

SciServer



Our work on Databases and SciServer

Gerard Lemson, Alex Szalay
IDIES, Johns Hopkins University

sciserver.org



JOHNS HOPKINS
UNIVERSITY



idies

Data dissemination with Relational Databases

- ▶ Sloan Digital Sky Survey (SDSS)
- ▶ *Millennium* Cosmological Simulations
 - Time evolution in “merger trees”, 3D spatial queries
 - Custom SQL, MyDBs

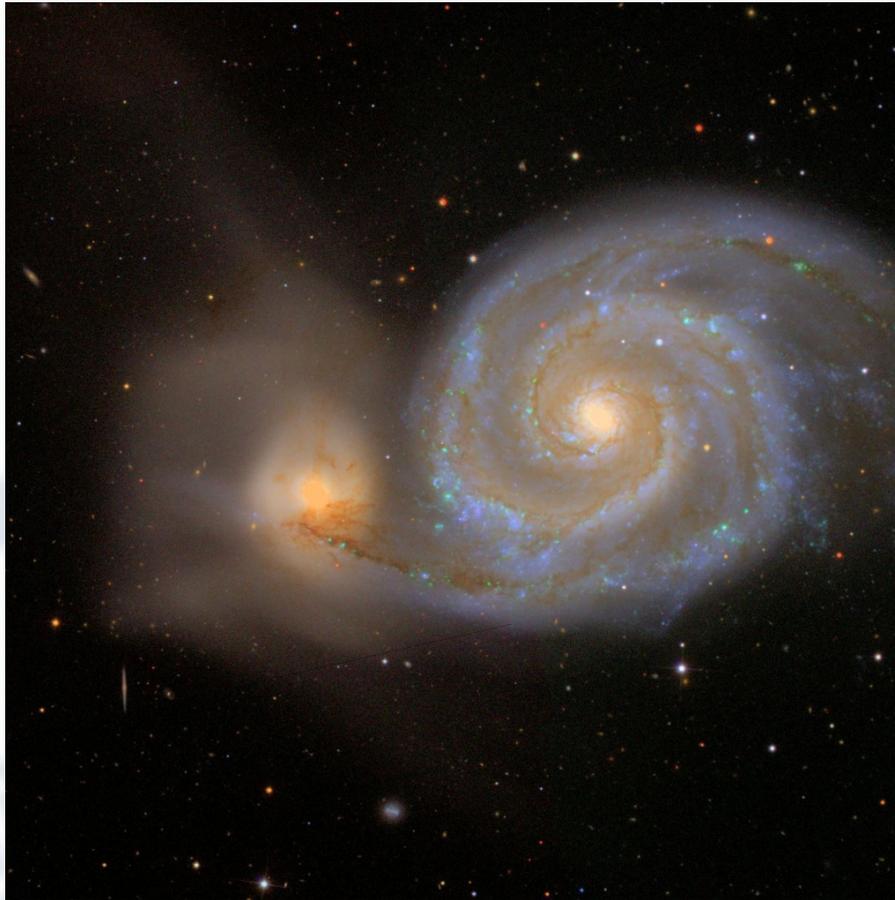


Image Credit: Sloan Digital Sky Survey

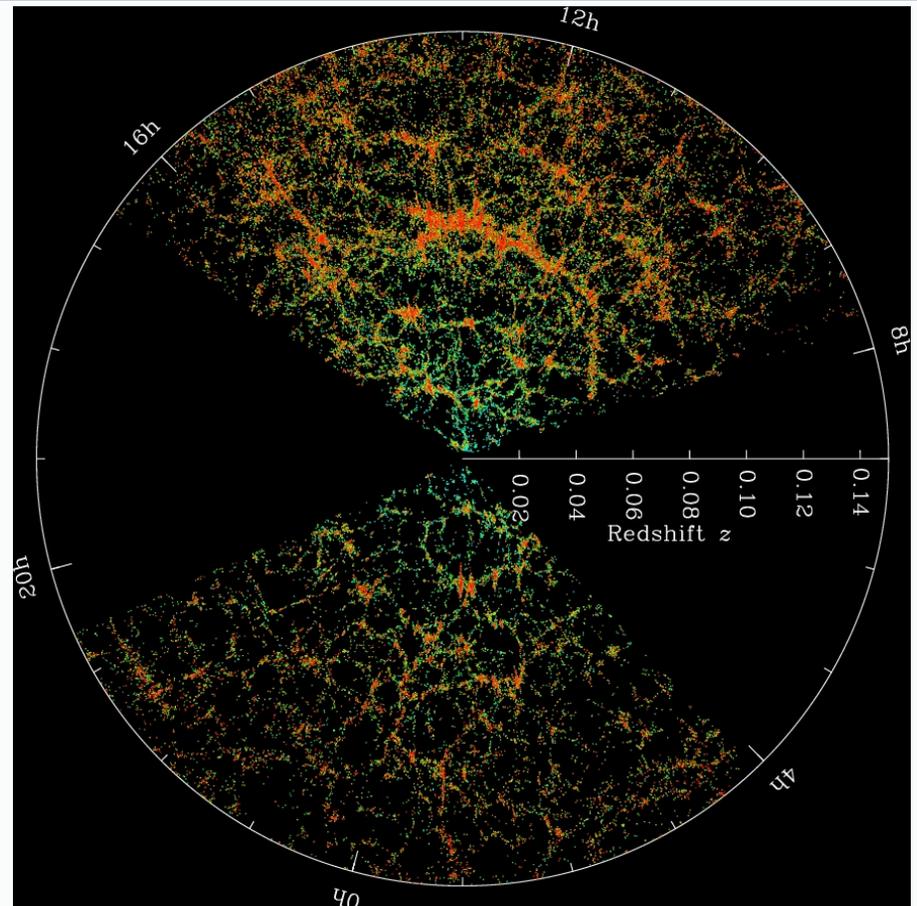
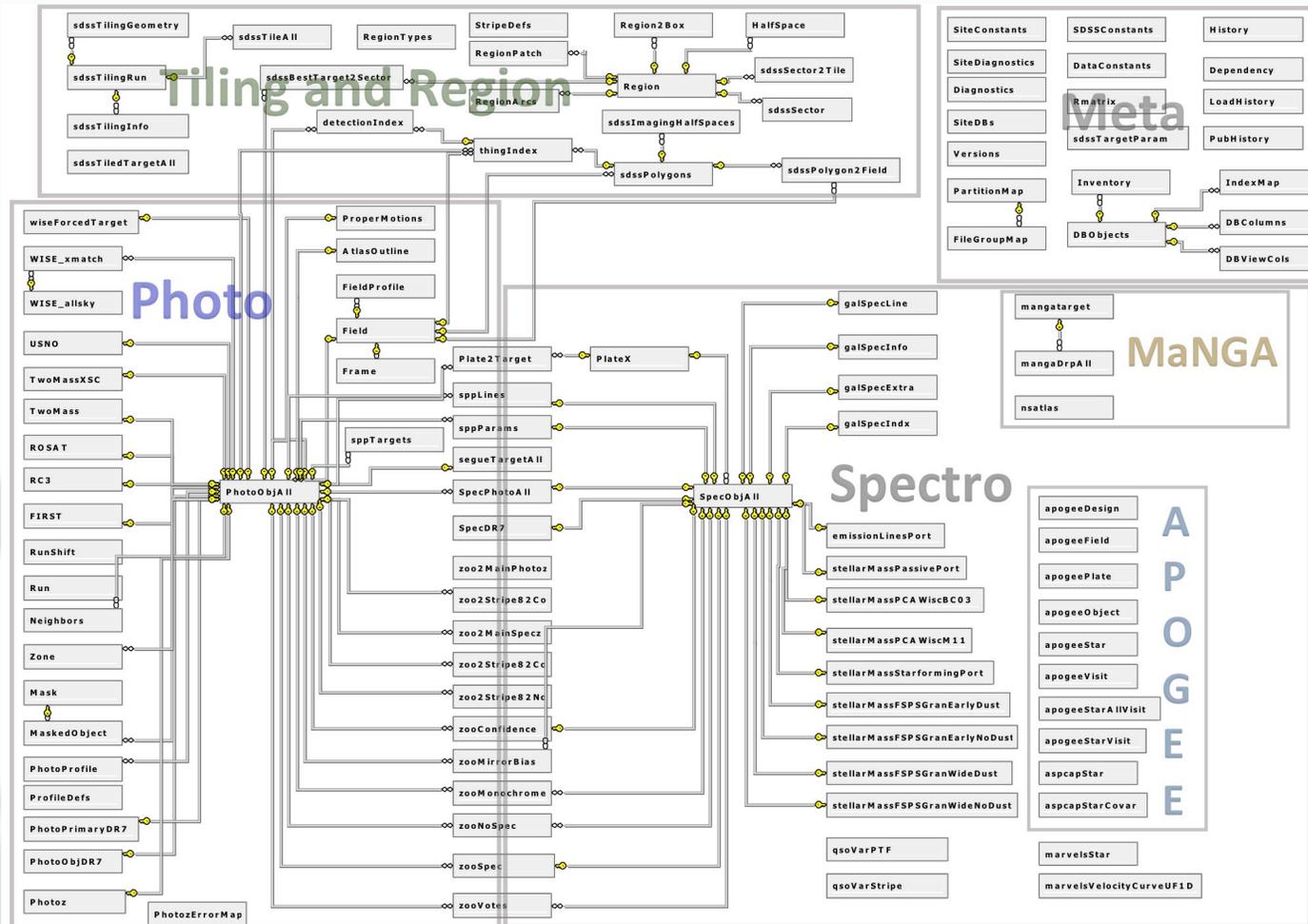


