



National Aeronautics and  
Space Administration

Jet Propulsion Laboratory  
California Institute of Technology  
Pasadena, California



**National Aeronautics and  
Space Administration**

**Jet Propulsion Laboratory**  
California Institute of Technology  
Pasadena, California

# Analyzing Hydrology Data with Apache Science Data Analytics Platform (SDAP)

Presented By: Frank Greguska (JPL)

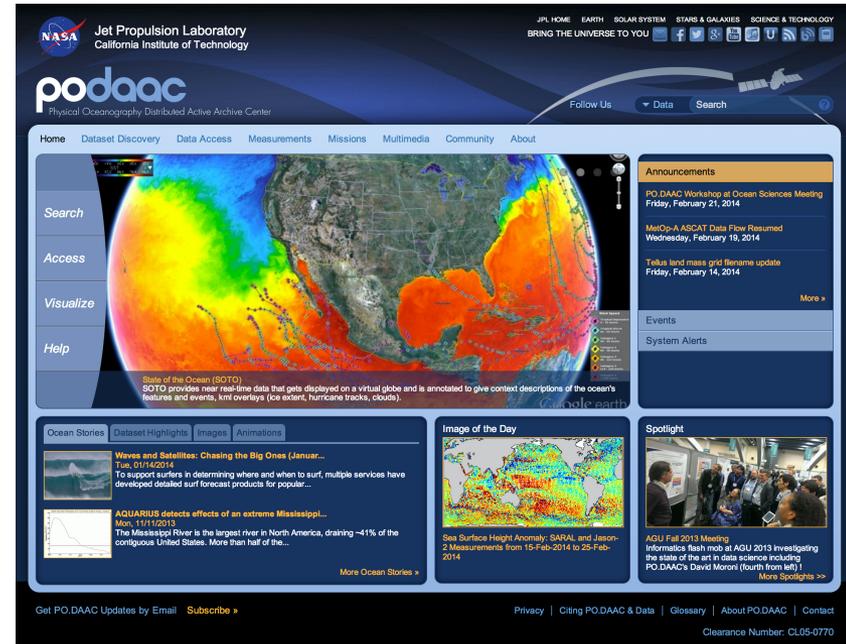
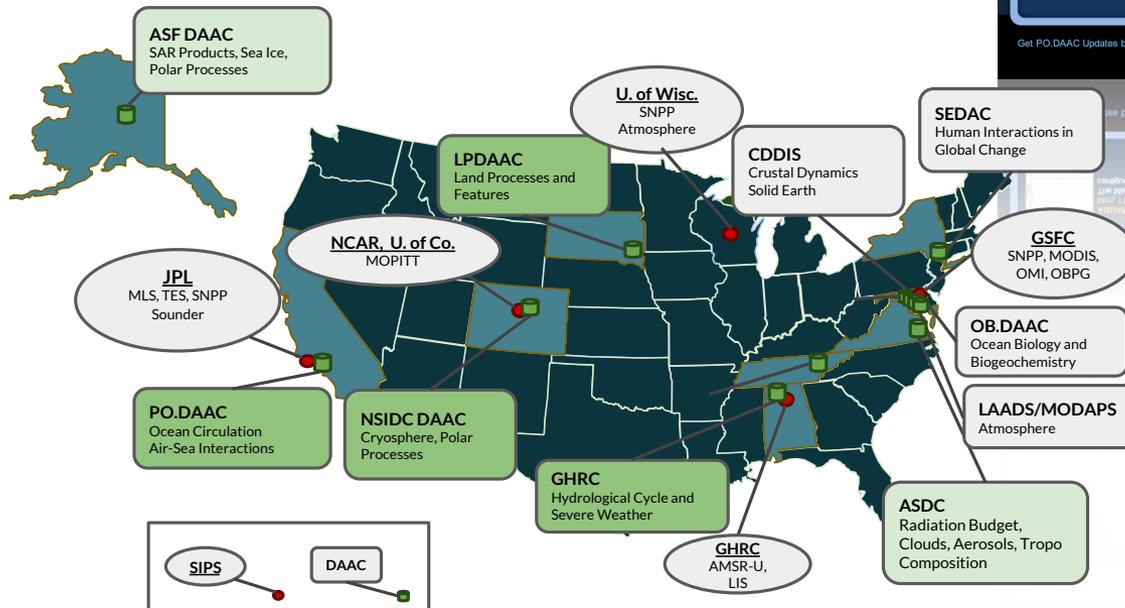
Jet Propulsion Laboratory  
California Institute of Technology  
4800 Oak Grove Drive, Pasadena, CA 91109-8099, U.S.A.

