SciServer Compute

bringing analysis close to the data

Jai Won Kim, Institute for Data Intensive Engineering and Science, Krieger School of Arts & Sciences, Johns Hopkins University

Astronomy

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

File Edit View Insert Cell Kernel Help

Code CellToolbar

```
import SciServer.LoginPortal 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.petror90_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                               -- order by matchhead objid
"""
queryResponse = SciServer.CasJobs.executeQuery(sql, "Stripe82",token=token)
obss = pandas.read_csv(queryResponse,index_col=None)
obss[:10]
```

Out[18]:

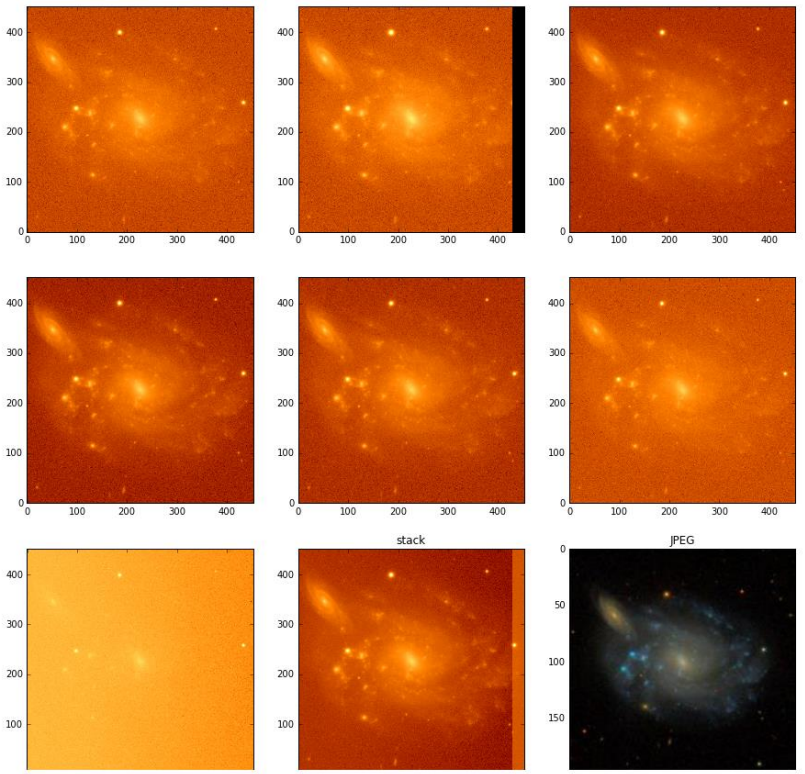
	head	match	matchcount	head_field	match_field	fits_g
0	8658194378960928809	8658194430499553320	57	8658194378960928768	8658194430499553280	http://das.:
1	8658194378960928809	8658194477742948377	57	8658194378960928768	8658194477742948352	http://das.:
2	8658194378960928809	8658194516375371821	57	8658194378960928768	8658194516375371776	http://das.:
3	8658194378960928809	8658194585083510793	57	8658194378960928768	8658194585083510784	http://das.:
4	8658194378960928809	8658194804163018771	57	8658194378960928768	8658194804163018752	http://das.:
5	8658194378960928809	8658194954470752297	57	8658194378960928768	8658194954470752256	http://das.:
6	8658194378960928809	8658195018907910161	57	8658194378960928768	8658195018907910144	http://das.:
7	8658194378960928809	8658195044651040803	57	8658194378960928768	8658195044651040768	http://das.:
8	8658194378960928809	8658195066151239724	57	8658194378960928768	8658195066151239680	http://das.:
9	8658194378960928809	8658195113395748890	57	8658194378960928768	8658195113395748864	http://das.:

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

File Edit View Insert Cell Kernel Help Python 3

Code CellToolbar

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



Materials Science

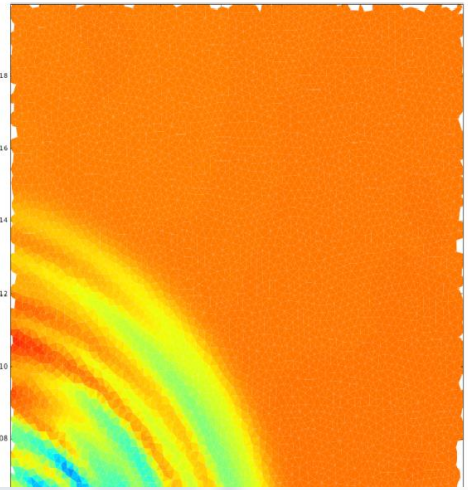
jupyter FragData - Analysis - Demo Last Checkpoint: 4 minutes ago (autosaved)

FragData - Analysis: Plotting Elements

```
In [1]: import SciServer.LoginPortal as Login
token=Login.getToken()
import SciServer.CasJobs as CJ
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

In [2]: query="""
select e.snapnum, e.elementid
,    e.x1,e.y1, e.x2 ,e.y2, e.s1 ,e.y3
,    e.s0,e.s1,e.s2,e.s3
,    0.5*(e.s0+e.s1) as t, e.s0*e.s0-e.s1*e.s2 as d
  from elemstress_sd1 e
  where e.snapnum=290
    and e.x1 between 0.00095 and 0.0002
    and e.y1 between 0.00095 and 0.0002
"""
# query CasJobs table. Using FragData as context
csv = CJ.executeQuery(query, "FragData")
elements=pandas.read_csv(csv,index_col=None)
l1=list(zip(elements.x1,elements.y1))
l2=list(zip(elements.x2,elements.y2))
l3=list(zip(elements.x3,elements.y3))
verts=list(zip(l1,l2,l3))
t=elements.t-np.sqrt(elements.t*elements.t-elements.d)
color=np.array(elements.s0)

In [4]: fig, ax = plt.subplots()
fig.set_size_inches(15,15, forward=True)
coll = PolyCollection(verts, array=color, cmap=cm.jet, edgecolors='none')
ax.add_collection(coll)
ax.autoscale_view()
ax.set_aspect('equal')
ax.set_xlim(min(elements.x1),max(elements.x1))
ax.set_ylim(min(elements.y1),max(elements.y1))
plt.show()
```



jupyter FragData - Plotting Cracks Last Checkpoint: 5 minutes ago (unsaved changes)

FragData - Analysis: Plotting Cracks

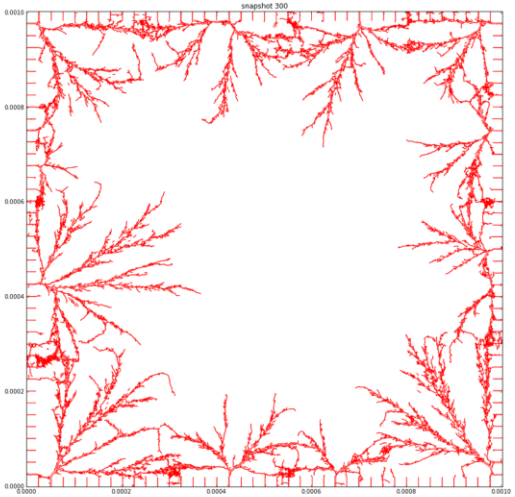
Most complex analysis and visualization is performed in python. This notebook contains relevant code.

```
In [1]: import SciServer.LoginPortal as Login
token=Login.getToken()
import SciServer.CasJobs as CJ
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

In [2]: # query for cracks
query="""
select e1.*
, n1.x as x1,n1.y as y1
, n2.x as x2,n2.y as y2
  from cracks_sd1 e1
  ,    nodes_sd1 n1
  ,    nodes_sd1 n2
where e1.snapnum=300
   and n1.snapnum=e1.snapnum
   and n1.nodeid=e1.node1
   and n2.snapnum=e1.snapnum
   and n2.nodeid=e1.node2
   """
queryResponse = CJ.executeQuery(query, "FragData")
cracks = pandas.read_csv(queryResponse,index_col=0)
# define collection of segments
cl1=list(zip(cracks.x1,cracks.y1))
cl2=list(zip(cracks.x2,cracks.y2))
clines = list(zip(cl1,cl2))

In [4]: fig, ax = plt.subplots()
fig.set_size_inches(15,15, forward=True)
lc = LineCollection(clines, linewidths=1,color='red')
ax.add_collection(lc)
ax.set_aspect('equal')
ax.set_xlim(0,max(cracks.x1))
ax.set_ylim(0,max(cracks.y1))
ax.set_title("snapshot 300")

plt.show()
```



jupyter FragData - Analysis - Fragments Last Checkpoint: 7 minutes ago (unsaved changes)