Image Credit: M. Blanton and SDSS



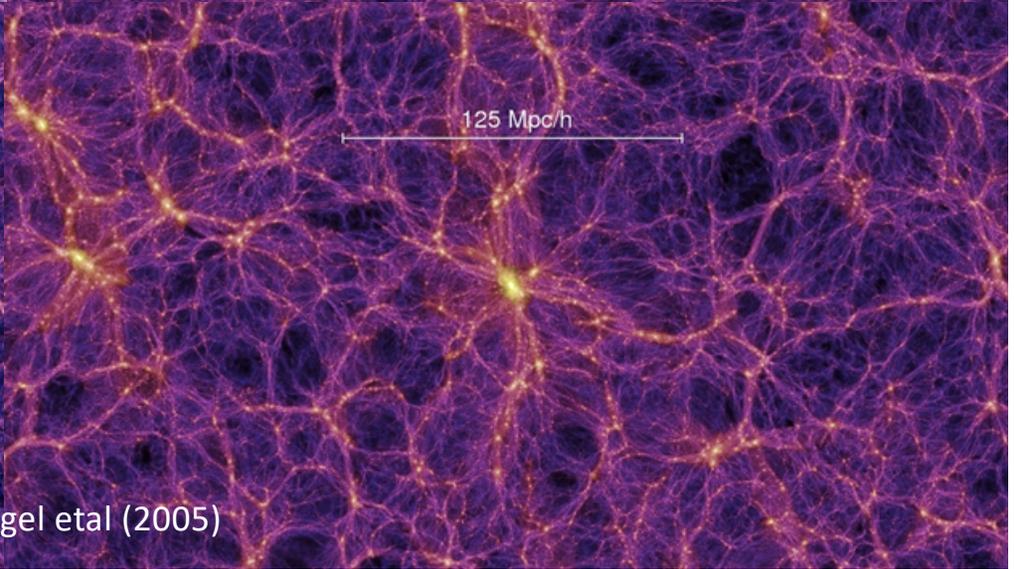
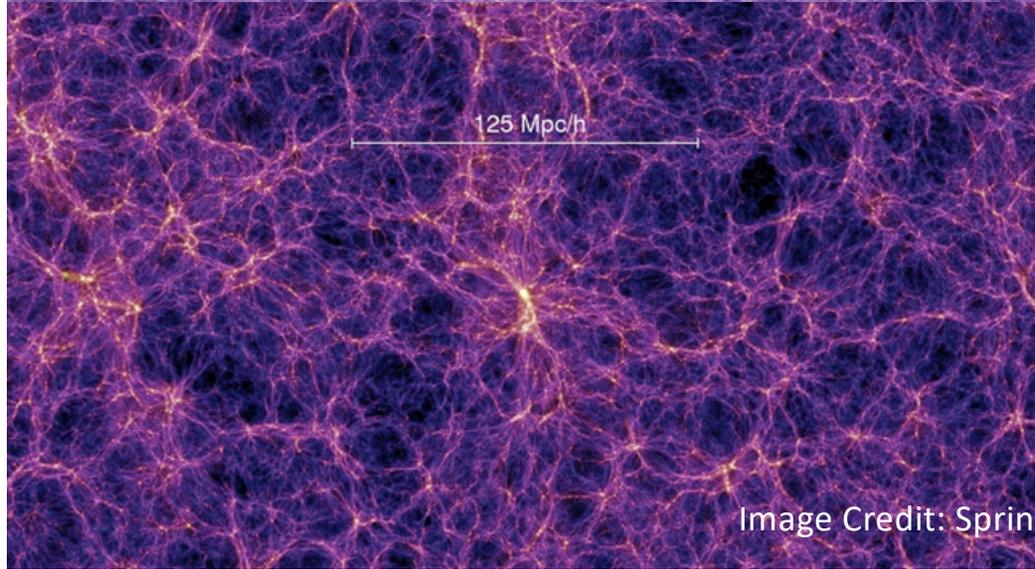
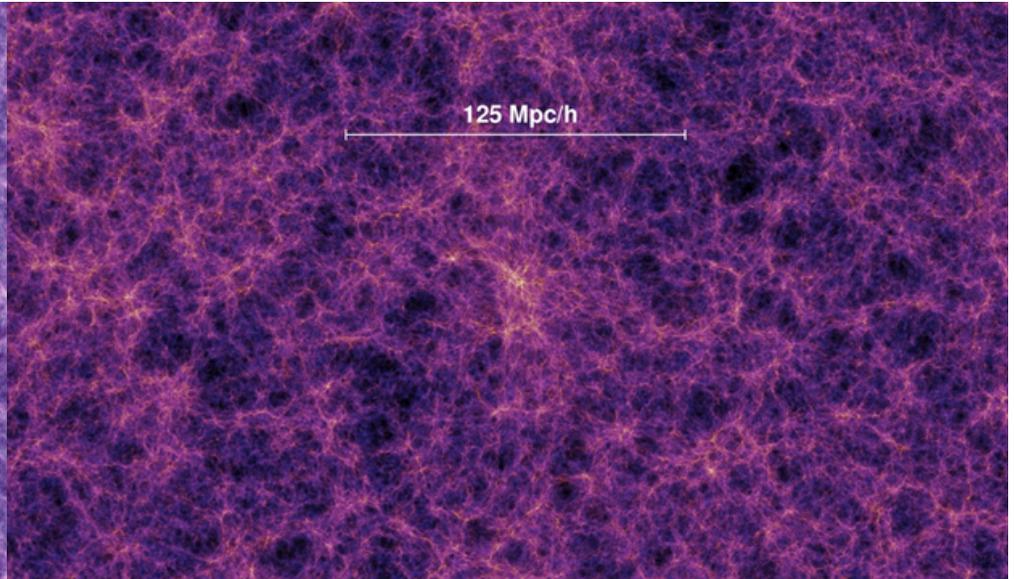