CL#xx-xxxx

© 2018 California Institute of Technology. Government sponsorship acknowledged. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsements by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

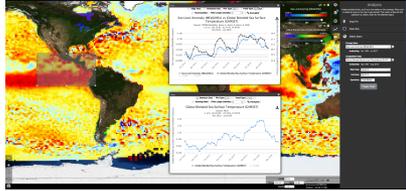
# PO.DAAC

- The **NASA Physical Oceanography Distributed Active Archive Center (PO.DAAC)** at Jet Propulsion Laboratory is an element of the **Earth Observing System Data and Information System (EOSDIS)**.
  - The EOSDIS provides science data to a wide communities of user for NASA's Science Mission Directorate.
- Archives and distributes data relevant to the physical state of the ocean
- The mission of the PO.DAAC is to preserve NASA's ocean and climate data and make these universally accessible and meaningful.

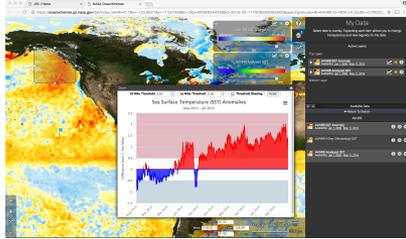


# Moving Beyond Archives

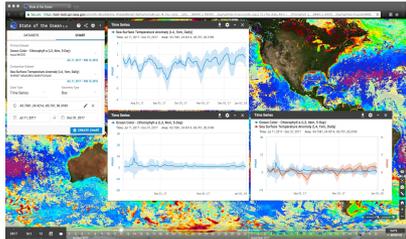
## NASA Sea Level Change Portal



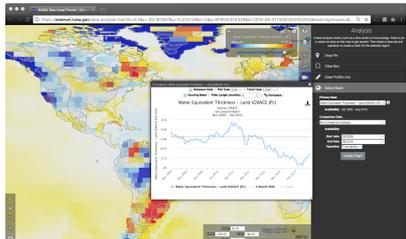
## Oceanographic Anomaly Detection



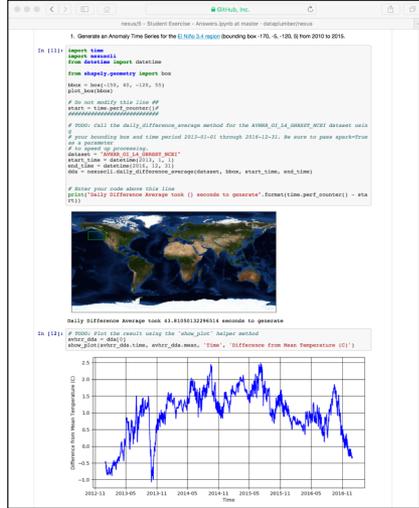
## PO.DAAC State Of The Ocean



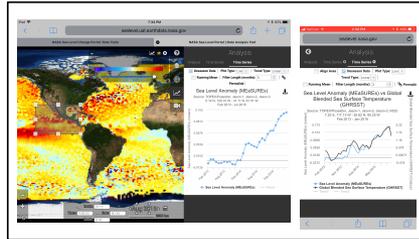
## Hydrological Basin Analysis



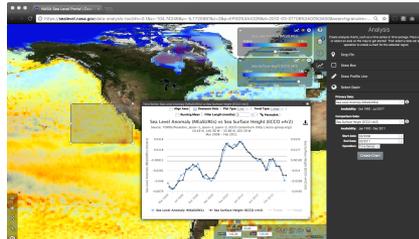
## Jupyter Notebook - Interactive Workbench



