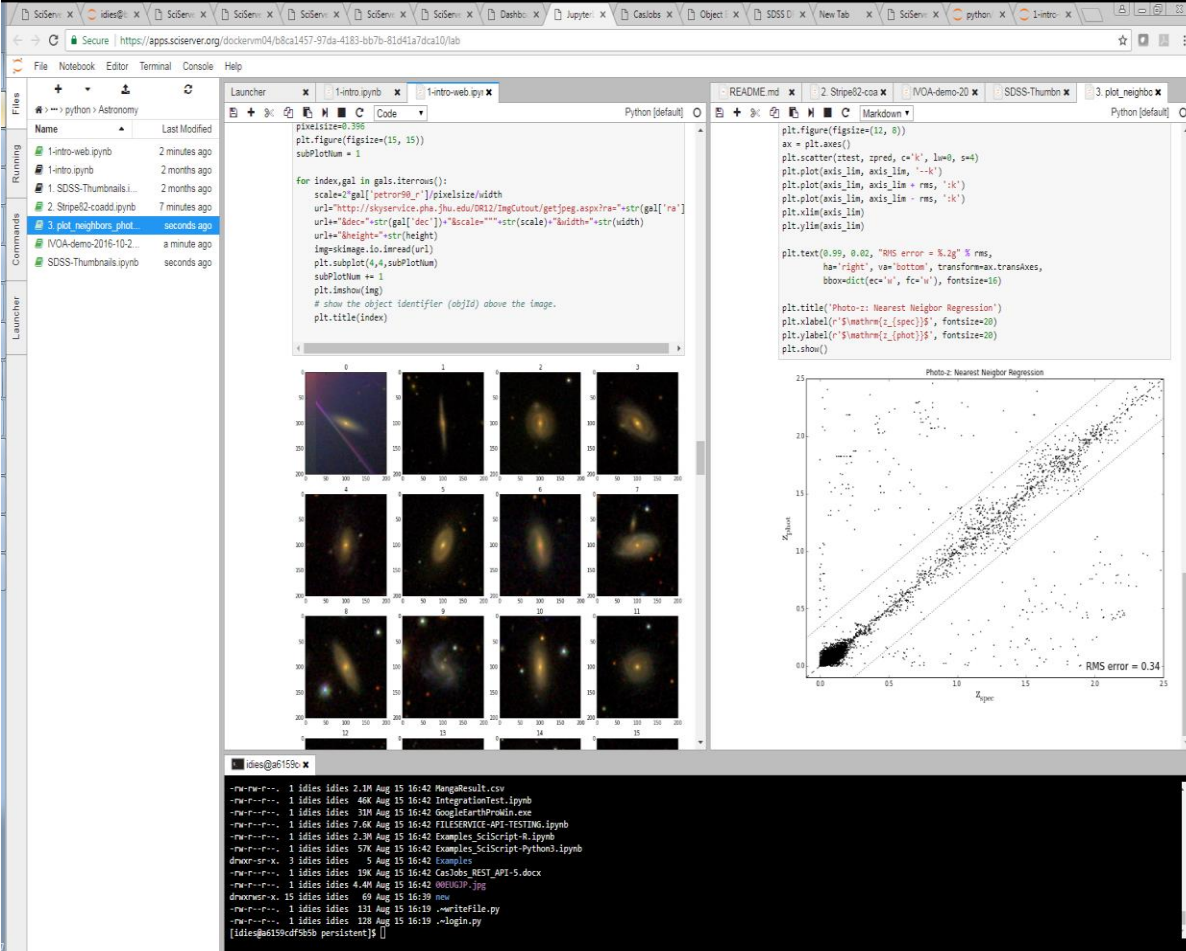




SciServer-Compute

Data-intensive computing with Jupyter Notebooks in Docker Containers.

- Containers give isolated Linux environment
- Private or public data volumes in file system.
- Notebooks in Python, R, Matlab.
- SciScript libraries: for loading external data into Notebook.
- Also Notebooks as **batch Jobs**.