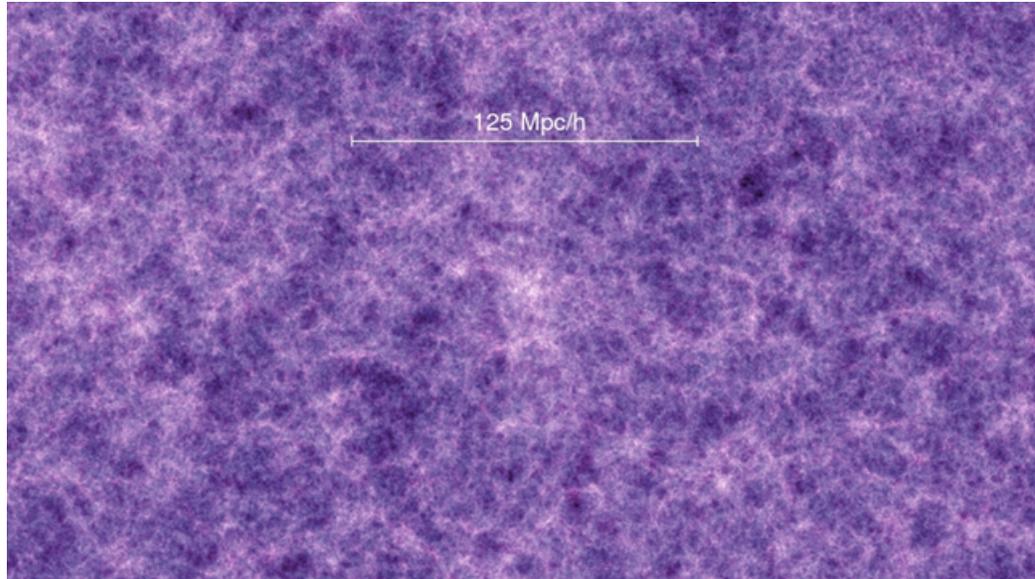
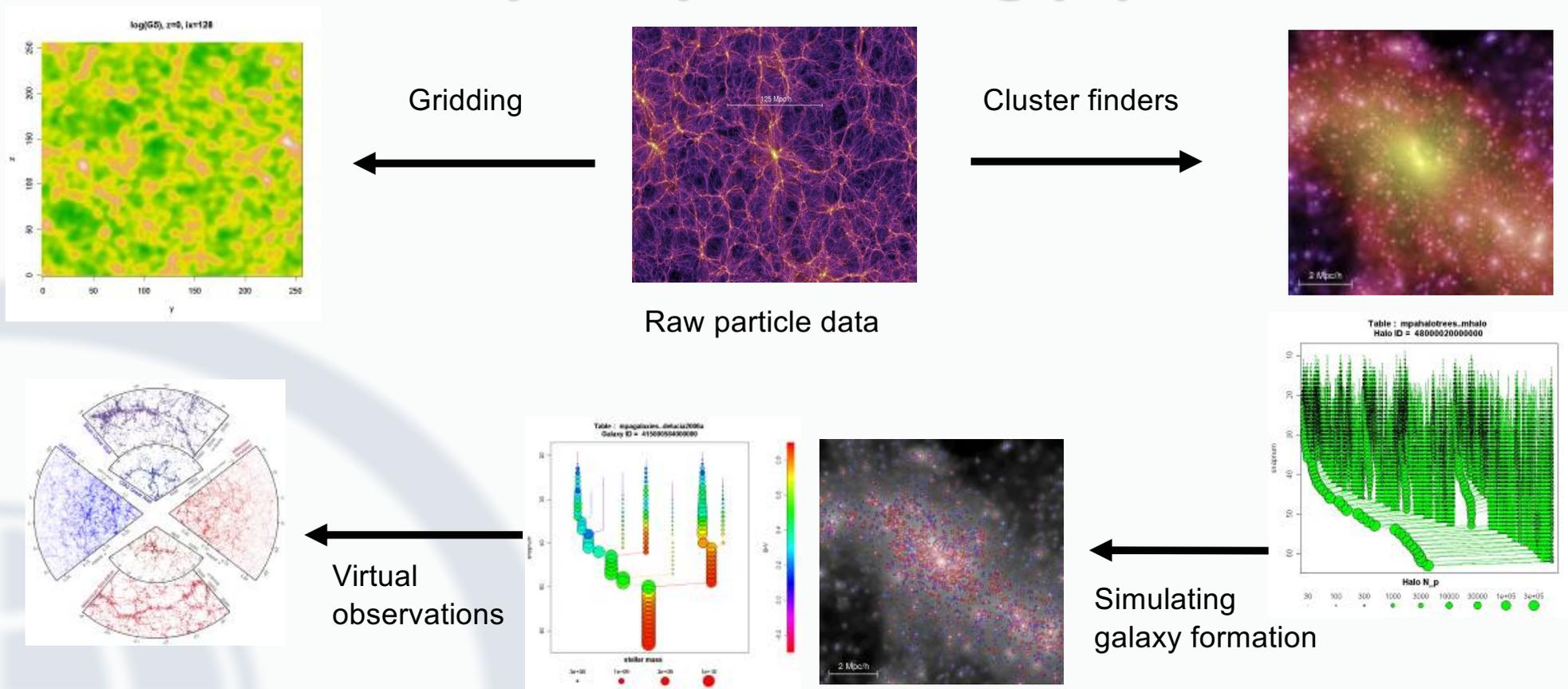
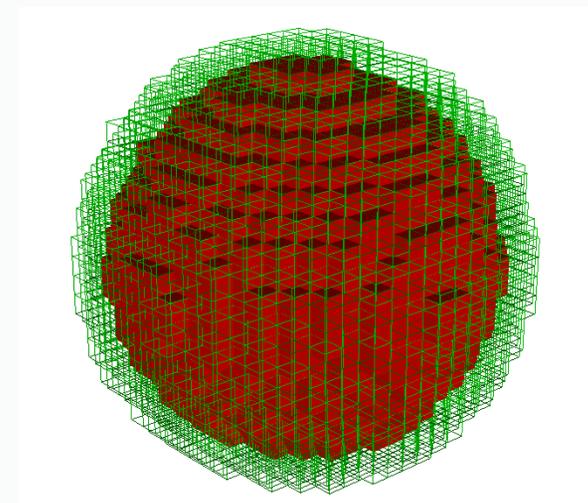