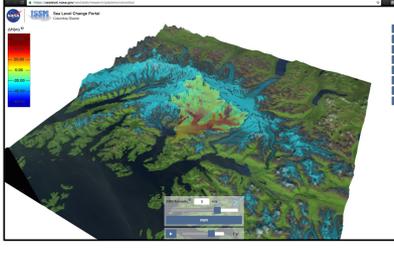
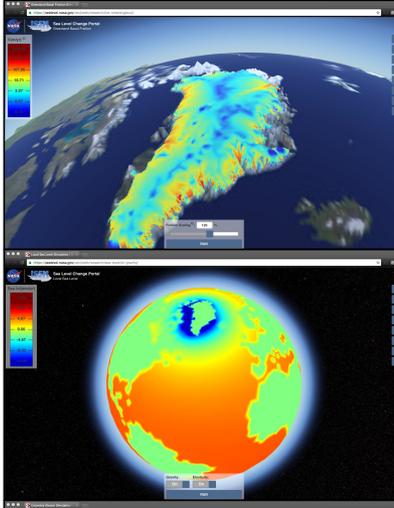
## Mobile Analysis



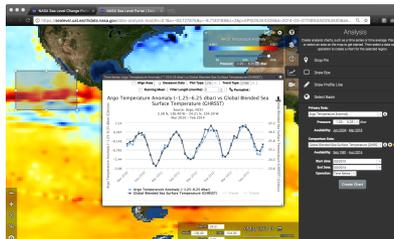
## In Situ Data Analysis



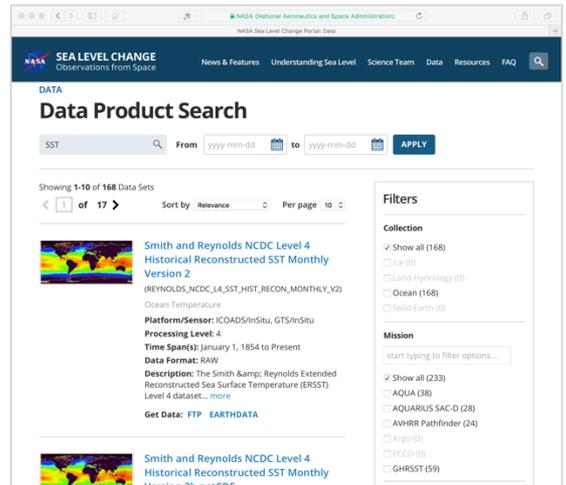
## Model Simulations



## Model - Observation Comparison



## Integrated Search and Discovery





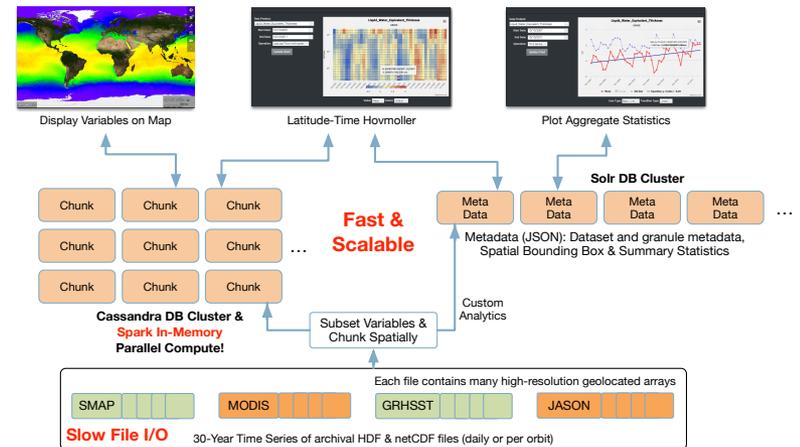
# Apache Science Data Analytics Platform (SDAP)

- **OceanWorks** is to establish an **Integrated Data Analytics Center** at the NASA Physical Oceanography Distributed Active Archive Center (PO.DAAC) for Big Ocean Science
- Focuses on technology integration, advancement and maturity
- Collaboration between JPL, Center for Atmospheric Prediction Studies (COAPS) at Florida State University (FSU), National Center for Atmospheric Research (NCAR), and George Mason University (GMU)
- Bringing together PO.DAAC-related big data technologies
  - Big data analytic platform
  - Anomaly detection and ocean science
  - Distributed in situ to satellite matchup
  - Dynamic datasets ranking and recommendations
  - Sub-second data search solution and metadata translation and services aggregation
  - Quality-screened data sub-setting
- All code open-sourced as Apache Science Data Analytics Platform (SDAP)