The screenshot displays a Jupyter Notebook environment within a Docker container. The interface includes a file browser on the left, a code editor in the center, and a terminal at the bottom. The code in the notebook performs the following steps:

```

pixelsize=0.396
plt.figure(figsize=(15, 15))
subplotnum = 1

for index,gal in gals.iterrows():
    scale=2*gal['petror90_r']/pixelsize/width
    url="http://skyservice.pha.jhu.edu/D012/ImgOutput/get?jpeg.aspx?ra="+str(gal['ra'])*url+ "&dec="+str(gal['dec'])+"&scale="+str(scale)+"&width="+str(width)
    url+"&height="+str(height)
    img=skimage.io.imread(url)
    plt.subplot(4,4,subplotnum)
    subplotnum += 1
    plt.imshow(img)
    # show the object identifier (objID) above the image.
    plt.title(index)
  
```

The notebook output shows a 4x4 grid of galaxy images (indices 0-15) and a scatter plot titled "Photo-z: Nearest Neighbor Regression". The scatter plot shows the relationship between z_{phot} (y-axis) and z_{spec} (x-axis), with a regression line and shaded confidence intervals. The RMS error is 0.34.

```

plt.figure(figsize=(12, 8))
ax = plt.axes()
plt.scatter(test, zpred, c='k', lw=0, s=4)
plt.plot(axLim, axLim, '-k')
plt.plot(axLim, axLim - rms, ':k')
plt.plot(axLim, axLim + rms, ':k')
plt.xlim(axLim)
plt.ylim(axLim)

plt.text(0.99, 0.02, "RMS error = %.2g" % rms,
        ha='right', va='bottom', transform=ax.transAxes,
        bboxdict(ecw='w', fc='w'), fontsize=16)

plt.title("Photo-z: Nearest Neighbor Regression")
plt.xlabel(r"$z_{\text{spec}}$", fontsize=20)
plt.ylabel(r"$z_{\text{phot}}$", fontsize=20)
plt.show()
  
```

The terminal at the bottom shows a list of batch jobs:

```

[ides@u6159c persistent]$
-rw-rw-rw. 1 ides ides 2.1M Aug 15 16:42 MangaResult.csv
-rw-rw-rw. 1 ides ides 48K Aug 15 16:42 IntegrationTest.ipynb
-rw-rw-rw. 1 ides ides 319 Aug 15 16:42 GoogleEarthWebin.exe
-rw-rw-rw. 1 ides ides 7.6K Aug 15 16:42 FILESERVICE-API-TESTING.ipynb
-rw-rw-rw. 1 ides ides 2.3M Aug 15 16:42 Examples_SciScript-R.ipynb
-rw-rw-rw. 1 ides ides 57K Aug 15 16:42 Examples_SciScript-Python3.ipynb
drwxr-xr-x. 3 ides ides 5 Aug 15 16:42 Examples
-rw-rw-rw. 1 ides ides 19K Aug 15 16:42 CasJobs_REST_API-5.docx
-rw-rw-rw. 1 ides ides 4.4M Aug 15 16:42 00EUG19.jpg
drwxr-xr-x. 15 ides ides 69 Aug 15 16:39 new
-rw-rw-rw. 1 ides ides 128 Aug 15 16:19 -writefile.py
-rw-rw-rw. 1 ides ides 128 Aug 15 16:19 -login.py
[ides@u6159c persistent]$
  
```


Short demo (?)