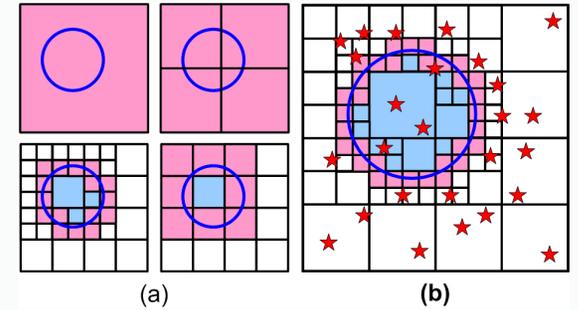
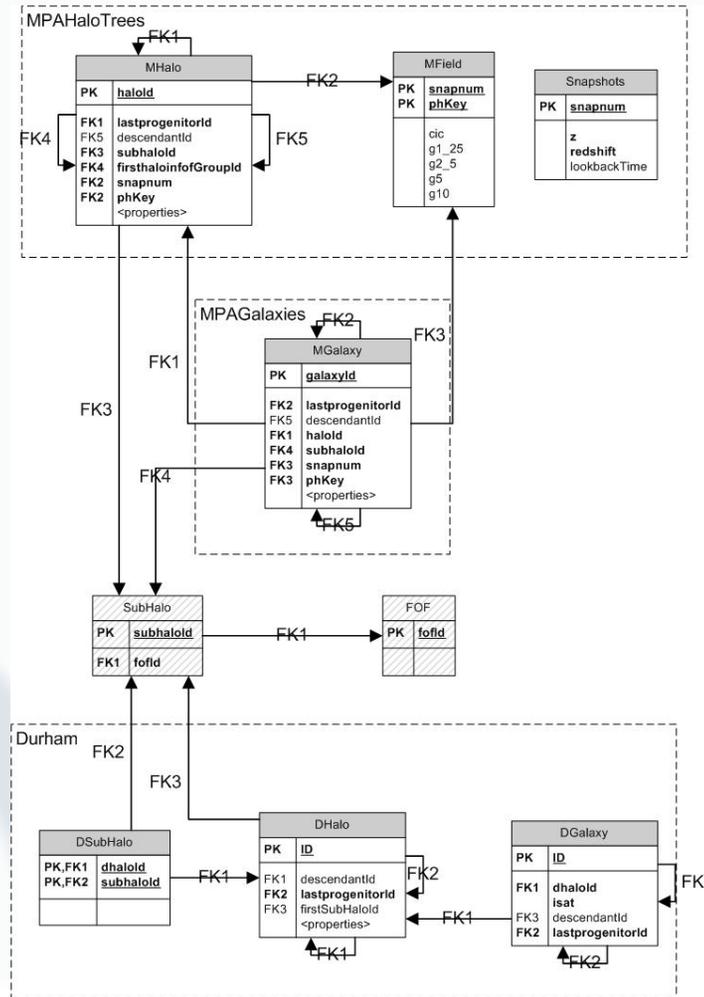
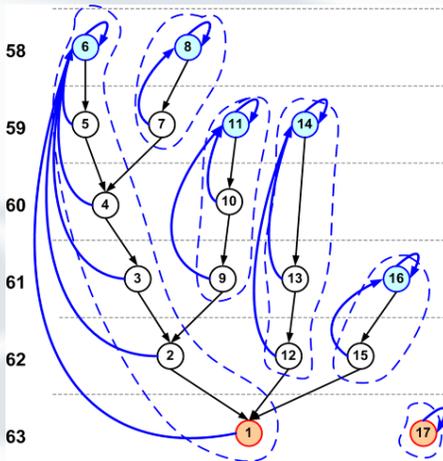
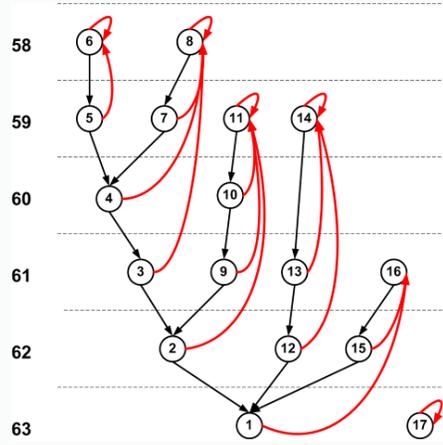