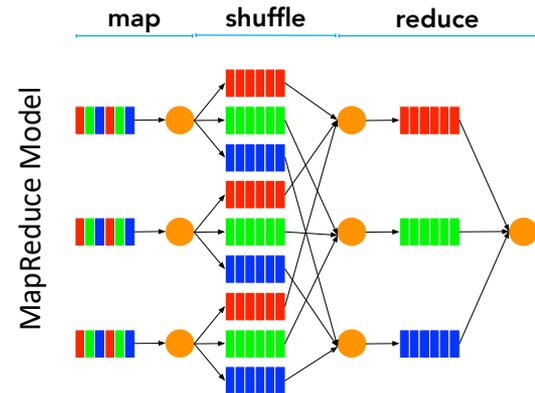
# SDAP Cloud Analytics: NEXUS

- **NEXUS** is a data-intensive analysis solution using a new approach for handling science data to enable large-scale data analysis
  - Streaming architecture for horizontal scale data ingestion
  - Scales horizontally to handle massive amount of data in parallel
  - Provides high-performance geospatial and indexed search solution
  - Provides tiled data storage architecture to eliminate file I/O overhead
  - A growing collection of science analysis webservices



NEXUS' Two-Database Architecture

- **MapReduce**: A programming model for expressing distributed computations on massive amount of data and an execution framework for large-scale data processing on clusters of commodity servers. - J. Lin and C. Dyer, "Data-Intensive Text Processing with MapReduce"
  - **Map**: splits processing across cluster of machines in parallel, each is responsible for a record of data
  - **Reduce**: combines the results from Map processes





# Jupyter Integration

- Python 3 module for easy integration
  - Source code: <https://github.com/apache/incubator-sdap-nexus/tree/master/client>
  - API Documentation: <https://htmlpreview.github.io/?https://github.com/apache/incubator-sdap-nexus/blob/master/client/docs/nexuscli/nexuscli.m.html>
- Exposes HTTP endpoints as functions
- Marshalls function input to JSON
- Unmarshalls server response to objects

```
def time_series(datasets, bounding_box, start_datetime, end_datetime,  
               spark=False)
```

Send a request to NEXUS to calculate a time series.

**datasets** Sequence (max length 2) of the name of the dataset(s)

**bounding\_box** Bounding box for area of interest as a `shapely.geometry.polygon.Polygon`

**start\_datetime** Start time as a `datetime.datetime`

**end\_datetime** End time as a `datetime.datetime`

**spark** Optionally use spark. Default: `False`

**return** List of `TimeSeries` namedtuples

SHOW SOURCE =

```
def daily_difference_average(dataset, bounding_box, start_datetime,  
                             end_datetime)
```

Generate an anomaly Time series for a given dataset, bounding box, and timeframe.

