SciServer

Collaborative data-driven science

Data Access and Analysis beyond SQL

Gerard Lemson
Alex Szalay, Mike Rippin
DIBBS/SciServer

JOHNS HOPKINS UNIVERSITY

NSF

idies

Motivation and History

- Started with the SDSS SkyServer
- Built in a few months in 2001
- Goal: instant access to rich content
- Idea: bring the analysis to the data
- Interactive access at the core
- Much of the scientific process is about data
  - Data collection, data cleaning, data archiving, data organization, data publishing, mirroring, data distribution, data analytics, data curation…

Form Based Queries

skyserver.sdss.org/dr12/en/tools/search/radial.aspx

blogs | news | nnx | jhu | Python | JSON | code | ML | personal | App Services - | U-SQL for Big Data -

SLOAN DIGITAL SKY SURVEY III

SkyServer DR12

Home | Data | Schema | Education | Astronomy | SDSS | Contact Us | Download | Site Search | Help

DR12 Tools

- Getting Started
- Famous places
- Get images
- Scrolling sky
- Visual Tools
- Search
  - Radial
  - Rectangular
  - Search Form
  - SQL
  - Imaging Query
  - Spectro Query
- Object Crossid
- CasJobs

Radial Search

NOTE: To be fair to other users, queries run from SkyServer search tools are restricted in how long they can run and how much output they return, by **timeouts** and **row limits**. Please see the Query Limits help page. To run a query that is not restricted by a timeout or number of rows returned, please use the CasJobs batch query service.

Type of search	(•) Equitorial (RA / Dec) ( ) Galactic (*l* / *b*)
RA	258.25
Dec	64.05
radius [arcmins]	3

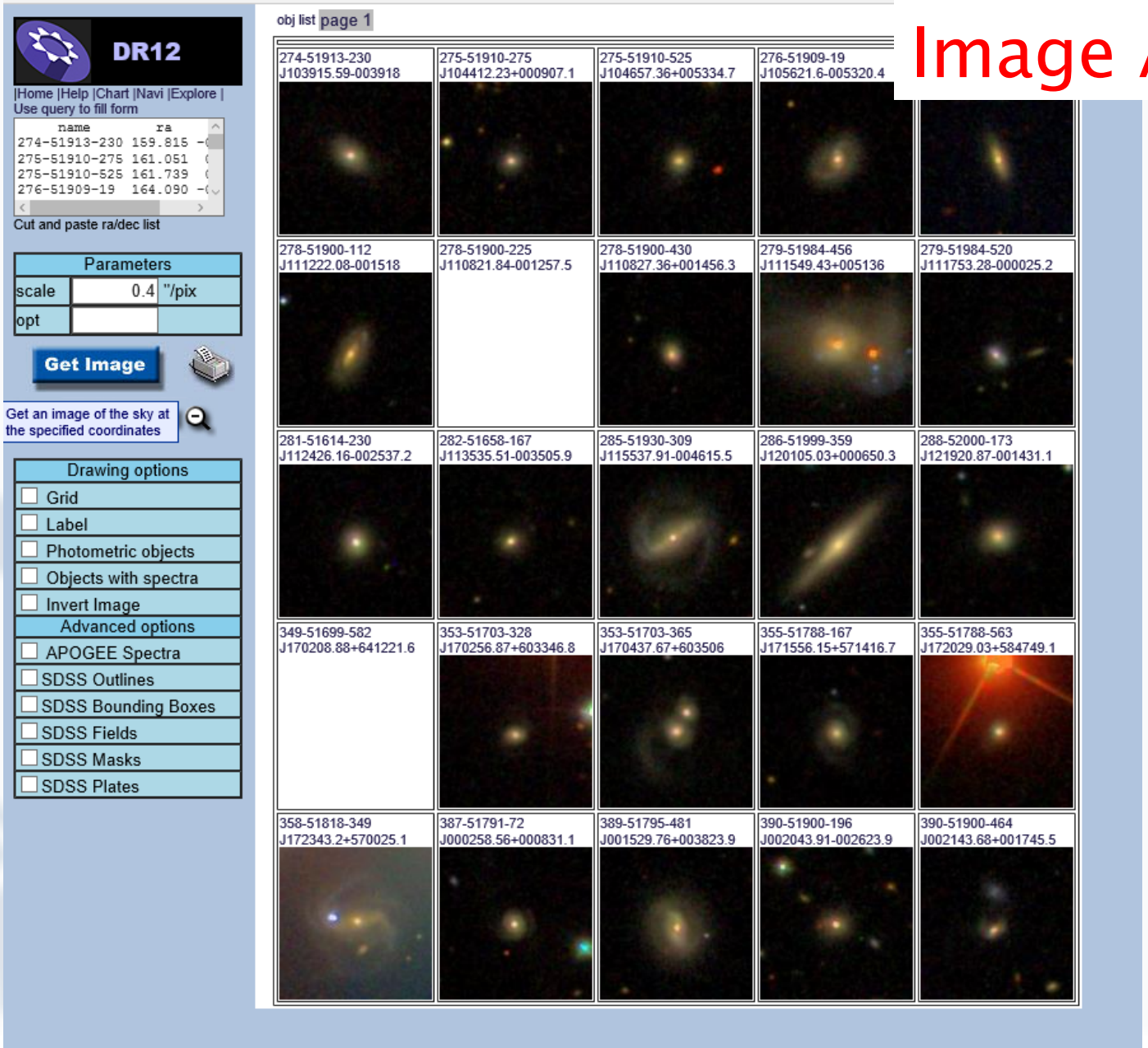
	Min		Max
☐	0	u	20
☐	0	g	20
☐	0	r	20
☐	0	i	20
☐	0	z	20

Output Format (•) HTML ( ) XML ( ) CSV ( ) JSON ( ) VOTable ( ) FITS

Submit Request | Limit number of output rows (0 for max) to 10 | Reset Form

Enter the *ra* and *dec* either in degrees or in h:m:s, d:m:s notation. The search *radius* is measured in arcminutes. Check the magnitudes you would like to constrain in your query. If you prefer not to use specific attributes,leave those rows unchecked. (If you do not insert constraints and select all entries, you will receive many records!)

Image Access



Custom SQL

SLOAN DIGITAL SKY SURVEY III

SkyServer DR12

Home | Data | Schema | Education | Astronomy | SDSS | Contact Us | Download | Site Search | Help

DR12 Tools

- Getting Started
- Famous places
- Get images
- Scrolling sky
- Visual Tools
- Search
  - Radial
  - Rectangular
  - Search Form
  - SQL
  - Imaging Query
  - Spectro Query
- Object Crossid
- CasJobs

SQL Search

This page allows you to directly submit a SQL (Structured Query Language) query to the SDSS database server. You can modify the default query as you wish, or cut and paste a query from the SDSS Sample Queries page.

Please note: To be fair to other users, queries run from SkyServer search tools are restricted in how long they can run and how much output they return, by **timeouts** and **row limits**. Please see the Query Limits help page. To run a query that is not restricted by a timeout or number of rows returned, please use the CasJobs batch query service.