Image Credit: Springel et al (2005)

Simulation post-processing pipeline





gavo.mpa-garching.mpg.de/MyMillennium/

Virgo - Millennium Database

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

Maximum number of rows to return to the query form:

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 on this page.

Mainly Halos:

Mainly Galaxies:

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

CasJobs

skyserver.sdss3.org/CasJobs/SubmitJob.aspx

SDSS Query / CasJobs

Help Tools Query History MyDB Import Groups Output Profile Queues SkyServer Logout deppm

Context Table (optional) Task Name

DR10 MyTable My Query

Samples Recent Clear [1 s]

Query complete! Syntax Plan Quick Submit

```

1 select top 40 p.objid,p.ra,p.dec,p.u,p.g,p.r,p.i,p.z
2 from photoobj_p
3 join dbo.fgetnearbyobjeq(44.41, 5.99, 40) n on p.objid=n.objjid
4 where p.g between 14 and 18
    
```

40 row(s)

objid	ra	dec	u	g	r	i	z
1237667228764668018	44.4251887464174	5.99796533218519	17.45296	16.15438	15.62593	15.41414	15.32727
1237667228764668016	44.4136115331179	6.01166729833912	15.93901	14.92868	14.99346	14.86291	12.71118
1237667228764668020	44.3902630919155	5.97695081867117	16.96872	14.88728	13.86288	13.33199	12.91775
1237667228764668021	44.4400546420409	5.97533084978952	18.35162	16.20482	15.22672	14.71813	14.34378
1237667228764668022	44.4371469033969	5.98117747486914	19.06305	17.71112	17.12041	16.85705	16.74301
1237670016198443013	44.40830802330024	6.02169136812494	16.27055	15.73907	15.74194	15.6974	13.71063
1237670016198443017	44.4238465056448	6.024376643852	16.90246	14.66859	13.60513	13.04366	12.63324
1237667228764668003	44.3721225178138	6.00777066389459	16.3297	14.95203	14.42902	16.75699	14.15577
1237667228764668019	44.4514249394326	5.9876249662911	15.91361	15.65941	15.36141	14.96405	13.73247
1237667228764668024	44.4465743657887	5.9847330396699	18.88211	16.72498	15.74462	15.21885	14.93993
1237667228764668244	44.4402786447451	5.9525910090616	16.85921	15.68296	15.02576	14.59115	14.3564
1237670016198443015	44.423185823296	6.02691535484237	16.87503	14.63161	13.5211	12.95495	12.57853
1237667228764667924	44.346577913547	5.97438474657849	18.82665	16.70258	15.79071	15.4536	15.26757
1237667228764667924	44.346577913547	5.97438474657849	18.82665	16.70258	15.79071	15.4536	15.26757

RESULTS Plot Save As HTML

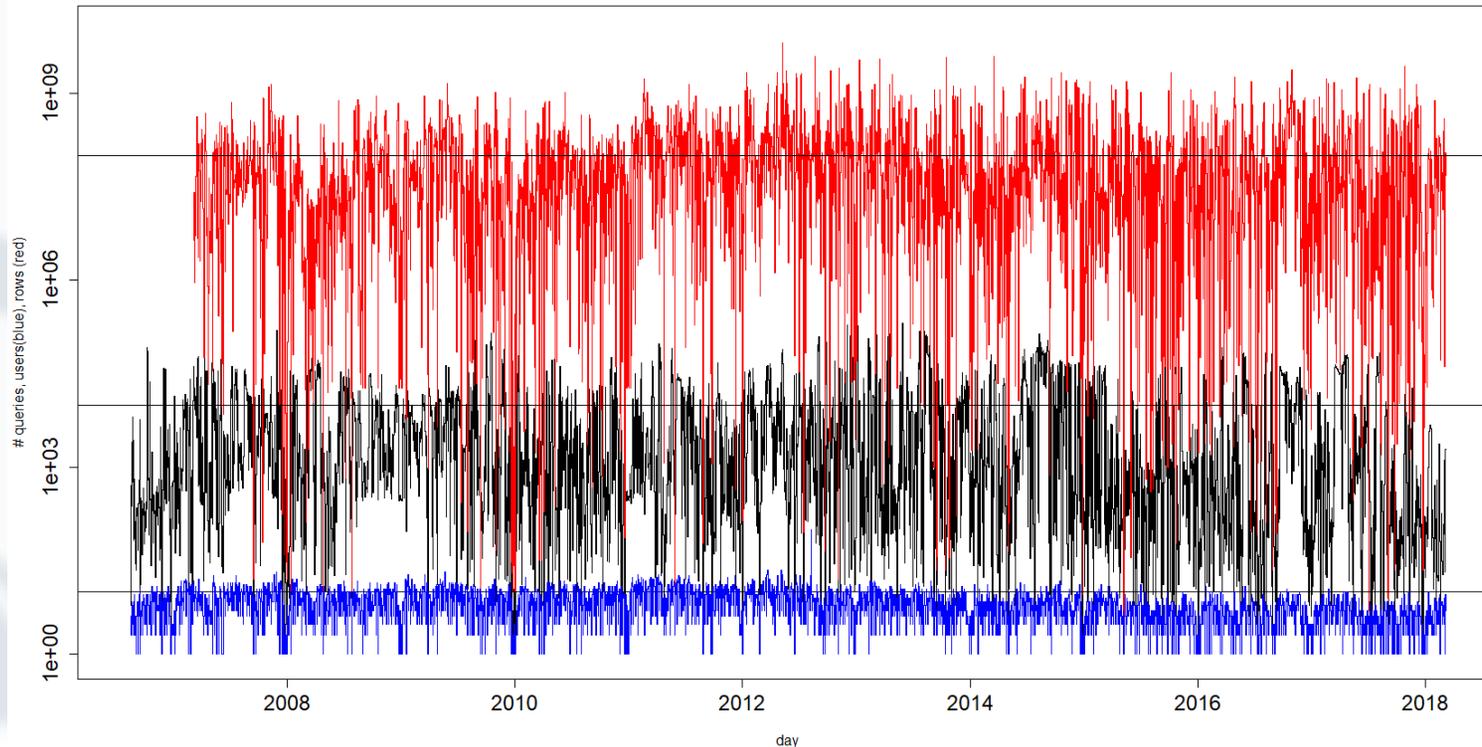
Display Query Results Both

Access statistics

> 1000 registered users

> 1000 publications (<http://www.mpa-garching.mpg.de/millennium/>)

total # queries = 35797095 = 8565/day
total # rows = 577505617538, = 145284432/day



SciServer: beyond databases and astrophysics

- ▶ **SciServer** is a system allowing Science Researchers across multiple domains to host and share their datasets and provide query and analysis tools for collaborative research.
- ▶ **Core Services:**
 - Science Data Hosting (Files and Databases)
 - Personal Storage (Files and Database)
 - Query hosted databases (Custom SQL)
 - Computational analysis on hosted data (Python R, Matlab, custom apps)
 - Collaboration and Sharing (create user groups, flexible access controls)
 - API Integration (easy to wrap SciServer with custom UIs)

SciServer Dashboard

Data, Collaboration, Compute



Files

You have 47 Shared User Volumes.
You have 33 Owned User Volumes.



Groups

You have 0 Group Invitations.
You have 47 Owned Groups.