**dataset** Name of the dataset as a String

**bounding\_box** Bounding box for area of interest as a `shapely.geometry.polygon.Polygon`

**start\_datetime** Start time as a `datetime.datetime`

**end\_datetime** End time as a `datetime.datetime`

**return** List of `TimeSeries` namedtuples

# Hydrology Demo

```
In [12]: import requests
import json
import time
import nexusccli
from datetime import datetime

nexusccli.set_target("https://oceanworks.jpl.nasa.gov", use_session=False)

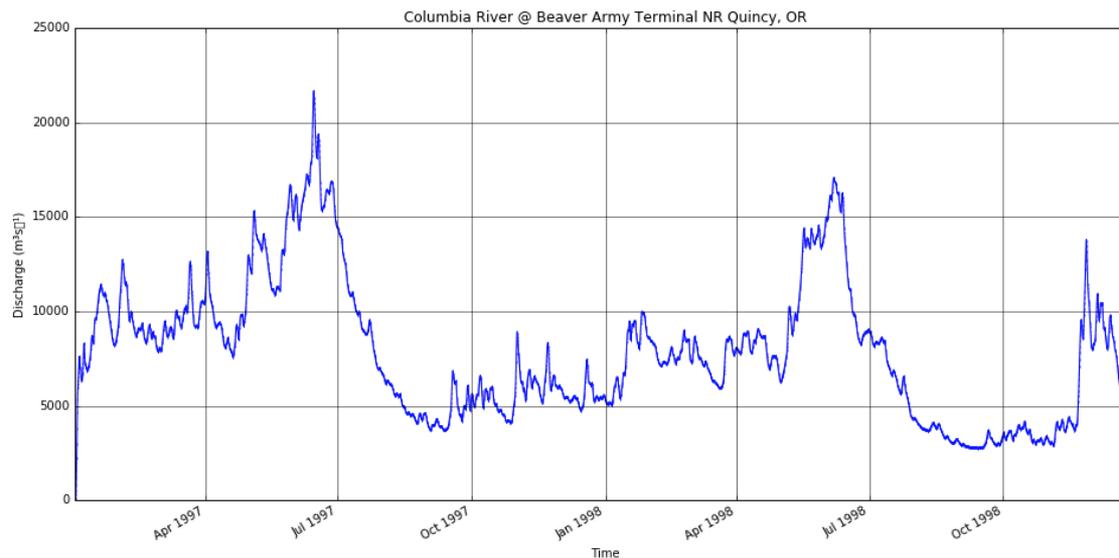
# Observation frequency every 3 hours. About 6,000 total measurements per river
# One point per river, ~600,000 rivers
# ~3.6 Billion measurements

ds = "RAPID_WSWM"
start_time = datetime(1997, 1, 1)
end_time = datetime(1998, 12, 31, 23, 59, 59)
metadataFilter = "rivid_i:24520424"

start = time.perf_counter()
result = nexusccli.subset(ds, None, start_time, end_time, None, metadataFilter)
print("Subsetting took {} seconds".format(time.perf_counter() - start))

show_plot([[point.time for point in result]], # x values
          [[point.variable['variable'] for point in result]], # y values
          'Time', # x axis label
          'Discharge (m³s⁻¹)', # y axis label
          title='Columbia River @ Beaver Army Terminal NR Quincy, OR'
          )
```

Target set to <https://oceanworks.jpl.nasa.gov>  
Subsetting took 0.5723049305379391 seconds



```
Out[12]: <module 'matplotlib.pyplot' from '/usr/local/anaconda/lib/python3.5/site-packages/matplotlib/pyplot.py'>
```

# Hydrology Demo

```
In [6]: import requests
import json
import time
import nexuscli
from datetime import datetime

nexuscli.set_target("https://oceanworks.jpl.nasa.gov", use_session=False)

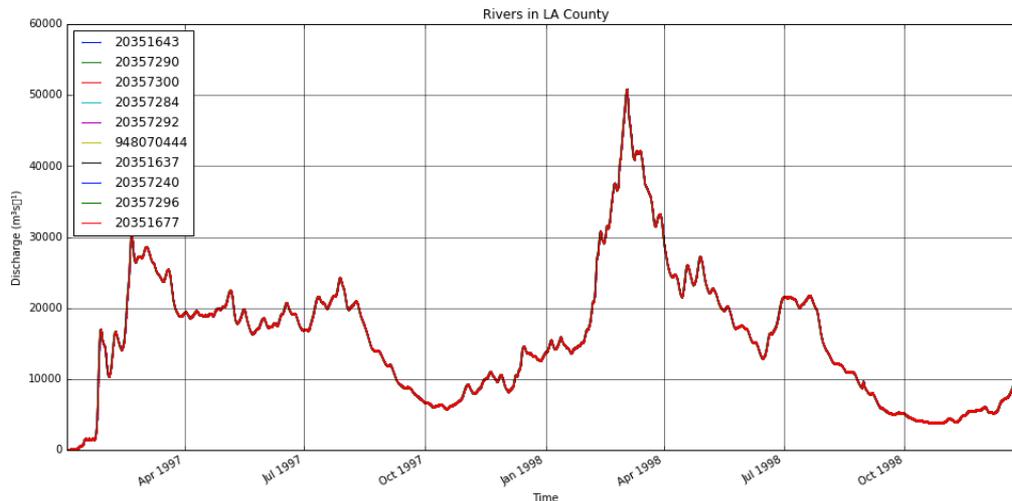
# River IDs for the 10 largest (by max discharge rate) Rivers in LA County
la_county_river_ids = [
    20351643, 20357290, 20357300, 20357284, 20357292,
    948070444, 20351637, 20357240, 20357296, 20351677]

ds = "RAPID_WSWM"
start_time = datetime(1997, 1, 1)
end_time = datetime(1998, 12, 31, 23, 59, 59)
la_county_river_data = list()

start = time.perf_counter()
for river_id in la_county_river_ids:
    metadataFilter = "rivid_i:{}".format(river_id)
    result = nexuscli.subset(ds, None, start_time, end_time, None, metadataFilter)
    la_county_river_data.append(result)
print("Subsetting took {} seconds".format(time.perf_counter() - start))

show_plot([[point.time for point in river] for river in la_county_river_data], # x values
[[point.variable['variable'] for point in river] for river in la_county_river_data], # y values
'Time', # x axis label
'Discharge (m³s⁻¹)', # y axis label
legend=[str(r) for r in la_county_river_ids],
title='Rivers in LA County'
)
```