Clear Query

```
-- This query does a table JOIN between the imaging (PhotoObj) and spectra
-- (SpecObj) tables and includes the necessary columns in the SELECT to upload
-- the results to the SAS (Science Archive Server) for FITS file retrieval.
SELECT TOP 10
   p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z,
   p.run, p.rerun, p.camcol, p.field,
   s.specobjid, s.class, s.z as redshift,
   s.plate, s.mjd, s.fiberid
FROM PhotoObj AS p
   JOIN SpecObj AS s ON s.bestobjid = p.objid
WHERE
   p.u BETWEEN 0 AND 19.6
   AND g BETWEEN 0 AND 20
```

Submit | ☐ Check Syntax Only? | Output Format: ◉ HTML ○ XML ○ CSV ○ JSON ○ VOTable ○ FITS | Reset

To find out more about the database schema use the Schema Browser.

For an introduction to the Structured Query Language (SQL), please see the Searching for Data How-To tutorial. In particular, please read the Optimizing Queries section.

The inclusion of the imaging and spectro columns for SAS upload in your query (as in the default query on this page) will ensure that when you press **Submit**, the appropriate button(s) are displayed on the query results page to allow you to upload the necessary information to the SAS to retrieve the FITS file data corresponding to your CAS query. The imaging columns needed for upload to the SAS are *run*, *rerun*, *camcol*, and *field*. The spectroscopic columns needed are *plate*, *mjd*, *fiberid*, and optionally *sprerun* (the latter requires a join with the PlateX table).

Batch Queries, MyDB

SDSS Query / CasJobs

Help | Tools | Query | History | MyDB | Import | Groups | Output | Schema Browser | Profile | Queues | SkyServer

MyDB | Local Only

Views
Tables
Functions
Procedures

Sort by... | All selected...

Rows	kB	Name
1	16	bar
1	16	bar2
1	16	course
42,086	1,032	CycleEdges
3,559,046	84,552	CycleEdges_SD1
157,289	9,416	Cycles_sd1
0	0	ExtrnImages
0	0	ExtrnKeyValue
3	16	ExtrnResources
3	16	ExtrnResourceTables
0	0	ExtrnTableMetadata
17	16	food4
237	24	halo_sample
102	16	halo_sample2
10	40	mydr12_table1
64	16	MyMRSnapshots
100	16	MyNewTable
10	40	MyTable
10	40	MyTableAgain
1,000	24	noCracks300_sd1
100	16	QuickResults
100	72	Results
100	72	Results_1
100	72	Results_2
254	24	StratHaloSample
0	0	T_mpahalo_csv_1_1
1,000	48	T_mr_csv_3_1
2,000	264	td_sample1
16	16	thumbnails
41,751	3,464	Thumbs
4	16	TSE01_TOC
50	16	TSE_R
17	16	TSE_TOC
4	16	tsecounts

thumbnails

Contains ~16 rows (~16 kB)

Notes | Sample | Job | Plot | Download | Publish | Rename | Drop

Table Schema type [size]

objId	ra	dec	petror90_r	fits_r	pubURL
bigint [8]	real [4]	real [4]	real [4]	varchar [128]	varchar [128]

objId	ra	dec	petror90_r	fits_r
1237657190905217071	0.01848758	-0.08335432	13.45019	http://dr12.sdss3.org/sas/dr
1237657190905217208	0.0325857	-0.04057997	11.8864	http://dr12.sdss3.org/sas/dr
1237652942101610636	0.04966388	14.09001	15.69221	http://dr12.sdss3.org/sas/dr
1237654382519779496	154.5588	58.99769	10.08662	http://dr12.sdss3.org/sas/dr
1237657192515829920	0.05327045	1.120216	16.45499	http://dr12.sdss3.org/sas/dr
1237657192515829922	0.04990565	1.121781	13.26728	http://dr12.sdss3.org/sas/dr
1237658300069249039	154.5637	7.038844	13.0831	http://dr12.sdss3.org/sas/dr
1237667536398516375	154.5617	20.66201	12.95956	http://dr12.sdss3.org/sas/dr
1237654604249563301	154.5744	3.34882	11.5982	http://dr12.sdss3.org/sas/dr
1237654605320028364	147.0676	3.905854	10.37041	http://dr12.sdss3.org/sas/dr
1237655370355245106	154.5736	59.20863	10.50707	http://dr12.sdss3.org/sas/dr
1237657589775859777	147.0741	47.71434	12.38772	http://dr12.sdss3.org/sas/dr
1237657770711187559	154.5731	53.39278	10.03565	http://dr12.sdss3.org/sas/dr
1237658300069249037	154.5821	7.049364	16.89148	http://dr12.sdss3.org/sas/dr
1237658613050179717	154.5945	45.95465	17.88555	http://dr12.sdss3.org/sas/dr
1237661064955297935	154.5768	10.6659	14.28678	http://dr12.sdss3.org/sas/dr

Cosmological Simulations

Virgo - Millennium Database

Documentation

CREDITS/Acknowledgments

Registration

News

FAQ

Public Databases
- DGalaxies
- DHaloTrees
- Guo2010a
- Guo2013a
- Henriques2012a
- Henriques2015a
- MField
- MillenniumII
- millimil
- miniMilII
- MMSnapshots
- MPAGalaxies
- MPAHaloTrees
- MPAMocks
- Snapshots

Private (MyDB) Databases
- AllSkyMaps (r)
- Aquarius (r)
- Galformod_db (r)
- galformodAdmin (r)
- GAVOFits (r)
- Guo2010aMocks (r)
- Lib (r)
- MillenniumTAP_db (r)
- MockLensing (r)
- MPEVO (r)
- MRObs (r)
- MRObs_db (r)
- MRXXL (r)
- overzier_db (r)
- Oxford (r)
- ROSAT (r)
- vipers_db (r)
- anusser_db (rw)
- bbozek_db (rw)
- gdmarleau_db (rw)
- gerard_db (rw) (context)
- jlee_db (rw)
- millenniumdbpaper_db (rw)
- SAMCompare_db (rw)
- sampling_db (rw)
- treedist_db (rw)
- TreeIndexes (rw)
- voids_db (rw)
- vvds_db (rw)
- wtwang_db (rw)

Welcome Gerard Lemson.
Streaming queries return unlimited number of rows in CSV format and are cancelled after 420 seconds.
Browser queries return maximum of 1000 rows in HTML format and are cancelled after 30 seconds.

There is a partial mirror of this database in Durham at http://galaxy-catalogue.dur.ac.uk:8080/Millennium/.
The Durham database does not contain all the latest L-Galaxies models but does contain more recent GALFORM models.

```
select PROG.*
  from millimil..MPAHalo PROG,
       millimil..MPAHalo DES
 where DES.haloId = 1
   and PROG.haloId between DES.haloId and DES.lastprogenitorId
```

Query (stream) | Query (browser) | Explain | Help

Maximum number of rows to return to the query form: 10

Previous queries:
List of all queries executed sofar in this session. Selecting a query will make it appear in the query window.
The link will show all of them in a separate window. Refreshing that window will load the latest queries again.

select PROG.* from millimil..MPAHalo PROG, millimil..MPAHalo DES \ Show all previous queries

Demo queries: click a button and the query will show in the query window.
Holding the mouse over the button will give a short explanation of the goal of the query. These queries are described in some more detail on th

Mainly Halos: H 1 | H 2 | H 3 | H 4 | H 5 | HF 1 | HF 2 | HF 3

Mainly Galaxies: G 1 | G 2 | G 3 | G 4 | G 5 | G 6 | HG 1 | HG 2 | GF 2

Metadata queries: The SQL statements under these buttons provide examples for querying and managing the state of a private database.
Holding the mouse over the button will give a short explanation of the goal of the statement.