Compute Jobs

You have 0 Jobs Running.
You have 0 Jobs Completed in 24 hours.



Activity Logs

You logged into the Dashboard on
02 Sep 2019 10:38:14 am.

SciServer Apps



CasJobs

Search online big relational databases collections, store the results online, and share them.



Compute

Analyze data with interactive Jupyter notebooks in Python, R and MATLAB.



Compute Jobs

Asynchronously run Jupyter notebooks in Python, R and MATLAB or commands.



SciDrive

Drag-and-drop file hosting and sharing services.



SkyServer

Access the Sloan Digital Sky Survey data, tutorials and educational materials.



SkyQuery

A scalable database system for cross-matching astronomical source catalogs.

Linking relational catalog to image archives

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

```

import sciServer.login.token as token
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.fGetUrfitsCFrame(d.fieldid, 'g') as fits_g,
       dbo.fGetUrfitsCFrame(d.fieldid, 'r') as fits_r,
       dbo.fGetUrfitsCFrame(d.fieldid, 'z') as fits_z,
       p.ra, d.ra as match_ra, p.dec, d.dec as match_dec
       , p.petroz90_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.objid=b.objid -- join with all the matches
join photoobjall d on c.objid=d.objid -- get match photoobj
order by d.fieldid
"""
queryResponse = SciServer.CasJobs.executeQuery(sql, "Stripe82", token=token)
obs = pandas.read_csv(queryResponse, index_col=None)
obs[:10]
    
```

	head	match	matchcount	head_field	match_field	fits_g
0	8658194378960928809	8658194430499553320	57	8658194378960928768	8658194430499553280	http://das.sdss.org/imaging/5622/40/corr
1	8658194378960928809	8658194477742948377	57	8658194378960928768	8658194477742948352	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	8658194378960928768	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 0x7f2aa174e6d8>

```
jupyter FragData - Analysis - Demo Last Checkpoint 4 minutes ago (autosaved)
File Edit View Insert Cell Kernel Help Python 3.0
CellToolbar
```

```
jupyter FragData - Plotting Cracks Last Checkpoint 5 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Help Python 3.0
CellToolbar
```

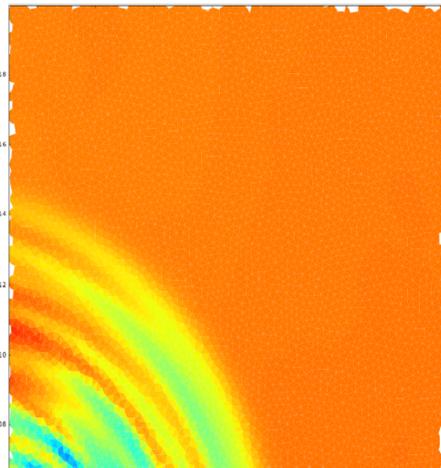
```
jupyter FragData - Analysis - Fragments Last Checkpoint 7 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Help Python 3.0
CellToolbar
```

FragData - Analysis: Plotting Elements

```
In [1]: import SciServer.LoginPortal as Login
import SciServer.LoginPortal as Login
import SciServer.CareOla as CareOla
import pandas
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.collections import LineCollection
from matplotlib.collections import LineCollection

In [2]: query="""
select e1.snapsnum, e1.elementsid
from elements e1
where e1.snapsnum=200
and e1.x1 between (.00000 and (.00002
and e1.y1 between (.00000 and (.00002
"""
# query CareOla table: using FragData as context
ctx = CQ.executeQuery(query, "FragData")
elements=pandas.read_csv(ctx, index_col=None)
l1=list(zip(elements.x1,elements.y1))
l2=list(zip(elements.x2,elements.y2))
l1=l1+l2
l2=l2+l1
l1=l1+l2
l2=l2+l1
l1=l1+l2
l2=l2+l1
l1=l1+l2
l2=l2+l1
edges=np.array(l1+l2)

In [4]: fig, ax = plt.subplots()
fig.set_size_inches(15,15, forward=True)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('Elements')
ax.grid(True)
ax.set_xlim(0,0.00002)
ax.set_ylim(0,0.00002)
plt.show()
```

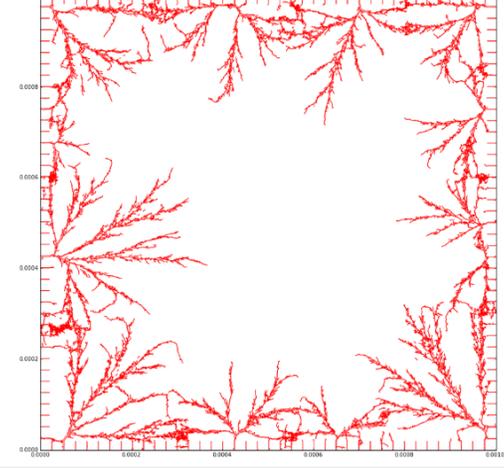


FragData - Analysis: Plotting Cracks

```
In [1]: Most complex analysis and visualization is performed in python. This notebook contains relevant code.
import SciServer.LoginPortal as Login
import SciServer.LoginPortal as Login
import SciServer.CareOla as CareOla
import pandas
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.collections import LineCollection
from matplotlib.collections import LineCollection

In [2]: # query for cracks
ctx = CQ.executeQuery(query, "FragData")
cracks = pandas.read_csv(ctx, index_col=0)
# define collection of segments
l1=list(zip(cracks.x1,cracks.y1))
l2=list(zip(cracks.x2,cracks.y2))
l1=l1+l2
l2=l2+l1
l1=l1+l2
l2=l2+l1
edges=np.array(l1+l2)

In [3]: fig, ax = plt.subplots()
fig.set_size_inches(15,15, forward=True)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('Cracks')
ax.grid(True)
ax.set_xlim(0,0.00002)
ax.set_ylim(0,0.00002)
plt.show()
```