Target set to <https://oceanworks.jpl.nasa.gov>  
Subsetting took 5.345879919826984 seconds



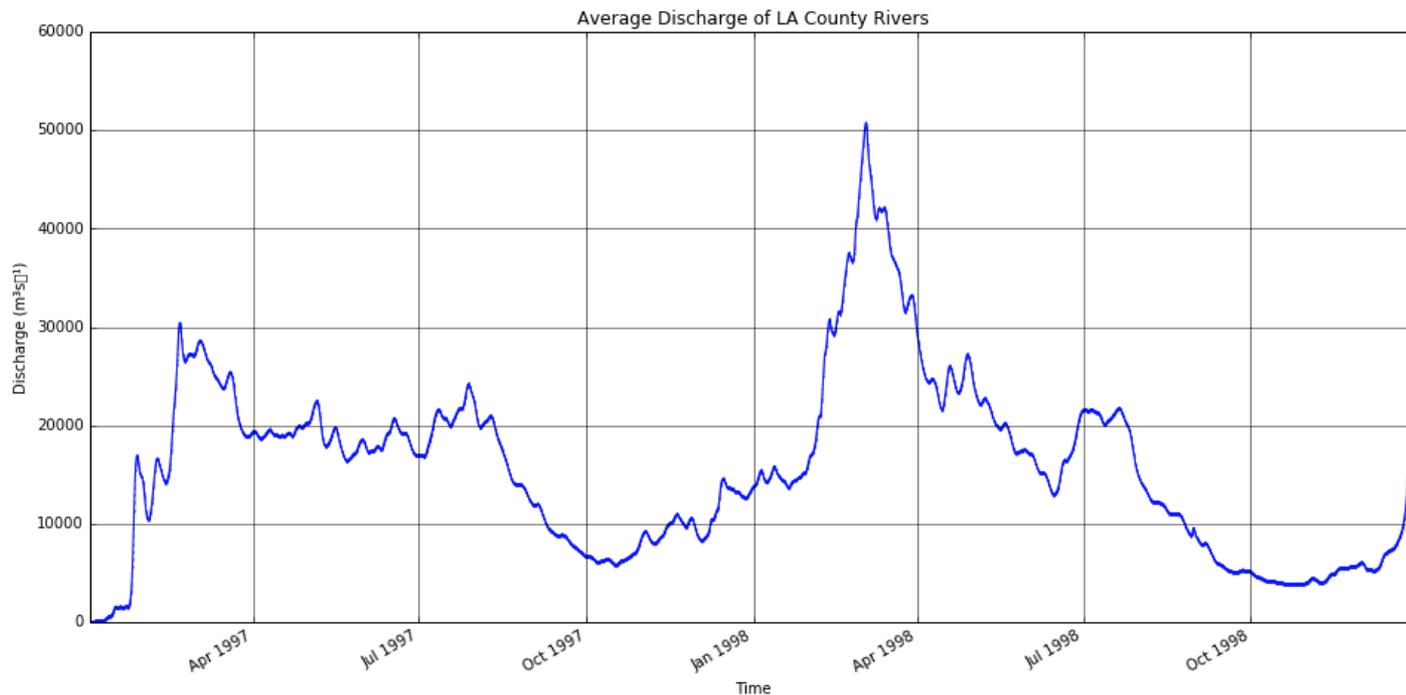
Out[6]: <module 'matplotlib.pyplot' from '/usr/local/anaconda/lib/python3.5/site-packages/matplotlib/pyplot.py'>