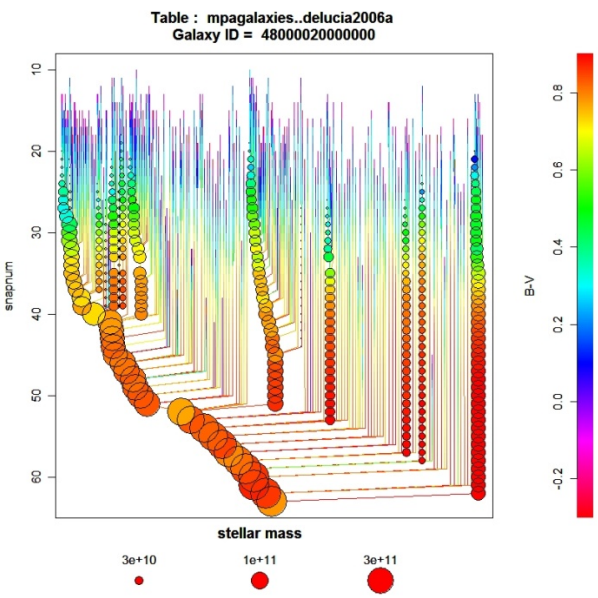
ShowTables | Show Views | Show Columns | Show Indexes | MyDB Size | MyDB Table Size

Create View | Drop Table | Create Index

Query time (in millisec) = 121
Number of rows retrieved from database = 1000

Reformat | CSV | Plot (VOPlot)

haloID	subHaloID	lastProgenitorId	treeId	snapNum	redshift	firstProgenitorId	nextProgenitorId	descendantId	firstHaloInFOFgroupId	nextHaloInFOFgro
1	62000000000000	2444	0	62	0.019932542	2	2445	0	1	2495
2	61000000000000	2294	0	61	0.041403063	3	2295	1	2	2496
3	60000000000000	2291	0	60	0.064493395	4	2292	2	3	2497
4	59000000000000	2222	0	59	0.08928783	5	2223	3	4	2653
5	58000000000000	2222	0	58	0.11588337	6	-1	4	5	2654
6	57000000000000	2182	0	57	0.14438343	7	2183	5	6	2655
7	56000000000000	2120	0	56	0.17489761	8	2121	6	7	2656
8	55000000000000	2074	0	55	0.20754863	9	2075	7	8	2657
9	54000000000000	2033	0	54	0.24246909	10	2034	8	9	2071



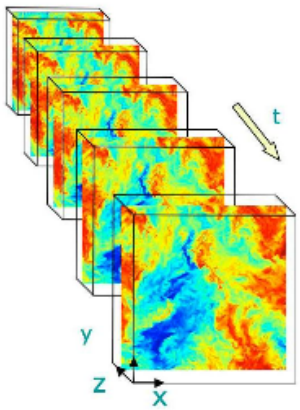
Johns Hopkins Turbulence Databases

Home | Database Access | Documentation | Links | Visualizations

Turbulence Database

NOTICE: All systems are currently functional. For past announcements, please click here

Welcome to the Johns Hopkins Turbulence Database (JHTDB) s



This website is a portal to an Open Numerical Turbulence Laboratory that enables access to multi-Terabyte turbulence databases. The data reside on several nodes and disks on our database cluster computers and are stored in small 3D subcubes. Positions are indexed using a Z-curve for efficient access.

Access to the data is facilitated by a Web services interface that permits numerical experiments to be run across the Internet. We offer C, Fortran and Matlab interfaces layered above Web services so that scientists can use familiar programming tools on their client platforms. Calls to fetch subsets of the data can be made directly from within a program being executed on the client's platform. Manual queries for data at individual points and times via web-browser are also supported. Evaluation of velocity and pressure at arbitrary points and time is supported using interpolations executed on the database nodes. Spatial differentiation using various order approximations (up to 8th order) and filtering are also supported (for details, see documentation page). Particle tracking can be performed both forward and backward in time using a second order accurate Runge-Kutta integration scheme. Subsets of the data can be downloaded in hdf5 file format using the data cutout service.

To date the Web-services-accessible databases contain a space-time history of a direct numerical simulation (DNS) of isotropic turbulent flow, in incompressible fluid in 3D, a DNS of the incompressible magneto-hydrodynamic (MHD) equations, a DNS of forced, fully developed turbulent channel flow, and a DNS of homogeneous buoyancy driven turbulence. The datasets comprise over 20 Terabytes for the isotropic turbulence data, 56 Terabytes for the MHD data, 130 Terabytes for the channel flow data and 27 Terabytes for the homogeneous buoyancy driven turbulence data. Basic characteristics of the data sets can be found in the datasets description page. Technical details about the database techniques used for this project are described in the publications.

The JHTDB project is funded by the US National Science Foundation.

Questions and comments? turbulence@pha.jhu.edu

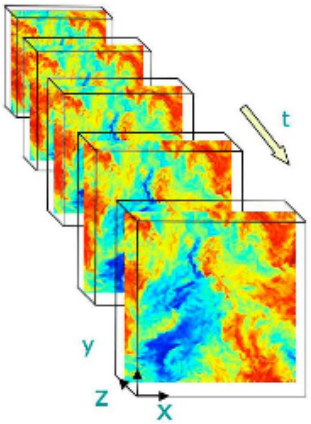
19,637,229,306,181 points queried

Please excuse our dust as we continue to develop this site. JHTDB is on-line but may periodcally be unavailable as we continue to add functionalities.

Johns Hopkins Turbulence Databases

Home | Database Access | Documentation | Links | Visualizations

Using JHTDB with Python



Installation

ubuntu 14.04 Bare-bone installation:

```
sudo apt-get install build-essential gfortran
sudo apt-get install python-setuptools
sudo apt-get install python-dev
sudo easy_install numpy
sudo python setup.py install
```

Note that doing this should, in principle, also install sympy on your system, since it's used by pyJHTDB. Happy fun installation:

```
sudo apt-get install build-essential gfortran
sudo apt-get install python-setuptools
sudo apt-get install python-dev
sudo apt-get install libpng-dev libfreetype6-dev
sudo apt-get install libhdf5-dev
sudo easy_install numpy
sudo easy_install h5py
sudo easy_install matplotlib
sudo python setup.py install
```

Web Service Access through Python

More information and source code can be found on github at https://github.com/idies/pyJHTDB

Last update: 8/28/2015 11:01:00 AM

Where Are We Going?