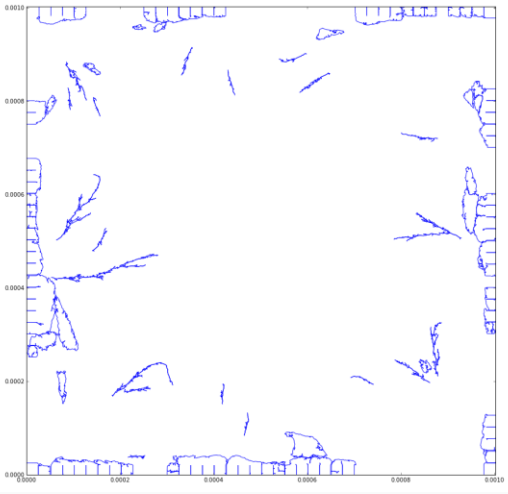
FragData - Analysis: Fragments

```
In [1]: import SciServer.LoginPortal as Login
token=Login.getToken()
import SciServer.CasJobs as CJ
import pandas
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.collections import LineCollection
import cycles as cy

In [2]: # query for edges bounding cracks: those with only 1 element
query="""
select e1.*
, n1.x as x1,n1.y as y1
, n2.x as x2,n2.y as y2
  from cracks_sd1 e1
  ,    nodes_sd1 n1
  ,    nodes_sd1 n2
where e1.snapnum=290
   and n1.snapnum=e1.snapnum
   and n1.nodeid=e1.node1
   and n2.snapnum=e1.snapnum
   and n2.nodeid=e1.node2
   """
queryResponse = CJ.executeQuery(query, "FragData")
cracks = pandas.read_csv(queryResponse,index_col=0)
l1=list(zip(cracks.x1,cracks.y1))
l2=list(zip(cracks.x2,cracks.y2))
lines = list(zip(l1,l2))

In [3]: # define graph of cracks
g=cy.Graph()
pos = list(zip(cracks.x1, cracks.y1))
g.addNodes(cracks.node1, pos)
edges=list(zip(cracks.node1,cracks.node2))
g.addEdges(edges)
cycles=cy.DetectCycles(g,minSize=50)

In [9]: fig, ax = plt.subplots()
fig.set_size_inches(15,15, forward=True)
for c in cycles:
    if len(c.edges) > 300:
        continue
    pos = [(e.f.data,e.t.data) for e in c.edges]
    lc = LineCollection(pos, linewidths=1,color='blue')
    ax.add_collection(lc)
x1=plt.xlim(0,max(cracks.x1))
y1=plt.ylim(0,max(cracks.y1))
```



Cosmology

jupyter CosmoUC_5_PlotHalo Last Checkpoint: 11 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help Notebook saved Python 3

Code CellToolbar

Dark-matter Halos in a Cosmological Simulation

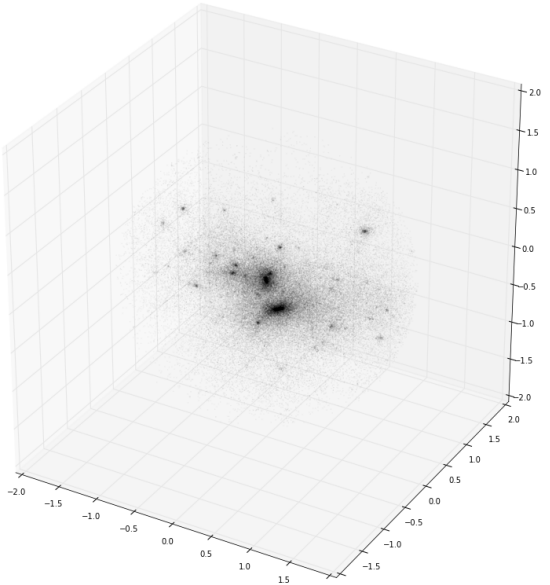
In [3]:
```
import SciServer.LoginPortal as Login
token = Login.getToken()
import SciServer.CasJobs
import pandas
import tables
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
```

