The NOAO Data Lab
Design, Capabilities and Community Development

Michael Fitzpatrick
for the Data Lab Team
Data Lab is Science Exploration Platform that provides:

- Repository for large datasets (catalogs & their parent images)
- Data access services (VO, SQL/ADQL, TAP, SIA, …)
- Virtual Storage (VOSpace, MyDB, notebooks)
- Gateway to remote data centers
- Data Sharing / Collaboration / Publication
- Website / authenticated portal / documentation
- Exploratory tools (survey coverage, catalogs overlays, …)
- Visualization tools (all-sky browser, custom APIs)
- Analysis facilities (Jupyter, utility methods, Compute Service)
Project Motivation
SMASH, DECaPS, Legacy Survey, DES

SMASH
Nidever et al 2017

log N per sq. deg.

0 7

National Optical Astronomy Observatory
Cerro Tololo Inter-American Observatory
Kitt Peak National Observatory
Community Science and Data Center

NOAO DATALAB

Cerro Tololo Inter-American Observatory
Kitt Peak National Observatory
Community Science and Data Center

SMASH, DECaPS, Legacy Survey, DES

DECaLS
Pl: Schlegal and Dey

SMASH
Nidever et al 2017

DECaPS
Schlafly et al 2017

log N per sq. deg.

0 7
SMASH, DECaPS, Legacy Survey, DES

DECaLS
PI: Schlegal and Dey

SMASH
Nidever et al 2017

DECaPS
Schlafly et al 2017

Dark Energy Survey

log N per sq. deg.
2.9 billion sources
30 billion measurements

NOAO Source Catalog

Nidever et al (2018)
Data Holdings

Catalog Database (100B rows, ~50TB)

Image Archive Holdings (~1.2PB)
Data Holdings

Catalog Data Holdings

- 316 tables in 38 different schema
- 15 different image services (by survey, reduction type)
- **Major Surveys:** DES, DESI, DECaLS, DECaPS, PHAT, S-PLUS, NSC, ....
- **Ref data:** AllWISE, Gaia, USNO, 2MASS, SDSS, ....
- **Planned(?) Datasets:** CRTS, Vista HS, UKIDSS, Chandra/XMM XSC, ....

- Q3C spatial indexing + data clustering + type ordering for performance
- TAP access to all tables (including image metadata)
- Cone Search services for all tables w/ positional data
- Image cutout services (SIA)
Design Principles to Consider

Know your audience

- Astronomers and Programmers will approach your system in different ways
- They will have a range of experience and skills
- A good UI will be inclusive
Design Principles to Consider

Allow multiple entry points into the system

- Web-based Portals
- Notebook Servers
- Command-line Tools
- Programmatic APIs

Novice User / Casual Browsing
Blackbelt User / Workflow Dev
Design Principles to Consider

Enable *user-developed* tools, don’t just supply them

- Support legacy code where possible
- Publish/Share code and notebooks
- Host a repo to support user contributions
- Allow for user scripting and workflow development
Design Principles to Consider

Use established standards, but hide complexity

• Provide *Astronomer Friendly* interfaces

• Provide *Low-Level* interfaces

• Realize these are *separate* things (and that’s okay)!

• Most of all: *Enable Science!*
Architecture/Service Overview

Clients
- Data Lab
  - Python API
  - Command Line
- External
  - Web Browser
  - VOSpace Clients
  - FUSE
  - VO Clients
    - TAPcat, ALADDIN, GAVO

Servers
- Auth
- Storage
- Job
- Query
- Resource Mgr
  - User
  - Group
  - Resource
  - Logging
- Job
- DALServer
- TAP/SIA/SCS/SSA
- Cutout
- XMatch
- VOSpace
- REDIS
- MyDB (Postgres)
- TAPDB (Postgres)
- VOSpace (MySQL)
- dlweb
- gp02
- Jupyter notebooks
- Redirect / Proxy
- Web server
- gp03
- gp04 / gp01
- gp03
- gp04 / gp01
- didb1
- VOSpace
VO Inside

VO Protocols and Data Access

- **VOSpace** for Virtual Storage
- **TAP** for Catalog Access
  - **ADQL** query language (custom funcs in development)
- **SCS** for Catalog Access
- **SIA** for Image Access
  - SIA v1 in use, SIA v2 available
  - **ObsCore** data model for images
- **SSA** available for Spectral Access (not currently used)

- **VOTable** support in client interfaces
- **HiPS** for Discovery Tool base image layers (MOC use planned)
- **Universal Worker Service** (planned for Compute Service)
- **VOSI** for VOSpace/TAP/Manager services
- **OAI Publishing Registry** for services planned
Public Data Access

• Access to public image / catalog data does not require an account

• Anonymous Jupyter notebook server allows a ‘Trial Run’ of Data Lab before creating an account
  • Scratch/demo notebooks, no persistence beyond session
  • Containerized environment initialized for each new user