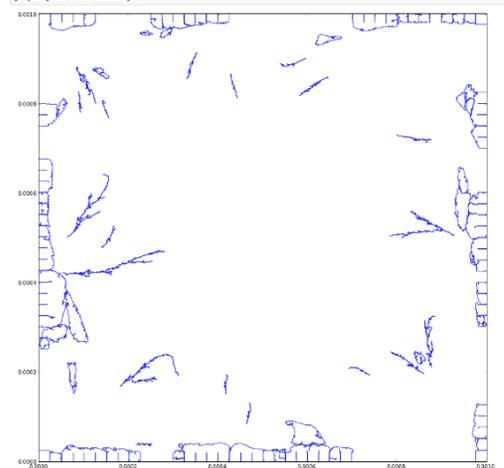


FragData - Analysis: Fragments

```
In [1]: import SciServer.LoginPortal as Login
import SciServer.LoginPortal as Login
import SciServer.CareOla as CareOla
import pandas
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from matplotlib.collections import LineCollection
from matplotlib.collections import LineCollection

In [2]: # define graph of cracks
g=graph(edges)
pos = list(zip(cracks.x1, cracks.y1))
edges=list(zip(cracks.model,cracks.model2))
g.add_edges(edges)
g.draw_networkx_pos1(g,pos,ax=ax)

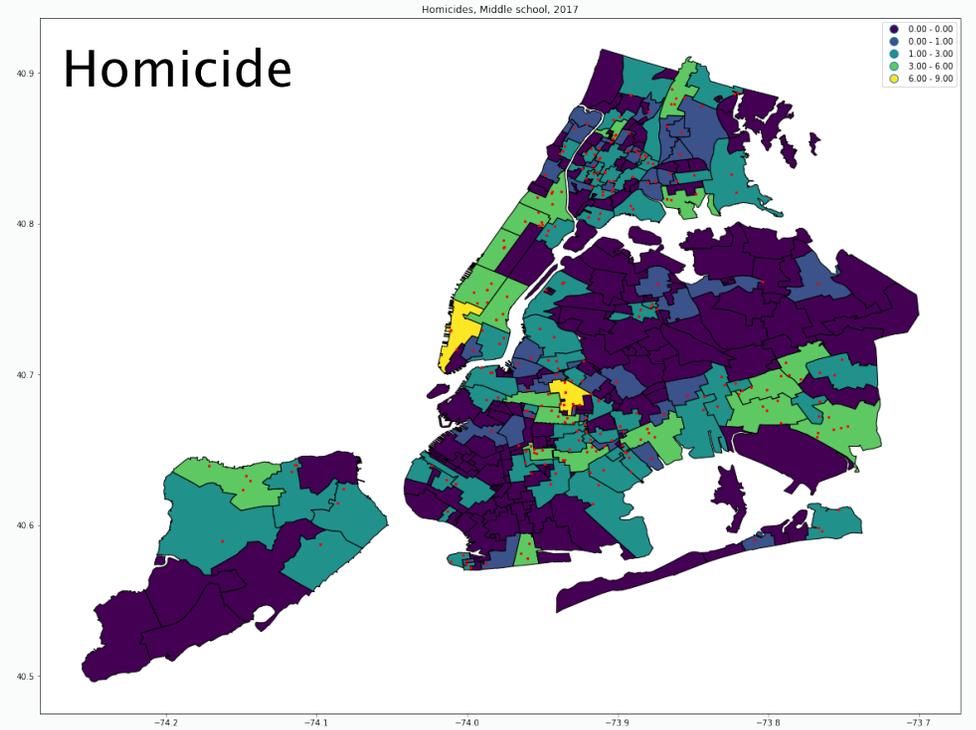
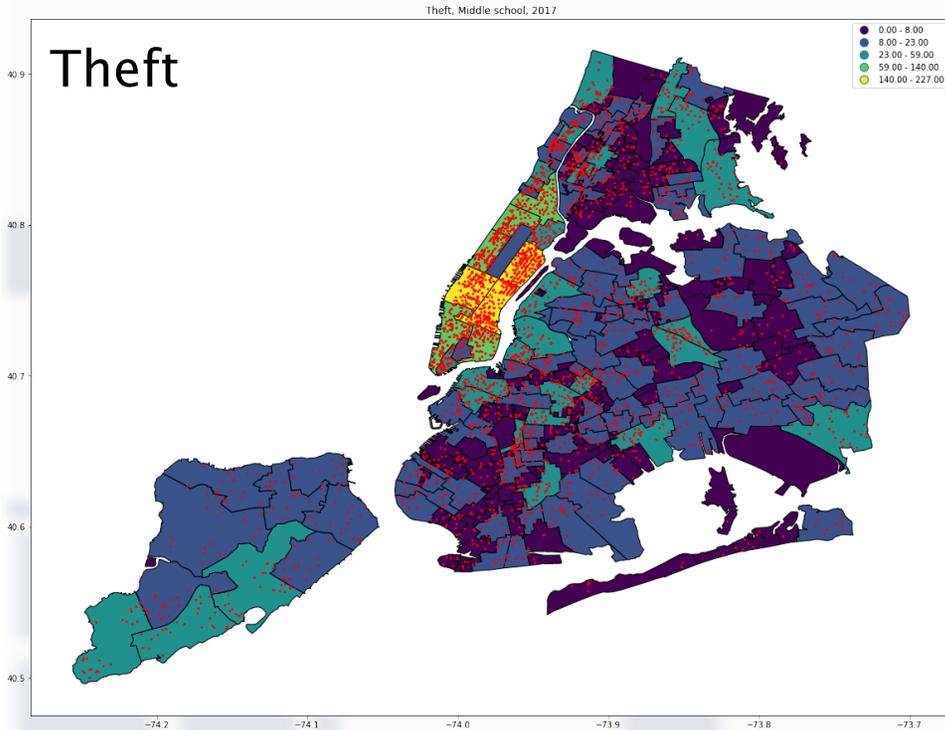
In [3]: fig, ax = plt.subplots()
fig.set_size_inches(15,15, forward=True)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('Fragments')
ax.grid(True)
ax.set_xlim(0,0.00002)
ax.set_ylim(0,0.00002)
plt.show()
```



Analyzing brittle materials under stress

(With Nitin Daphalpurkar)

Crimes in school districts in NY. To be correlated to school attendance with J. Burdick-Will