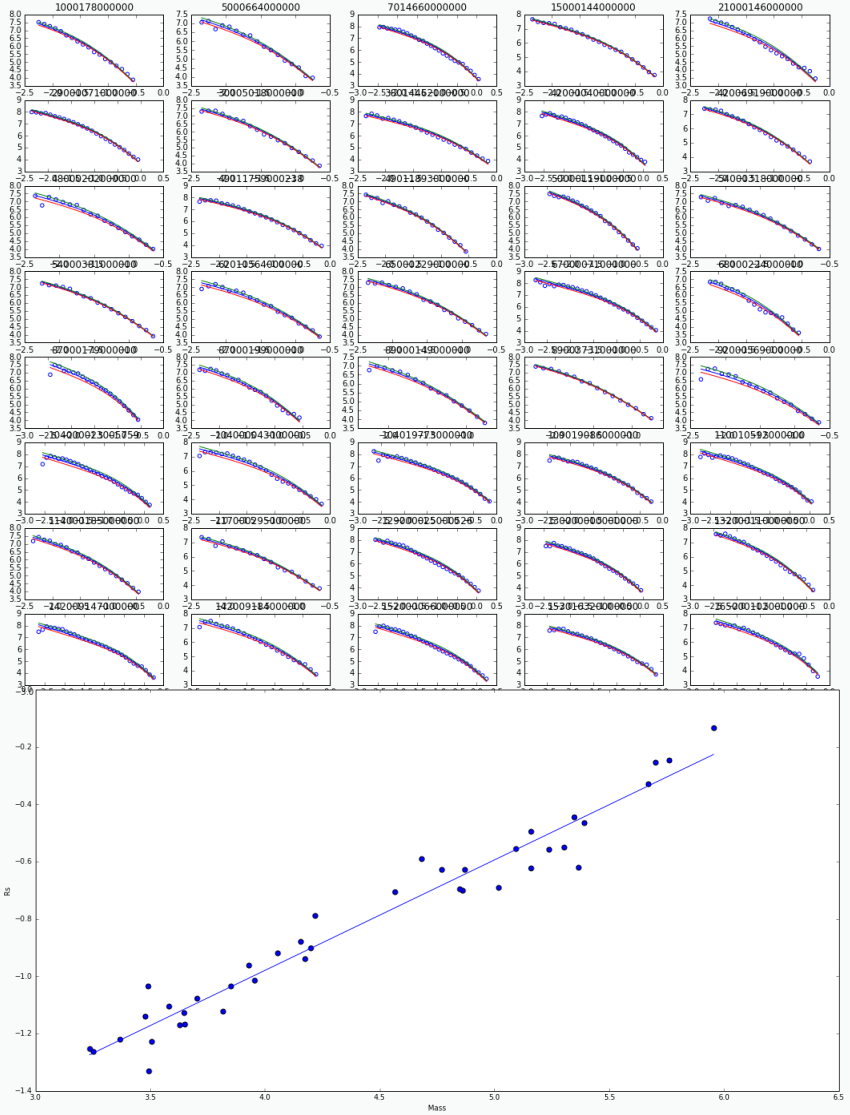
In [10]:
```
%%time
queryString = """
select top 100000 p.x-hh.x as x,p.y-hh.y as y,p.z-hh.z as z
from mpahalotrees.mr hh
   cross apply dbo.MillenniumParticles(hh.snapnum,
     dbo.Sphere::New(hh.x,hh.y,hh.z,3*hh.halfmassradius).ToString()) p
  where hh.haloid=84000007000000 order by newid()
    """
responseStream = SciServer.CasJobs.executeQuery(queryString, token=token,context="SimulationDB")
df = pandas.read_csv(responseStream, index_col=None)
```

```
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=0.001)
```

Out[13]: <mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x7fe86428b9e8>





your data here?