CAS + DAS

jupyter 2. Stripe82-coadd-Copy1 Last Checkpoint: 3 minutes ago (unsaved changes)

```

import sciServer.Login as login
token=login.getToken()
import pandas
import tables
import numpy as np
import astropy
from astropy.io import fits
from astropy import wcs
import skimage.io
import urllib
import os
import matplotlib
import matplotlib.pyplot as plt
    
```

In [18]:

```

sql="""
SELECT a.objid as head, c.objid2 as match, b.matchcount,
p.fieldid as head_field, d.fieldid as match_field,
dbo.fGetUrlFitsCFrame(d.fieldid, 'g') as fits_g,
dbo.fGetUrlFitsCFrame(d.fieldid, 'r') as fits_r,
dbo.fGetUrlFitsCFrame(d.fieldid, 'z') as fits_z,
p.ra, d.ra as match_ra, p.dec, d.dec as match_dec
, p.petrors90_r
from (select top 1 * from galaxy where objid=8658194378960928809) a
join matchhead b on a.objid=b.objid -- join with matchhead
join photoobj p on a.objid=p.objid -- get matchhead photoobj
join match c on c.objid1=b.objid -- join with all the matches
join photoobjall d on c.objid2=d.objid -- get match photoobj
order by d.fieldid
"""
queryResponse = SciServer.CasJobs.executeQuery(sql, "Stripe82", token=token)
objs = pandas.read_csv(queryResponse, index_col=None)
objs[:10]
    
```

	head	match	matchcount	head_field	match_field	fits_g
0	8658194378960928809	865819443049955320	57	8658194378960928768	8658194430499553280	http://das.sdss.org/imaging/5622/40/corr/
1	8658194378960928809	865819447742948377	57	8658194378960928768	865819447742948352	http://das.sdss.org/imaging/5633/40/corr/
2	8658194378960928809	8658194516375371821	57	8658194378960928768	8658194516375371776	http://das.sdss.org/imaging/5642/40/corr/
3	8658194378960928809	8658194585083510793	57	8658194378960928768	8658194585083510784	http://das.sdss.org/imaging/5658/40/corr/
4	8658194378960928809	8658194804163018771	57	8658194378960928768	8658194804163018752	http://das.sdss.org/imaging/5709/40/corr/
5	8658194378960928809	8658194954470752297	57	8658194378960928768	8658194954470752256	http://das.sdss.org/imaging/5744/40/corr/
6	8658194378960928809	8658195018907910161	57	8658194378960928768	8658195018907910144	http://das.sdss.org/imaging/5759/40/corr/
7	8658194378960928809	8658195044651040803	57	8658194378960928768	8658195044651040768	http://das.sdss.org/imaging/5765/40/corr/
8	8658194378960928809	8658195066151239724	57	8658194378960928768	8658195066151239680	http://das.sdss.org/imaging/5770/40/corr/
9	8658194378960928809	8658195113395748890	57	8658195113395748864	8658195113395748864	http://das.sdss.org/imaging/5781/40/corr/

jupyter 2. Stripe82-coadd-Copy1 Last Checkpoint: a minute ago (autosaved)