- Interactive science on petascale data
- Sustain and enhance our astronomy effort
- Create scalable open numerical laboratories
- Scale system to many petabytes
- Deep integration with the "Long Tail"
- Large footprint across many disciplines
  - Also: Genomics, Oceanography, Materials Science
- Use commonly shared building blocks
- Major national and international impact

New Model

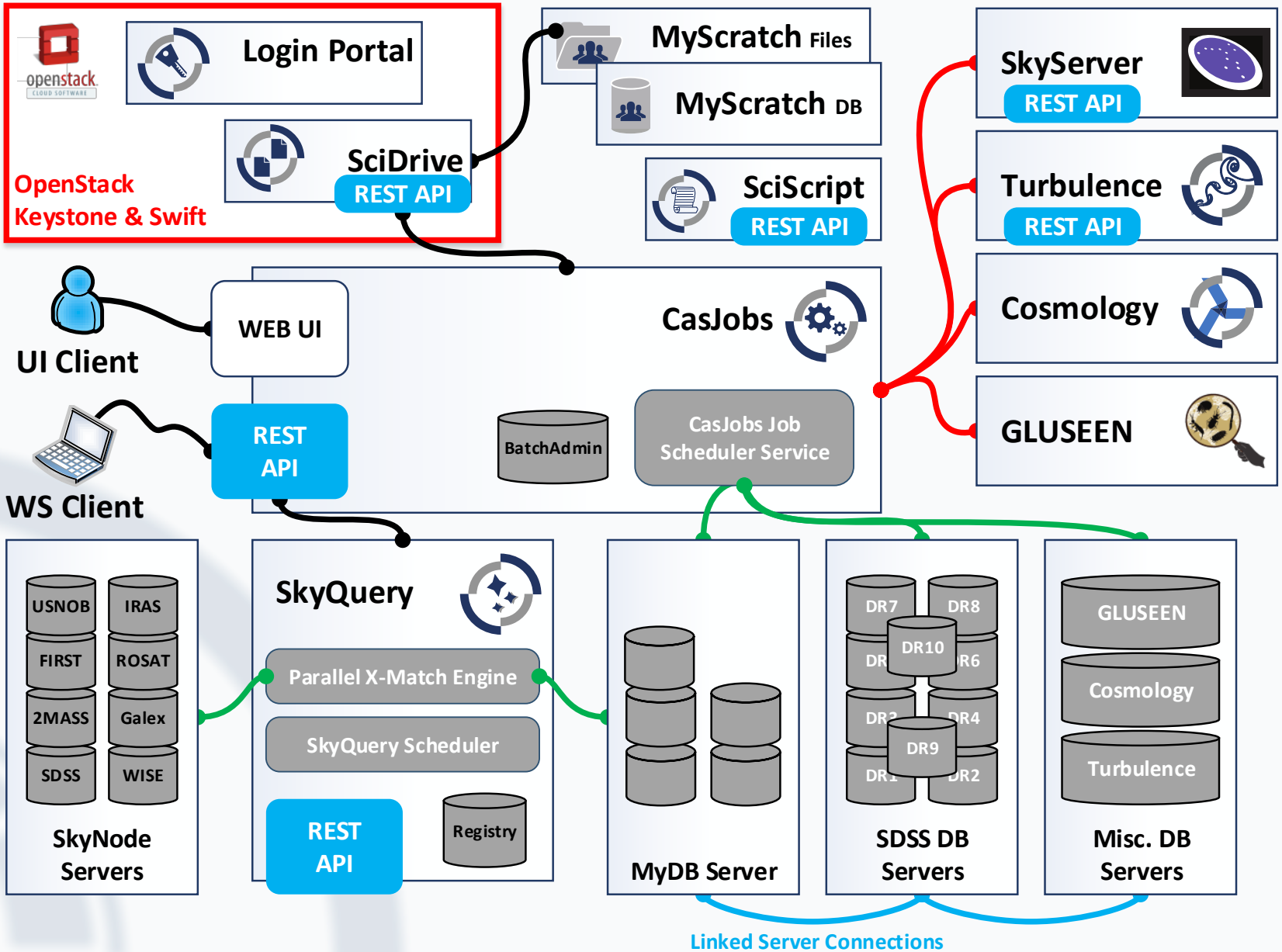
- Offer more computing resources server side
- Augment and combine SQL queries with easy-to-use scripting tools
- Heavy use of virtual machines
- Interactive portal via iPython/Matlab/R
- Batch jobs
- Enhanced visualization tools

Main Components

- CasJobs
  - SQL, MyDB, batch
  - FileDB: Raw data access from within RDB
- SciDrive
  - Dropbox-like, on-drop event handling
- SciServer/compute
  - Interactive/batch python, R, Matlab in Docker container
- MyScratch (File & DB)
- SSO on all components
- All published through REST