• Downloadable `datalab` client and API allow anonymous desktop and programmatic access to Data Lab tools and services

• Web tools (`Data Discovery, Query Interface, etc`) can use anonymous access
User Account Creation

- Sign-up form for account request on web site
- New accounts require approval before resources are allocated
- Currently open to professional researchers only

Future plans for
- EPO/student accounts (e.g. custom notebooks)
- Group accounts
- Limited public access using temporary storage/compute services
## Resource Allocations

<table>
<thead>
<tr>
<th>Item</th>
<th>Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual disk storage</td>
<td>1 TB</td>
</tr>
<tr>
<td>MyDB database storage</td>
<td>250 GB</td>
</tr>
<tr>
<td>Jupyter notebook storage</td>
<td>no hard limit, but not unlimited</td>
</tr>
<tr>
<td>Sync query timeout</td>
<td>120 sec (max: 600 sec)</td>
</tr>
<tr>
<td>Async query timeout</td>
<td>24-hrs (max: unlimited)</td>
</tr>
</tbody>
</table>

- Storage quotas are not (presently) strictly enforced
- Query timeouts can be extended using an API argument
**Python API**

```python
from dl import storeClient as sc

sc.ls(format='long')
sc.ls('vos://sdss/spec')

data = sc.get('mydata.csv')
stat = sc.put('./mydata.csv', to='sdss')

stat = sc.cp('*.fits', '/raw')
```

**datalab Command Client**

```
% datalab ls -l
% datalab ls vos://sdss/spec

% datalab get 'vos://*.fits' out='./raw/
% datalab put ./*mydata.csv to='vos://sdss'

% datalab mkdir '/raw'
% datalab mv fr='*.fits' to='/raw'
```
**Python API**

```python
from dl import queryClient as qc

query = 'select * from usno.a2'
df = qc.query(query, fmt='pandas')

jobID = qc.query(adql=query)
while qc.status(jobID) != 'COMPLETED':
    time.sleep(1)
data = qc.results(jobID)
```

**datalab Command Client**

```bash
% sql = 'select * from usno.a2'
% datalab query $sql fmt=csv out=mydata.csv

% datalab query $sql async=true out='vos://mydata'
% datalab query $sql out='mydb://mydata'
% datalab import mydata.csv mydata
```
User Manuals

- Overviews
- HowTo guides
- API documentation
- Detailed science cases
- Glossary
- SQL examples
- Tips & tricks
Default Notebooks

- **Getting Started** guides
- **HowTo** guides
- Data Access Overviews
- Science Examples

![Jupyter Notebook Server Interface](image-url)
5.2 Construct and Submit a SQL Query

We apply a box search, selecting the RA and Dec limits that align with the region indicated above.

```python
In [67]: %time

# Write SQL query statement as a string.
query2 = ""
    SELECT obj_id, ra, dec, mean_mag_g_r, mean_mag_r_i,
        mean_mag_g, mean_mag_r, mean_mag_i, fracflux_g, fracflux_r, fracflux_i
    FROM decaps_drl.object
    WHERE (ra>134.7 AND ra<135.7 AND dec<-43.7 AND dec>-44.9
        AND NOT mean_mag_g_r='NaN' AND NOT mean_mag_g_r='Infinity' AND NOT mean_mag_g_r='-Infinity'
        AND NOT mean_mag_r_i='NaN' AND NOT mean_mag_r_i='Infinity' AND NOT mean_mag_r_i='-Infinity')
    LIMIT 100000"

# Submit the SQL query and return the output as a Pandas dataframe.
df2 = qbo.query(sql=query2, fmt='pandas')

CPU times: user 58 ms, sys: 9 ms, total: 67 ms
Wall time: 12.4 s
```
Default Notebooks

- **Getting Started** guides
- **HowTo** guides
- Data Access Overviews
- Science Examples

Notebook Server

```python
ax2.set_ylim([max(y2) * 0.5, min(y2) - 0.5])
cbl = plt.colorbar(im2, ax=ax2, label='log(N)')
```
Default Notebooks

- **Getting Started** guides
- **HowTo** guides
- Data Access Overviews
- Science Examples
Community Development

Science/Example/Tutorial Notebooks:
• Presently a “Contact Us” model to contribute, would like to move to a “Self Publish” with a directory

Source Code:
• Repos to be made public once we complete packaging/docs and remove configuration details
• Client pull requests being accepted
• (Future) User-contributed compute containers (e.g. Docker hub)

User Feedback!
Come see us at AAS 233 in Seattle
Lessons Learned

IF YOU BUILD IT

THEY WILL COME
Lessons Learned

IF YOU BUILD IT, AND ADVERTISE IT A LOT
THEY WILL TRICKLE IN
Lessons Learned

Data Lab user registration history

- daily rate
- all users
- regular users (no demo, test, internal)

registrations / day
registered users (cumulative)

date

Thank You

Get your account:
http://datalab.noao.edu

Contact us:
datalab@noao.edu
@NOAODataLab