Out [17]: <matplotlib.text.Text at 0x7f2aa174a6d8>

Dark matter halos Millennium Simulation

```

jupyter CosmoUC_5_PlotHalo Last Checkpoint: 11 minutes ago (autosaved)
File Edit View Insert Cell Kernel Help Notebook saved Python 3
+ - < > ↺ ↻ ⏪ ⏩ Code Cell Toolbar

Dark-matter Halos in a Cosmological Simulation

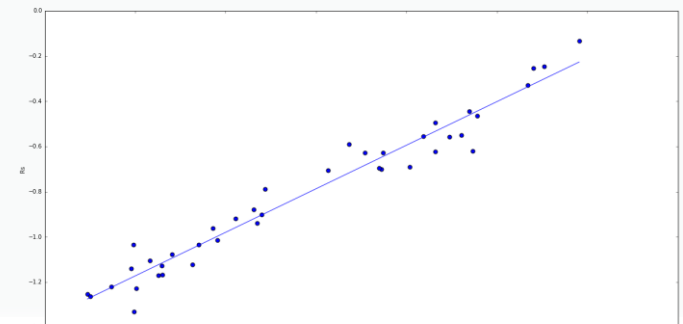
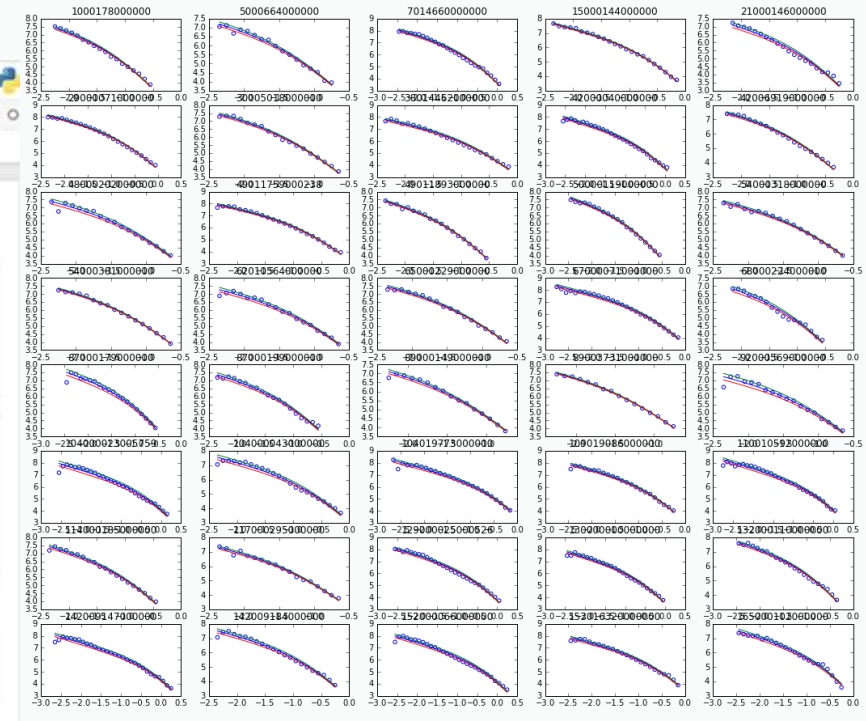
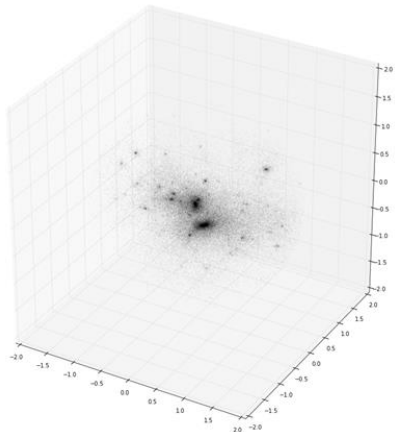
In [3]: import SciServer.LoginPortal as Login
        token = Login.getToken()
        import SciServer.CasJobs
        import pandas
        import tables
        import numpy as np
        import matplotlib.pyplot as plt
        from mpl_toolkits.mplot3d import Axes3D

In [10]: %%time
        queryString = """
        select top 100000 p.x-hh.x as x,p.y-hh.y as y,p.z-hh.z as z
        from mpahalotrees.mr hh
        cross apply dbo.MillenniumParticles(hh.snapnum,
        dbo.Sphere::New(hh.x,hh.y,hh.z,3*hh.halfmassradius).ToString()) p
        where hh.haloId=84000007000000 order by newid()
        """
        responseStream = SciServer.CasJobs.executeQuery(queryString, token=token,context="SimulationDB")
        df = pandas.read_csv(responseStream, index_col="Name")

        CPU times: user 351 ms, sys: 184 ms, total: 535 ms
        Wall time: 5.27 s

In [13]: fig = plt.figure(figsize=(15, 15))
        ax = fig.add_subplot(111, projection='3d')
        ax.scatter(df.x,df.y, df.z,s=

Out [13]: <mpl_toolkits.mplot3d.art3d.Pr
    
```



Thank you