Collaborative data-driven science

SciServer

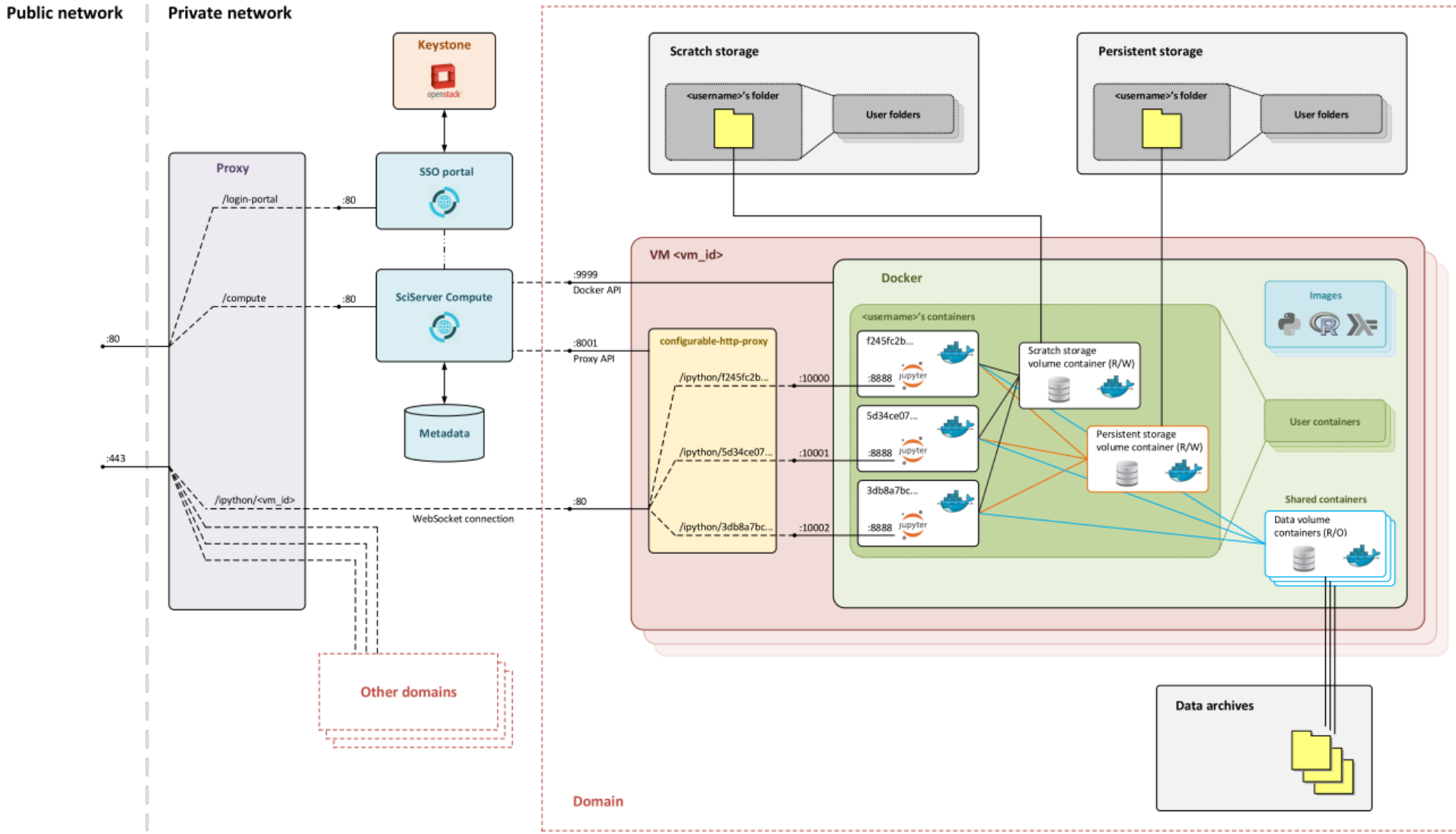


Latest: SciServer-Compute

- Jupyter Notebooks in Docker
  - http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261
  - https://developer.rackspace.com/blog/how-did-we-serve-more-than-20000-ipython-notebooks-for-nature/
- Python, R, Matlab
- Flexible way to attach data sets in volume containers
- Extended to batch jobs

Collaborative data-driven science

SciServer



jupyter 1. SDSS-Thumbnails Last Checkpoint: 03/17/2016 (unsaved changes)

File Edit View Insert Cell Kernel Help

Code CellToolbar

3. Query an astronomy database (SDSS/DR12)

Write SQL statement and send it to CasJobs' REST API. Uses synchronous mode as the query is quite small. asynch mode is av queueu and the result will be stored in a table in MyDB or MyScratch/DB.

TODO make example with batch query mode.

In [4]:

```
# query obtained from SkyServer interface
# Queries the Sloan Digital Sky Serveys' Data Release 12.
# For schema and documentation see http://skyserver.sdss.org
#
# This query finds galaxies in the SDSS database that have a spectrum taken and which have a si
# larger than 10 arcsec.
# We return
query="""
SELECT TOP 16 p.objId,p.ra,p.dec,p.petror90_r
  FROM galaxy AS p
   JOIN SpecObj AS s ON s.bestobjid = p.objid
WHERE p.u BETWEEN 0 AND 19.6
  AND p.g BETWEEN 0 AND 17
  AND p.petror90_r > 10
"""

# query CasJobs table. Using DR12 as context. I.e. a connection is made to DR12 when running th
queryResponse = CasJobs.executeQuery(query, "dr12",token=token)
# parse results into pandas.DataFrame for further in memory processing
gals = pandas.read_csv(queryResponse,index_col=0)
```

```
executeQuery POST response:  200 OK
```

In [5]:

```
# show the table
gals
```

Out[5]:

	ra	dec	petror90_r
objId			
1237645941824356443	40.285690	-0.714957	14.72862
1237645943978393694	55.123625	0.872487	18.85110
1237645943979311221	57.248385	0.925979	14.60275
1237645943979114622	56.847420	0.875488	15.66479
1237646588246688297	80.135126	1.186679	11.14294
1237648672922337955	245.700633	-0.844301	10.51866
1237648673459667234	246.782081	-0.492432	11.98310

Astronomy

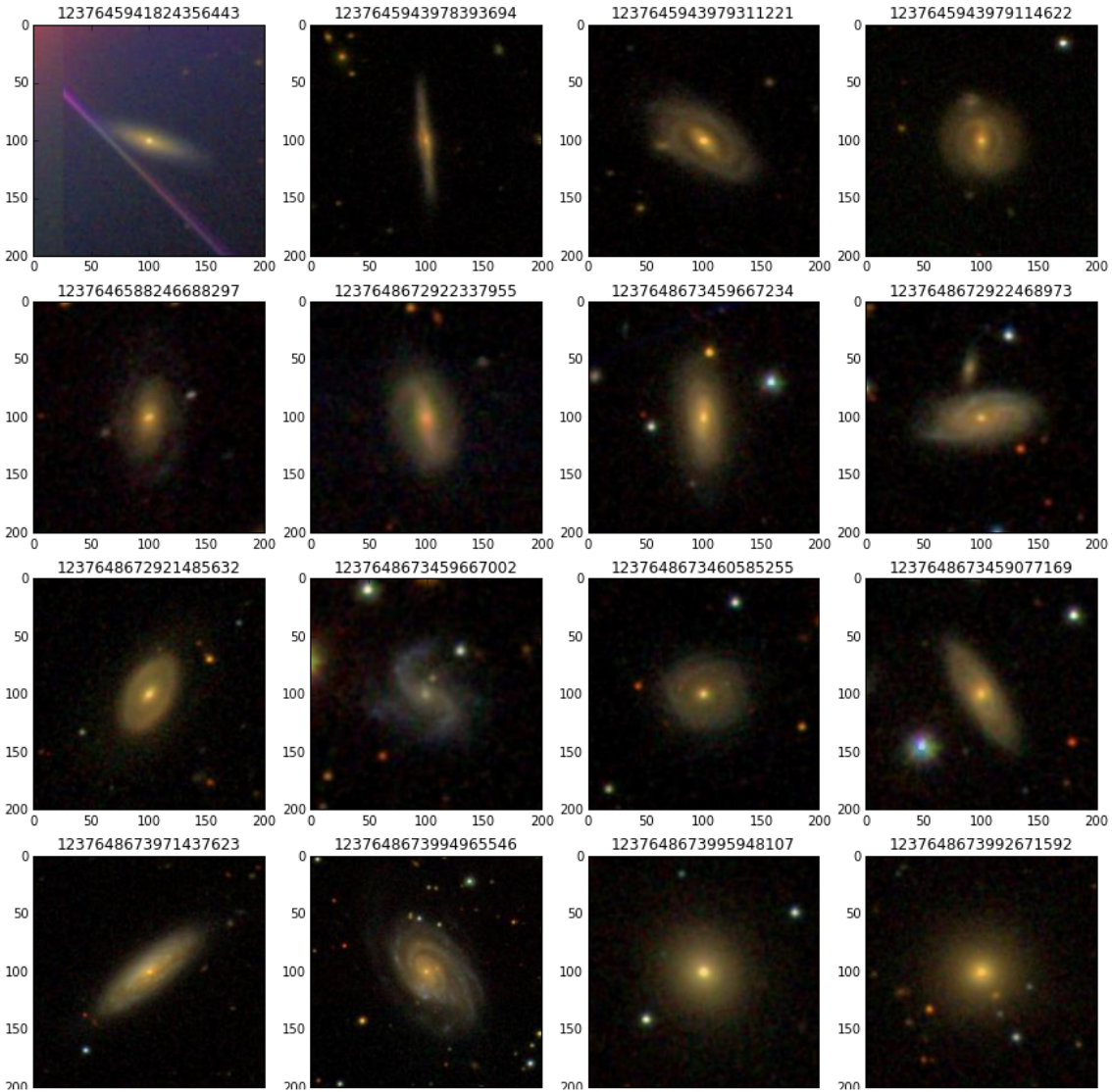
Collaborative data-driven science

SciServer

jupyter 1. SDSS-Thumbnails Last Checkpoint: 03/17/2016 (unsaved changes)

File Edit View Insert Cell Kernel Help

Code CellToolbar



Collaborative data-driven science

SciServer

jupyter FragData - Analysis I Last Checkpoint: 40 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Help Python 3

Markdown Cell Toolbar: None

FragData - Analysis I: Plotting fields and cracks

Most complex analysis and visualization is performed in python. This notebook contains relevant code.