Analyzing metre in middle English poetry

(Christopher Cannon, Mark Patton)

```
In [6]: # iterate over features and print the original and best_parse version of each line
for line in txt.lines():
    best_parse = line.bestParse() # most plausible parse
    all_pauses = line.allPases() # all plausible parses

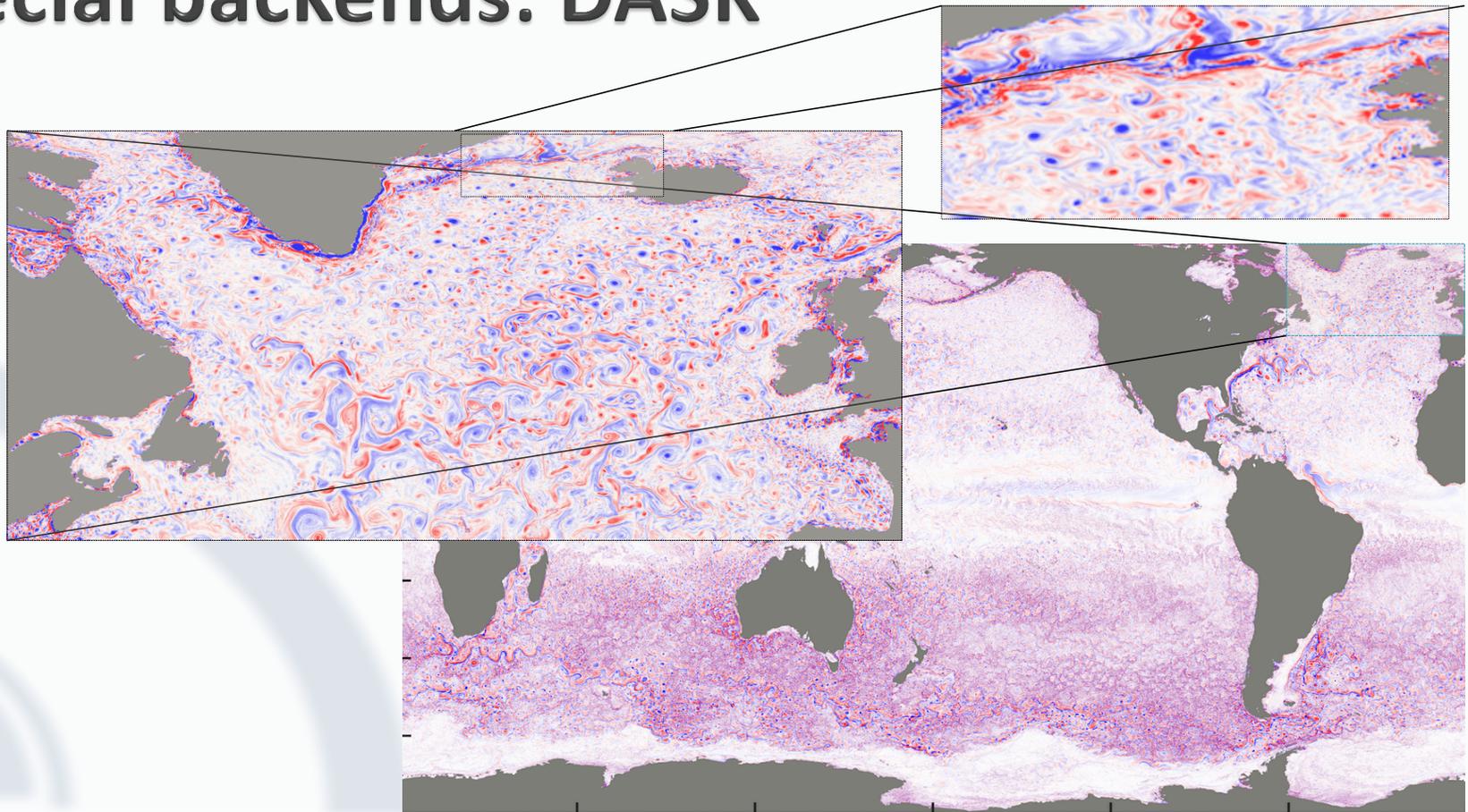
    first_word = line.words()[0]
    last_syllable = line.syllables()[-1]
    last_syllable_rime = line.rimes()[-1]
    last_syllable_rime_phonemes = last_syllable_rime.phonemes()

    # print original line and best parsed version nicely, assuming maximum linelength< MAXLEN=60
    MAXLINE=60
    sline=str(line)
    blanks=" "*(MAXLINE-len(sline))
    print sline,blanks,best_parse
```

Turning and turning in the widening gyre
 The falcon cannot hear the falconer;
 Things fall apart; the centre cannot hold;
 Mere anarchy is loosed upon the world,
 The blood- dimmed tide is loosed, and everywhere
 The ceremony of innocence is drowned;
 The best lack all conviction, while the worst
 Are full of passionate intensity.

TURN|ing.and*|TURN|ing|IN|the|WIDEN|ing|GY|re
 the|FAL|con|CAN|not|HEAR|the|FAL|coner
 things*|FALL|ap|ART|the|CEN|tre|CAN|not|HOLD
 mere*|AN|ar|CHY*|is|LOOSED|up|ON|the|WORLD
 the|BLOOD|dimmed*|TIDE|is|LOOSED|and|EV|ery|WHERE
 the|CER|e|MO|ny.of*|IN|no|CENCE*|is|DROWNED
 the|BEST|lack*|ALL|con|VIC|tion|WHILE|the|WORST
 are|FULL|of|PAS|sion|ATE*|in|TEN|si|TY*

Special backends: DASK



Custom user interfaces

Johns Hopkins Turbulence Database

SciUI - JHU Turbulence DB - Cutout Service

Log in

Johns Hopkins Turbulence Database gerard -

Instructions:

- Enter your turbulence token and select your desired cut out parameters.
- Note, if you keep the default token you cannot create cutouts larger than 4096 cells.
- Click on the submit button to start a new cutout job.
- Links to the results can be found in the job list below.

Job Name:

Token:

Enter your personal turbulence token here.

Dataset:

Function:

	Start	End	Stride
T [1-4000]	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
X [1-2048]	<input type="text" value="1"/>	<input type="text" value="10"/>	<input type="text" value="1"/>
Y [1-612]	<input type="text" value="1"/>	<input type="text" value="10"/>	<input type="text" value="1"/>
Z [1-1536]	<input type="text" value="1"/>	<input type="text" value="10"/>	<input type="text" value="1"/>

Jobs List

Show entries Search:

Submitted	Runtime Info	Input	Results
2019-09-04 15:28:16	Job name: __turbcutout__20190904 Status: SUCCESS Job Id: 56601 Duration: 9.546s	{"dataset": "channel", "function": "u", "output_filename": "channel", "stride": 1, "stride_x": 1, "stride_y": 1, "stride_z": 1, "ts": 1, "token": "edu.jhu.pha.turbulence.testing-201406", "ts": 1, "xe": 10, "xs": 1, "ye": 10, "ys": 1, "ze": 10, "zs": 1}	Job Folder channel.h5 channel.vmf notebook
2019-08-29 10:59:04	Job name: __turbcutout__FOOBAR Status: SUCCESS Job Id: 56529 Duration: 9.566s	{"dataset": "channel", "function": "u", "output_filename": "channel", "stride": 1, "stride_x": 1, "stride_y": 1, "stride_z": 1, "ts": 1, "token": "edu.jhu.pha.turbulence.testing-201406", "ts": 1, "xe": 10, "xs": 1, "ye": 10, "ys": 1, "ze": 10, "zs": 1}	Job Folder channel.h5 channel.vmf notebook
2019-07-29 12:50:17	Job name: __turbcutout__	{"dataset": "isotropic1024coarse", "filter_width": 1, "function": "u", "output_filename": "isotropic1024coarse", "stride": 1,	Job Folder isotropic1024coarse.h5

