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...
Collaborative data-driven science

Form Based Queries

[Image of SkyServer DR12 interface]

**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](#).

<table>
<thead>
<tr>
<th>Type of search</th>
<th>Equatorial (RA / Dec)</th>
<th>Galactic (l / b)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RA</td>
<td>258.25</td>
<td></td>
</tr>
<tr>
<td>Dec</td>
<td>64.05</td>
<td></td>
</tr>
<tr>
<td>radius [arcmins]</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
</tr>
</tbody>
</table>

Output Format: [HTML] [XML] [CSV] [JSON] [VOTable] [FITS]

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

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.)
Custom SQL

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.

```
-- 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

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).
Collaborative data-driven science

Batch Queries, MyDB

<table>
<thead>
<tr>
<th>View</th>
<th>Name</th>
<th>Size</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycleedges</td>
<td>cycleedges</td>
<td>1,032</td>
<td>42,086</td>
</tr>
<tr>
<td>CycleEdges_SD1</td>
<td>cycleedges</td>
<td>84.552</td>
<td>3,559,046</td>
</tr>
<tr>
<td>Cycles_sd1</td>
<td>cycleedges</td>
<td>9.416</td>
<td>157,289</td>
</tr>
<tr>
<td>Cycle_edges</td>
<td>cycleedges</td>
<td>16.82</td>
<td>1</td>
</tr>
<tr>
<td>Cycle_edges2</td>
<td>cycleedges</td>
<td>16.82</td>
<td>1</td>
</tr>
<tr>
<td>Cycle_edges3</td>
<td>cycleedges</td>
<td>16.82</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table Schema</th>
<th>objid</th>
<th>ra</th>
<th>dec</th>
<th>petror90_r</th>
<th>fits_r</th>
<th>pubURL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>123765</td>
<td>0.01848758</td>
<td>-0.08335432</td>
<td>13.45019</td>
<td><a href="http://dr12.sdss3.org/sas/dr12/">http://dr12.sdss3.org/sas/dr12/</a></td>
<td></td>
</tr>
</tbody>
</table>
NOTICE: All systems are currently functional. For past announcements, please click here.

Welcome to the Johns Hopkins Turbulence Database (JHTDB)!

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 periodically be unavailable as we continue to add functionalities.
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 libfreetype5-dev
sudo apt-get install libhdf5-dev
sudo easy_install numpy
sudo easy_install h5py
sudo easy_install matplotlib
sudo python setup.py install
```

More information and source code can be found on github at https://github.com/ldies/pyJHTDB
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
Latest: SciServer-Compute

- Jupyter Notebooks in Docker
  - [https://developer.rackspace.com/blog/how-did-we-serve-more-than-20000-ipython-notebooks-for-nature/](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
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. async mode is available 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 Surveys' 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 size larger than 10 arcsec.
# We return
query=====
SELECT TOP 16 objId, p.ra, p.dec, p.petr90_r
FROM galaxy AS p
JOIN SpecObj AS s ON s.bestobjid = p.objId
WHERE p.u BETWEEN 0 AND 19.6
AND p.q BETWEEN 0 AND 17
AND p.petr90_r > 10
==

# query CasJobs tables. Using DR12 as context. i.e. a connection is made to DR12 when running this query
response = 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
```

<table>
<thead>
<tr>
<th>objId</th>
<th>ra</th>
<th>dec</th>
<th>petr90_r</th>
</tr>
</thead>
<tbody>
<tr>
<td>1237645941824359443</td>
<td>40.285696</td>
<td>-0.714957</td>
<td>14.72862</td>
</tr>
<tr>
<td>123764594378363604</td>
<td>55.123852</td>
<td>0.872487</td>
<td>18.86110</td>
</tr>
<tr>
<td>123764594379311121</td>
<td>57.248385</td>
<td>0.925979</td>
<td>14.02575</td>
</tr>
<tr>
<td>1237645943791141622</td>
<td>56.847420</td>
<td>0.875488</td>
<td>15.66479</td>
</tr>
<tr>
<td>1237646588246688257</td>
<td>60.135126</td>
<td>1.186679</td>
<td>11.14294</td>
</tr>
<tr>
<td>123764687292337965</td>
<td>245.709633</td>
<td>0.844301</td>
<td>10.51686</td>
</tr>
<tr>
<td>1237646872459667234</td>
<td>245.782081</td>
<td>0.492432</td>
<td>11.98310</td>
</tr>
</tbody>
</table>
Collaborative data-driven science

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 making it available as a system variable for the session
$ token must be replaced with raw one once it has acquired
with open("/home/Idle/fragments/token","r") as f:
    token = f.read().strip(\"\")

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
snapshot=200
nxmin=0.05008
nxmax=0.0618
nymin=0.0005
nymax=0.00115

query=""
declare $xmin float = "\"$nxmin\"", $xmax float = "\"$nxmax\""
declare $ymin float = "\"$nymin\"", $ymax float = "\"$nymax\"
declare $xmin integer, $xmax integer, $ymin integer, $ymax integer
select $xminFloor(($xmin-min)/$dx), $yminFloor(($ymin-min)/$dy), $xmaxFloor(($xmax-min)/$dx), $ymaxFloor(($ymax-min)/$dy) from $Simulations

where suffix=="""suffix"

select a, snapshot, a.elementId
    , x1, y1
    , x2, y2
    , x3, y3
    , x5, y5
    , x7, y7
    , x15, y15
    , x30, y30, x50, y50 as d
    , 0.5*(x1+y1) as t, c.s2.x30-c.s1.x3 as d
from cells=="""suffix""" cell

where a.snapshot=="""suffix""a

and c.xmax between $xmin and $xmax
and c.xmin between $xmin and $xmax
and c.nymax==snapshot
and c.ymax==snapshot

$ query CasJobs table. Using FragData as context
<datetime.datetime.now()>!
queryResponse = SciServer.CasJobs.executeQuery(query, "FragData")
<datetime.datetime.now()>!
# parse results into pandas.DataFrame for further in memory processing
elements = pandas.read_csv(queryResponse, index_col=None)
elements['datetime'].sortby()
print("Found {} elements with total: {}".format(len(elements), elements.count()[0]))
print("Found {} rows, query time: {} / parse time: {}".format(len(elements), queryResponse))

In [13]: $ prepare for building notebooks
Collaborative data-driven science

Materials Science
Collaborative data-driven science

Turbulence
Collaborative data-driven science

```python
order_by pos

_hist=pandas.read_csv(queryResponse)

executeQuery POST response: 200 OK

In [26]:
plt.figure(figsize=(20,3))
plt.title('Chromosome 14')

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

Genomics
## Roadmap

<table>
<thead>
<tr>
<th>Step</th>
<th>Time (Months)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup</td>
<td></td>
</tr>
<tr>
<td>Re-Engineering and Component Integration</td>
<td></td>
</tr>
<tr>
<td>SDSS</td>
<td></td>
</tr>
<tr>
<td>Pilot Science Integration</td>
<td></td>
</tr>
<tr>
<td>Architectural Scaleout</td>
<td></td>
</tr>
<tr>
<td>Analysis Capability</td>
<td></td>
</tr>
<tr>
<td>Collaborative Environments</td>
<td></td>
</tr>
<tr>
<td>Science Integration</td>
<td></td>
</tr>
<tr>
<td>Outreach, Collaboration and Education</td>
<td></td>
</tr>
</tbody>
</table>

*Time (Months): 6 12 18 24 30 36 42 48 54 60*
Thank you.
I’ll be very happy to demo and discuss our services.