Plotting a snapshot

In [1]:
```
# standard first block for defining the token and makinhg it available as a system variable for the session
# token must be replaced with new one once it has expired
with open('/home/idies/keystone.token', 'r') as f:
    token = f.read().rstrip('\n')
```

In [2]:
```
import SciServer.CasJobs
import pandas
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.collections import PolyCollection
from matplotlib.collections import LineCollection
import datetime
```

In [4]:
```
# query elements of a particular snapshot that have overlap with a given region
snapnum=200
suffix="_SD1"
xmin=0.00005
xmax=0.00015
ymin=0.00005
ymax=0.00015

query="""
declare @xmin float="""+str(xmin)+""", @xmax float = """+str(xmax)+"""
declare @ymin float="""+str(ymin)+""", @ymax float = """+str(ymax)+"""
declare @ixmin integer, @ixmax integer, @iymin integer, @iymax integer
select @ixmin=floor((@xmin-xmin)/dx)
,      @ixmax=floor((@xmax-xmin)/dx)
,      @iymin=floor((@ymin-ymin)/dy)
,      @iymax=floor((@ymax-ymin)/dy)
  from Simulations
 where suffix='"""+suffix+"""'
;
select e.snapnum, e.elementid
,    e.x1 ,e.y1
,    e.x2 ,e.y2
,    e.x3 ,e.y3
,    e.s0,e.s1,e.s2,e.s3
,    0.5*(e.s0+e.s3) as t, e.s0*e.s3-e.s1*e.s2 as d
  from cells"""+suffix+""" c
  ,    elemstress"""+suffix+""" e
 where c.snapnum="""+str(snapnum)+"""
   and c.ixmax between @ixmin and @ixmax
   and c.iymax between @iymin and @iymax
   and e.snapnum=c.snapnum
   and e.elementId=c.elementId
"""
# query CasJobs table. Using FragData as context
t0=datetime.datetime.now()
queryResponse = SciServer.CasJobs.executeQuery(query, "FragData")
t1=datetime.datetime.now()
# parse results into pandas.DataFrame for further in memory processing
elements = pandas.read_csv(queryResponse,index_col=None)
t2=datetime.datetime.now()
print("Found "+str(elements.count()[0])+" rows. query time: "+str(t1-t0)+" / parse time: "+str(t2-t1))
```

```
executeQuery POST response:  200 OK
Found 5133 rows. query time: 0:00:01.659539 / parse time: 0:00:00.046810
```

In [5]:
```
# prepare for buiding polygons
```

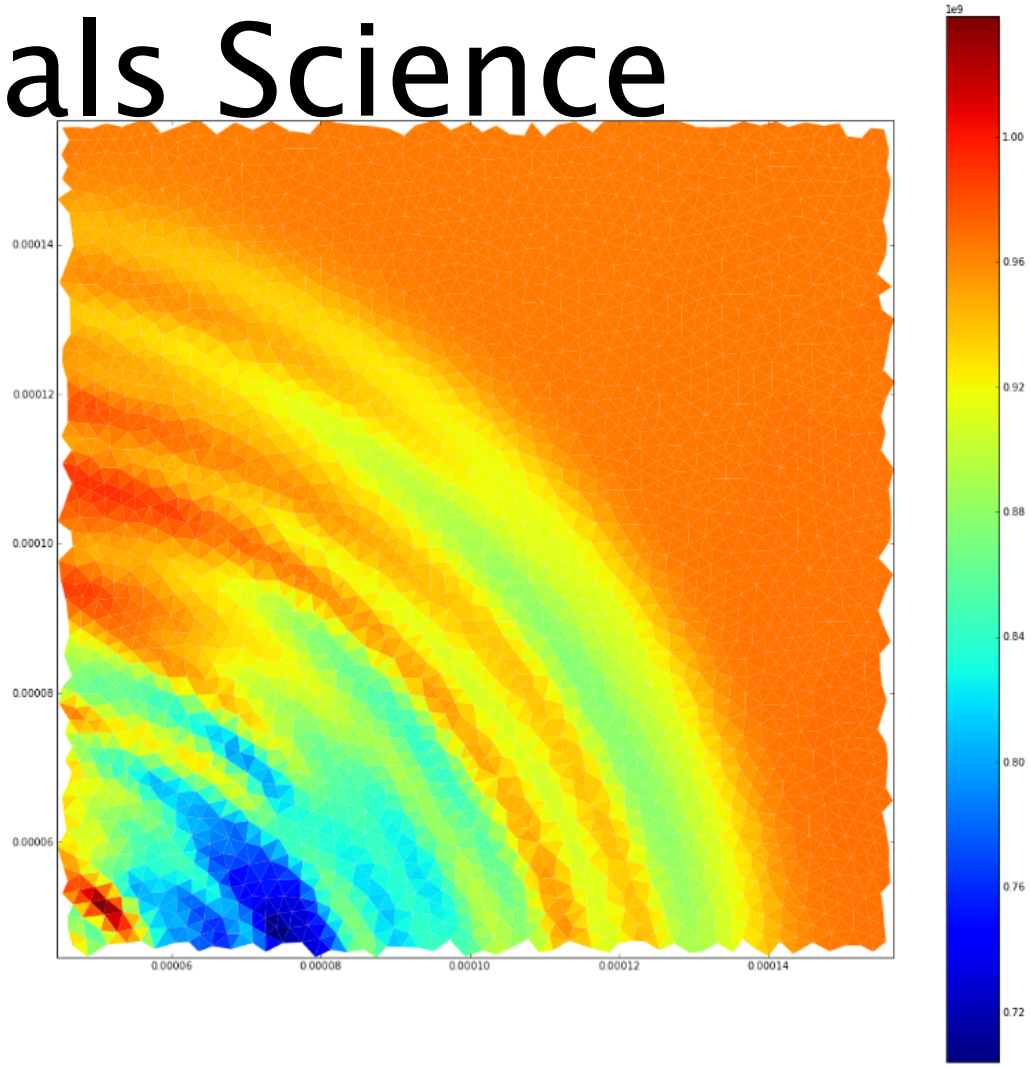
Materials Science

jupyter FragData - Analysis I Last Checkpoint: 5 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

Markdown Cell Toolbar: None

```
plt.ylim(min(elements.y1),max(elements.y1))
# Add a colorbar for the PolyCollection
fig.colorbar(coll, ax=ax)
plt.show()
```



Materials Science

jupyter FragData - Analysis | Last Checkpoint: 7 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help Python 3