# Hydrology Demo

```
In [7]: import numpy

discharge_rates = numpy.array([[point.variable['variable'] for point in river]
                               for river in la_county_river_data])
single_river_time_steps = numpy.array([point.time for point in next(iter(la_county_river_data))])

avg_discharge_rates = numpy.mean(discharge_rates, axis=0)
show_plot([single_river_time_steps], # x values
          [avg_discharge_rates], # y values
          'Time', # x axis label
          'Discharge (m3s-1)', # y axis label
          title='Average Discharge of LA County Rivers'
          )
```



```
Out[7]: <module 'matplotlib.pyplot' from '/usr/local/anaconda/lib/python3.5/site-packages/matplotlib/pyplot.py'>
```



# Hydrology Demo

```
In [8]: import time
import nexuscli
import shapely.wkt
from datetime import datetime

from shapely.geometry import box

nexuscli.set_target("https://oceanworks.jpl.nasa.gov", use_session=False)

la_county_wkt = \
    "POLYGON((-118.9517 34.8233, -117.6462 34.8233, -117.6462 32.7969, -118.9517 32.7969, -118.9517 34.8233))"

# TRMM Data only goes back to beginning of 1998
bbox = shapely.wkt.loads(la_county_wkt)
datasets = ["TRMM_3B42_daily"]
start_time = datetime(1997, 12, 31)
end_time = datetime(1998, 12, 31, 23, 59, 59)

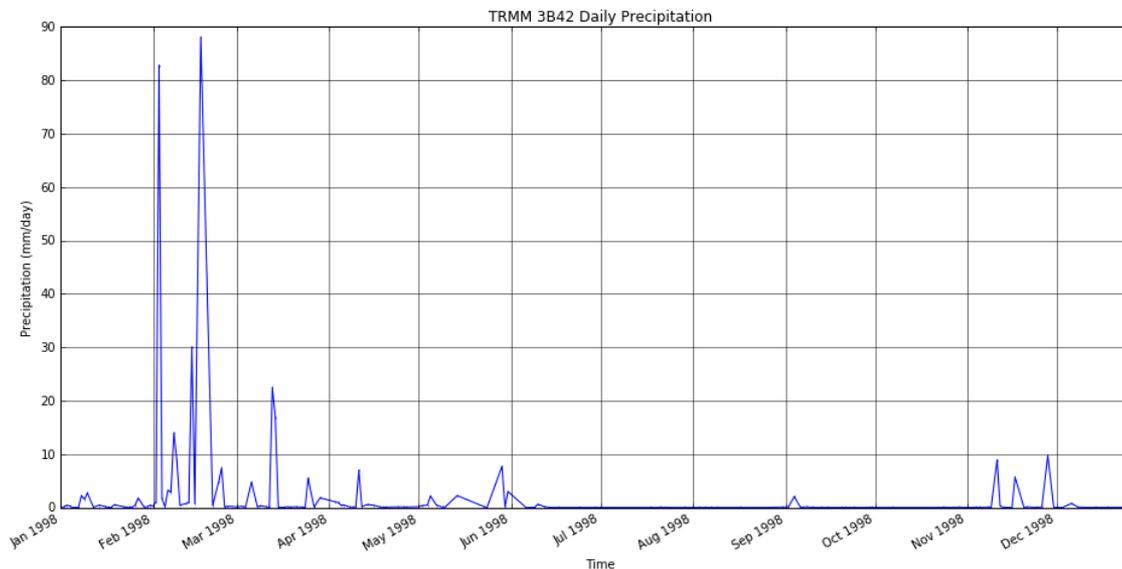
start = time.perf_counter()
ts = nexuscli.time_series(datasets, bbox, start_time, end_time, spark=True)
trmm_ts = ts[0]

print("Time Series took {} seconds to generate".format(time.perf_counter() - start))

show_plot([trmm_ts.time], [trmm_ts.mean], 'Time', 'Precipitation (mm/day)', title='TRMM 3B42 Daily Precipitation')
```

