# Introducing Apache SDAP (Incubating)

## An Integrated Data Analytic Center for Big Science Problems


Dr. Lewis John McGibbney

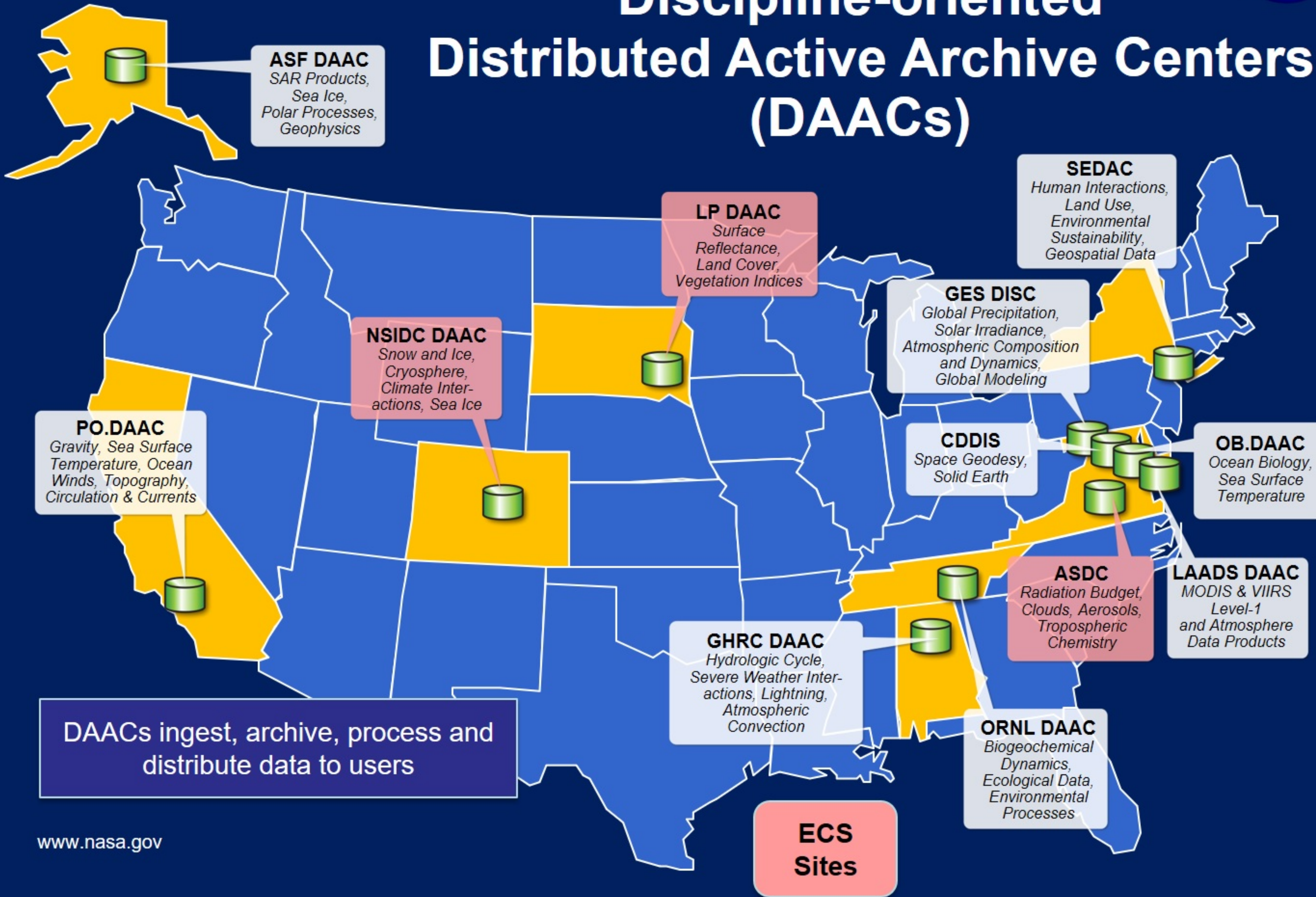
Jet Propulsion Laboratory
California Institute of Technology