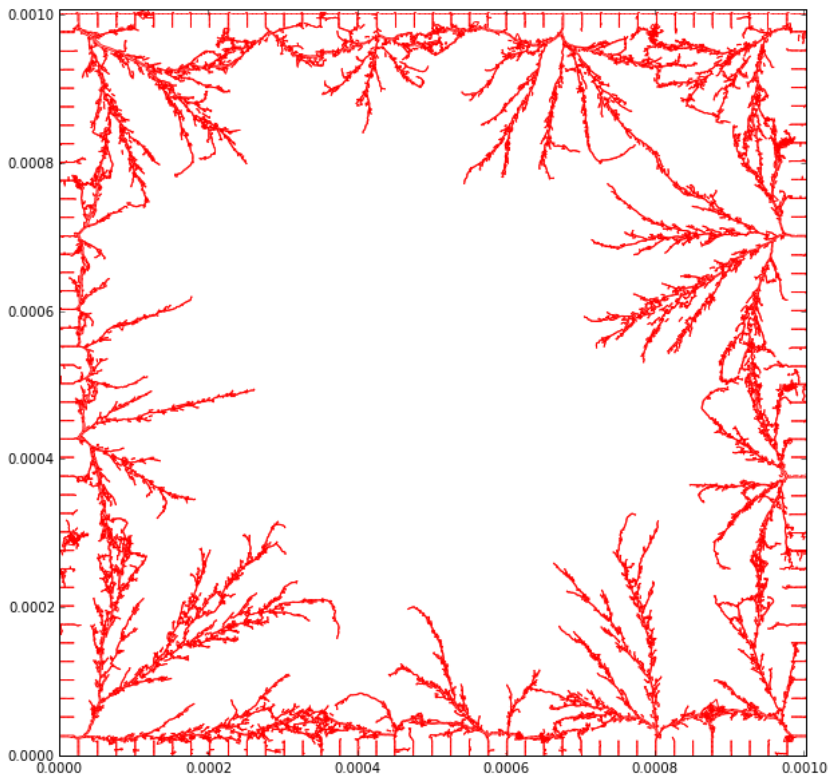
Markdown Cell Toolbar: None

```
l2=list(zip(cracks.x2,cracks.y2))
lines = list(zip(l1,l2))
```

```
In [53]: fig, ax = plt.subplots()
         fig.set_size_inches(10,10,forward=True)
         lc = LineCollection(lines, linewidths=1,color='red')
         ax.add_collection(lc)

         plt.xlim(0,max(cracks.x1))
         plt.ylim(0,max(cracks.y1))
```

```
Out[53]: (0, 0.001008152)
```



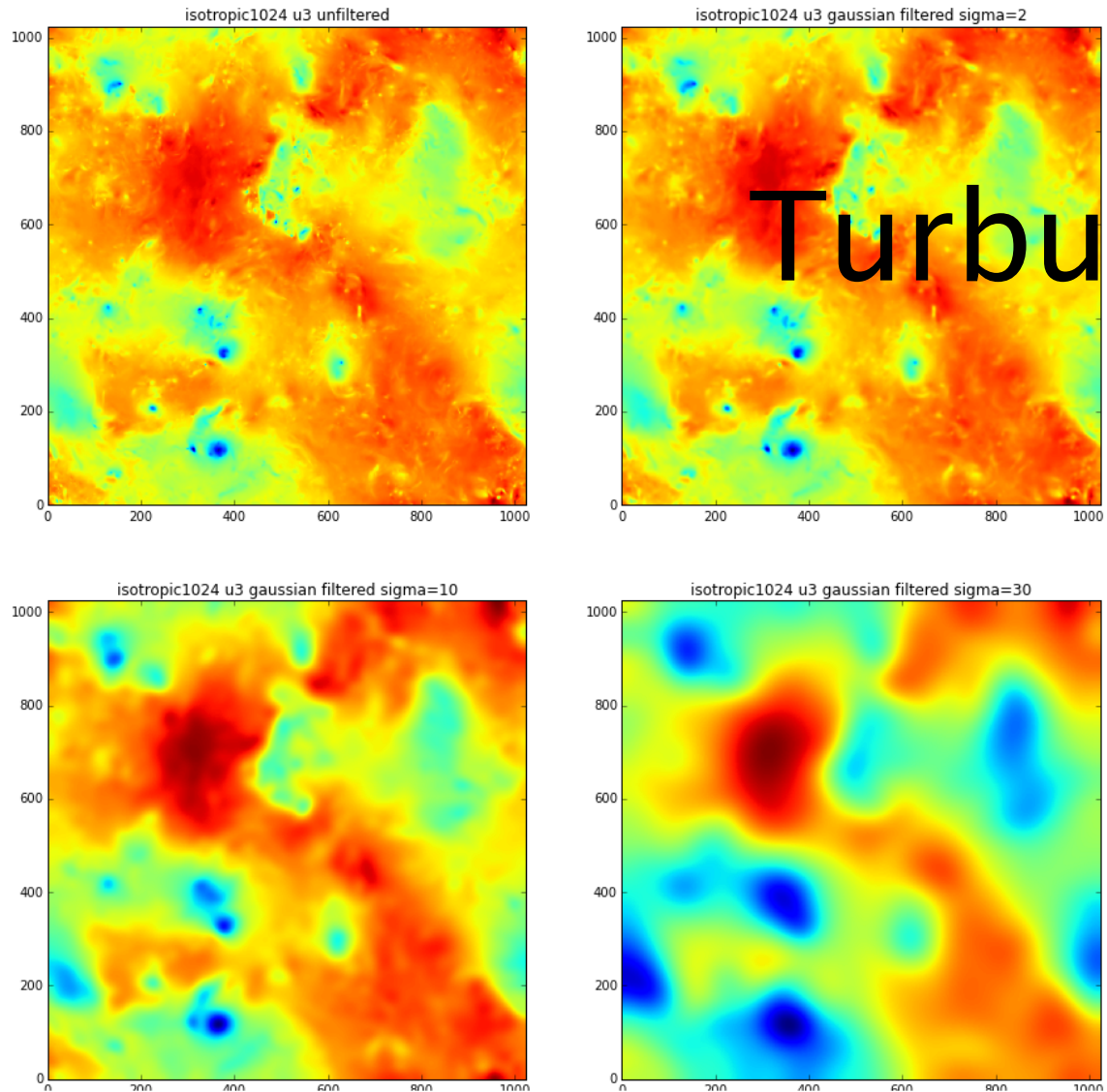
Collaborative data-driven science

SciServer

jupyter TurbUC_1_Velgaussianfilters_Plot-alt-Copy1 Last Checkpoint: 11 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help

Markdown Cell Toolbar: None



Turbulence

Collaborative data-driven science

SciServer

jupyter TSE Last Checkpoint: 10/29/2015 (unsaved changes)

File Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

```
order by pos
"""
queryResponse=SciServer.CasJobs.executeQuery(query,"TSE01",token=token)
_hist=pandas.read_csv(queryResponse)
```