Target set to <https://oceanworks.jpl.nasa.gov>

Time Series took 4.150158584117889 seconds to generate



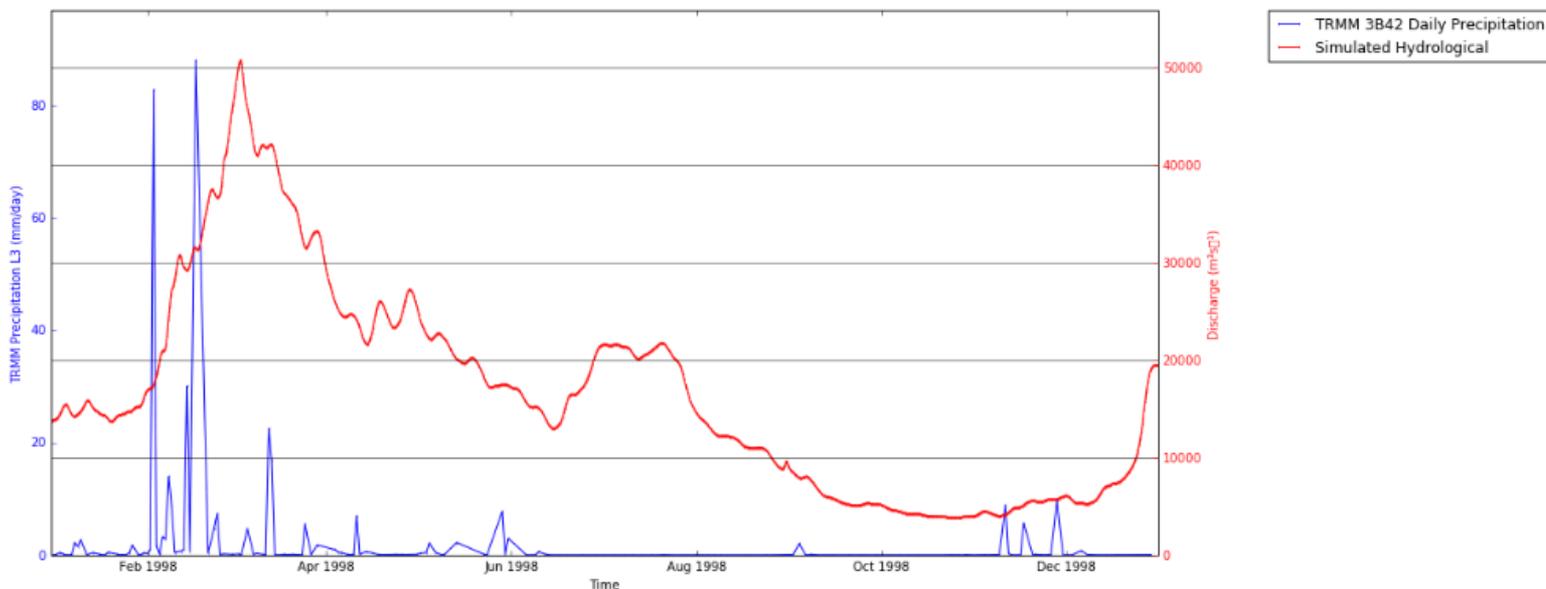
```
Out[8]: <module 'matplotlib.pyplot' from '/usr/local/anaconda/lib/python3.5/site-packages/matplotlib/pyplot.py'>
```

# Hydrology Demo

```
In [16]: import numpy
from pytz import UTC

river_data_1998 = numpy.argwhere(single_river_time_steps > datetime(1997, 12, 31).replace(tzinfo=UTC))

show_plot_two_series(
    trmm_ts.time, single_river_time_steps[river_data_1998],
    trmm_ts.mean, avg_discharge_rates[river_data_1998],
    'Time', 'TRMM Precipitation L3 (mm/day)', 'Discharge (m³s⁻¹)',
    'TRMM 3B42 Daily Precipitation', 'Simulated Hydrological'
)
```





- <http://sdap.apache.org>
- <https://github.com/apache?q=incubator-sdap>
- [greguska@jpl.nasa.gov](mailto:greguska@jpl.nasa.gov)