## Agenda

1. Background and Project Motivation
2. Introducing the Apache SDAP (Incubating) Project
3. Use Cases
4. Conclusion

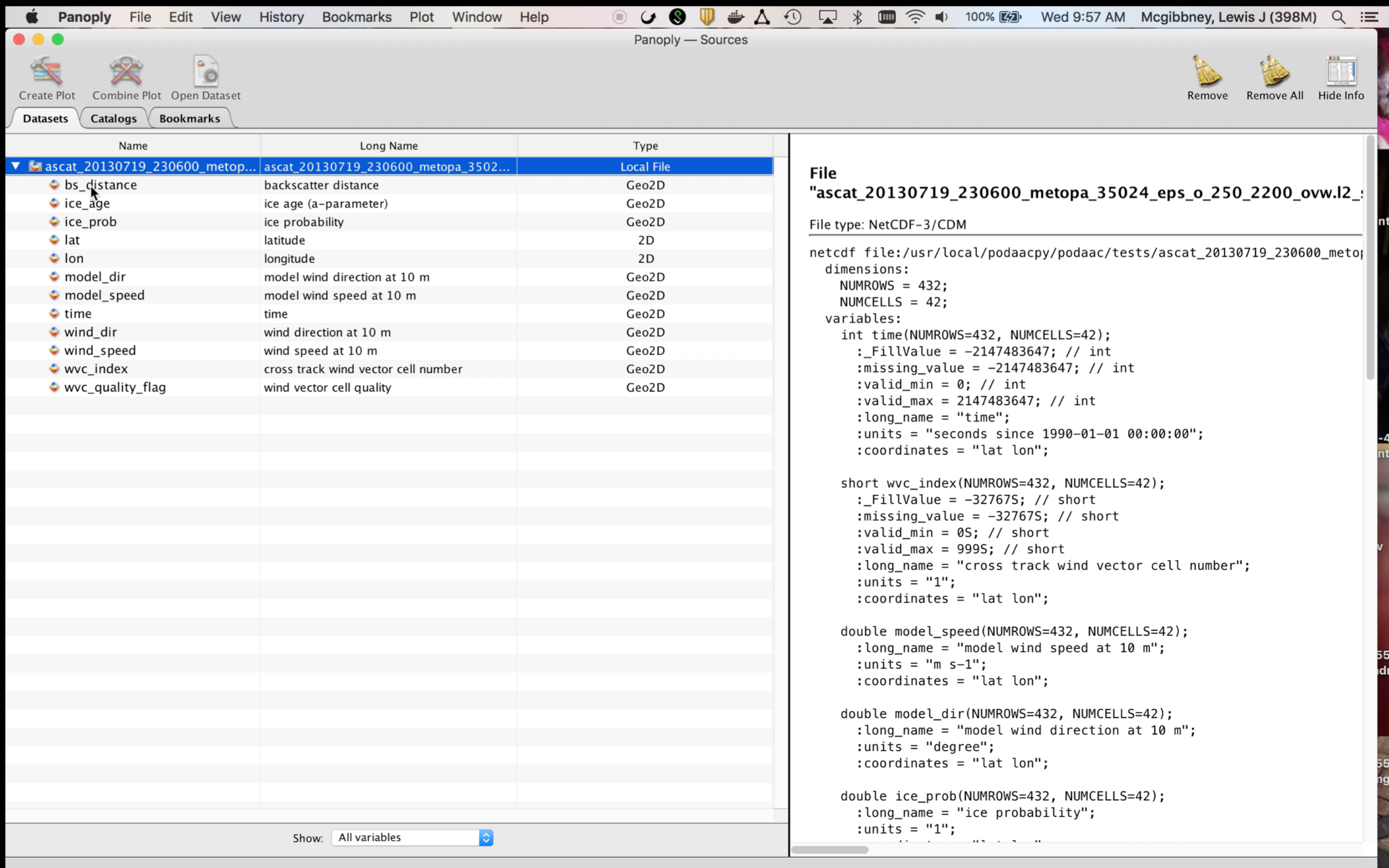
National Aeronautics and Space Administration

# Discipline-oriented Distributed Active Archive Centers (DAACs)



# NASA Data Centers

- **Mainly focused on data archival and distribution. Data is write optimized and is traditionally stored in binary array container files (HDF5, NetCDF4) called granules**