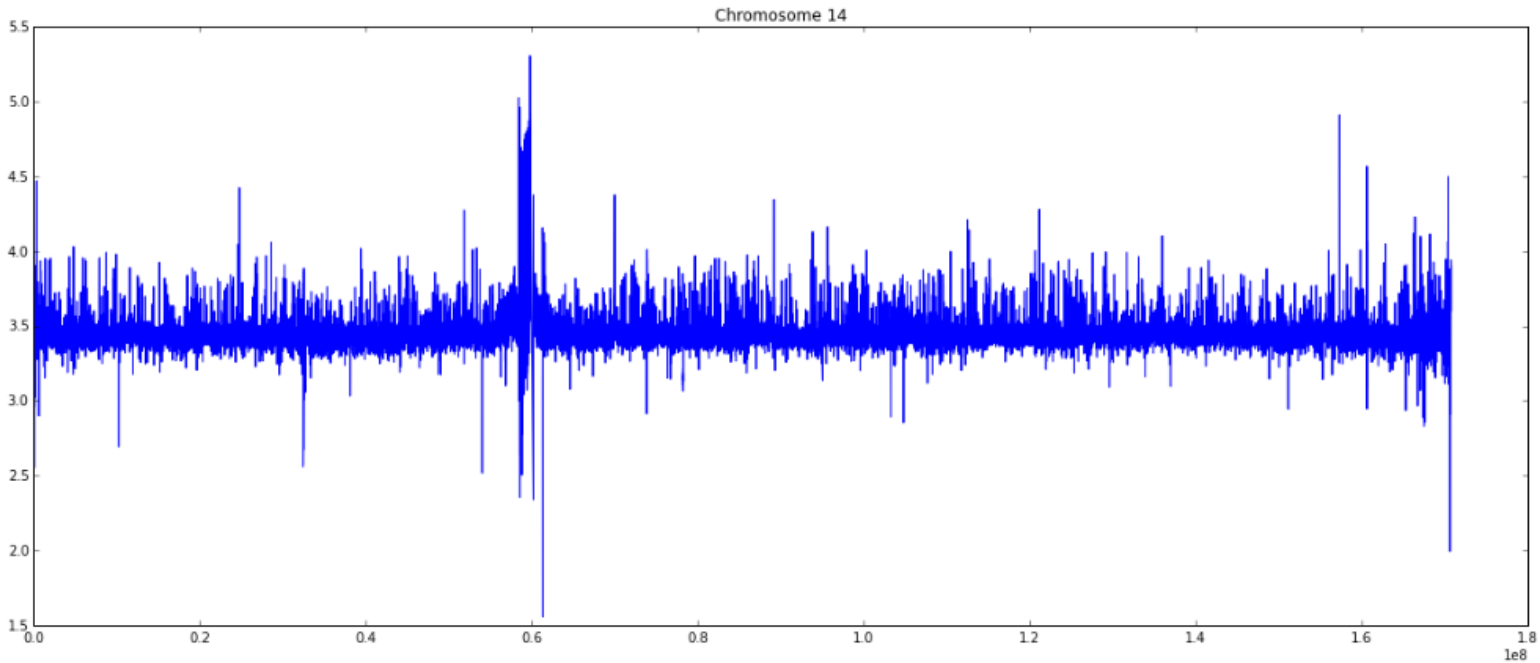
```
executeQuery POST response:  200 OK
```

In [26]:

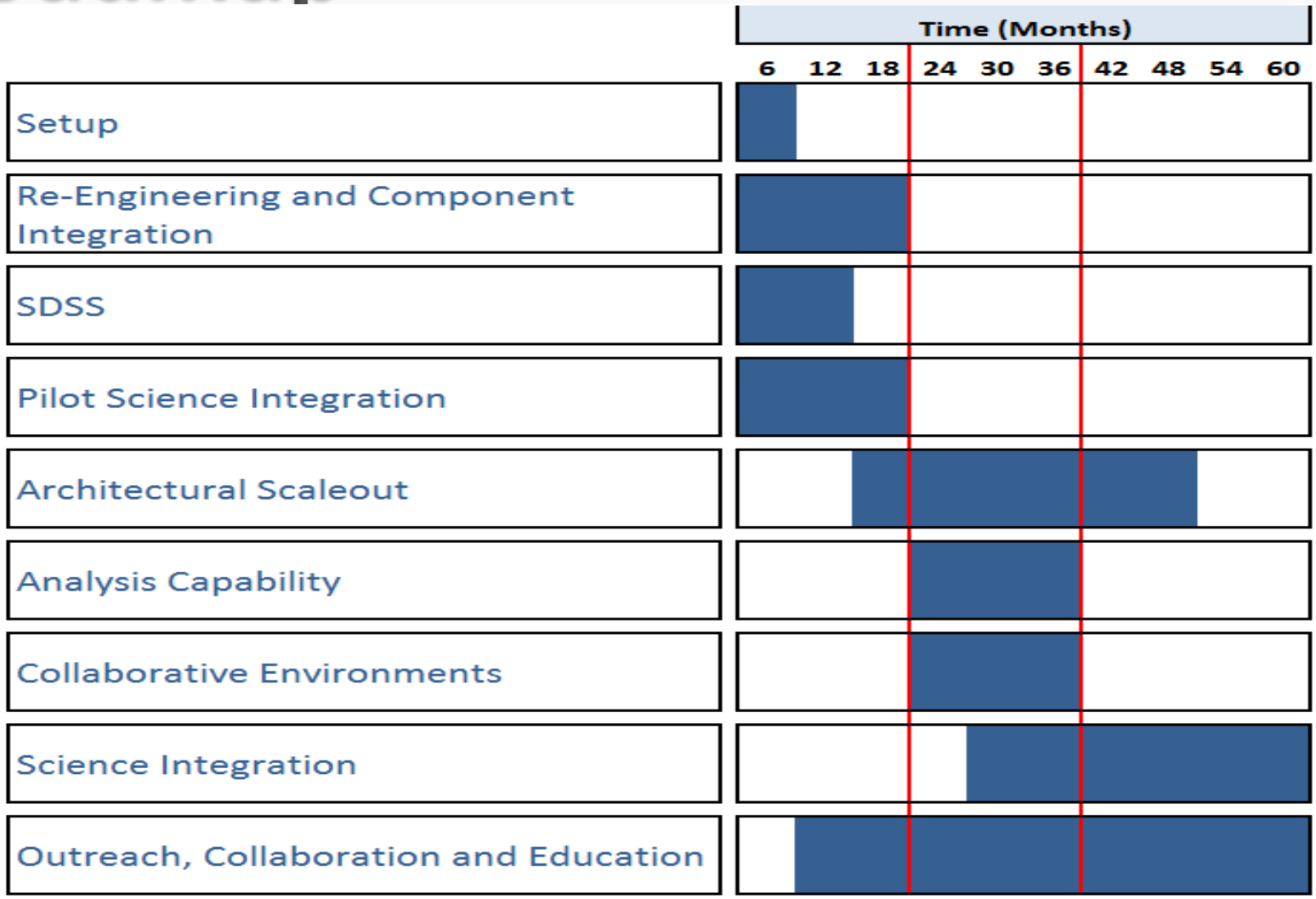
```
plt.figure(figsize=(20,8))
plt.title('Chromosome 14')

# restrict to small sample for demo purposes
plt.plot(_hist.pos,np.log10(_hist.num))
plt.show()
```

Genomics



Roadmap


Time (Months)
6 12 18 24 30 36 42 48 54 60
Setup
Re-Engineering and Component Integration
SDSS
Pilot Science Integration
Architectural Scaleout
Analysis Capability
Collaborative Environments
Science Integration
Outreach, Collaboration and Education

SciServer

Collaborative data-driven science

Thank you.

I'll be very happy to demo and discuss our services

JOHNS HOPKINS UNIVERSITY

NSF

idies