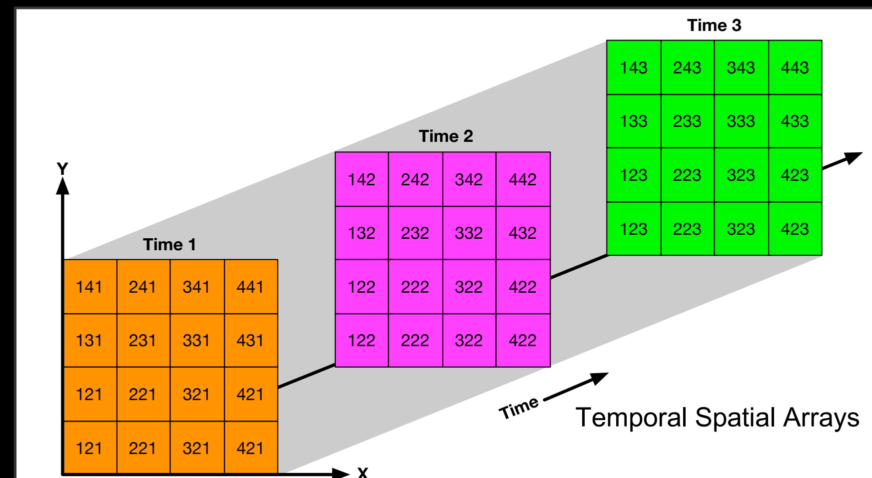


# NASA Data Centers

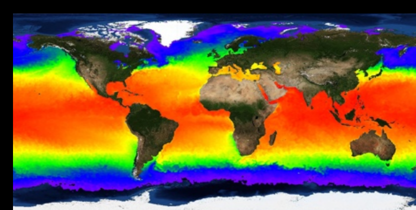
- **Mainly focused on data archival and distribution. Data is write optimized and is traditionally stored in binary array container files (HDF5, NetCDF4) called granules**
- **With additional services**
  - Search and Discovery – faceted, spatial, keyword, ranking, etc.
  - Data subsetting – home grown solutions, OPeNDAP, etc.
  - Visualization – visual discovery, DAAC specific application, NASA-wide applications e.g. Earth Data Search Client, Worldview, etc.
- **Limitations**
  - Lack of interoperability between tools and services: metadata standard, keyword, spatial coverage (0-360 or -180..180), temporal representation, etc.
  - Making sure the most relevant measurements return first
  - Visualization is nice, but it doesn't provide enough information about the ***event/phenomenon*** captured in the image.
  - With large amount of observational data, data centers need to do more than just storing bits
    - "Is the red blob in the middle of Pacific normal at this time of the year?"
    - "Are there any relevant news and publications in relate to what I am looking at?"
    - "What other measurements and/or phenomena relate to the period and location I am looking at?"
    - "I can see the observation from satellite, are there any relevant in situ data I can look at?"

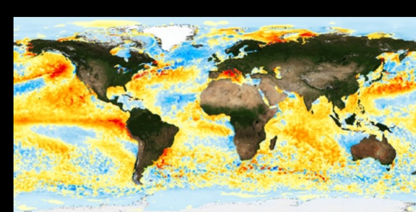
# Traditional Methods for Satellite Data Analysis

Search → Download → Compute



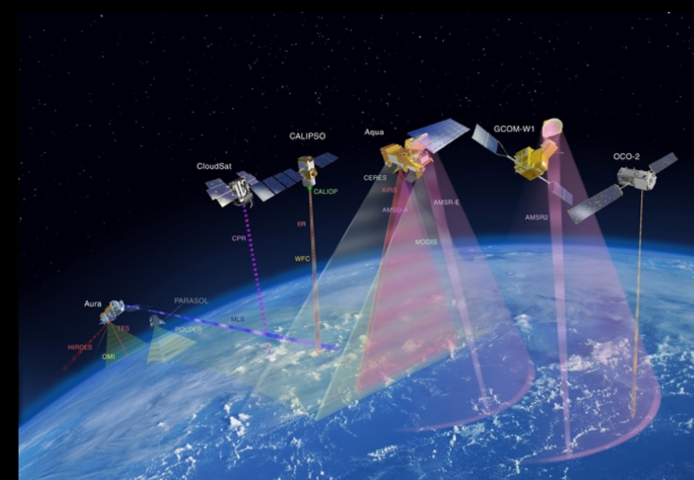

- Depending on the data volume (size and number of files)
- It could take many hours of download – (e.g. 10yr of observational data could yield thousands of files)
- It could take many hours of computation
- It requires expensive local computing resource (CPU + RAM + Storage)
- After result is produced, purge downloaded files



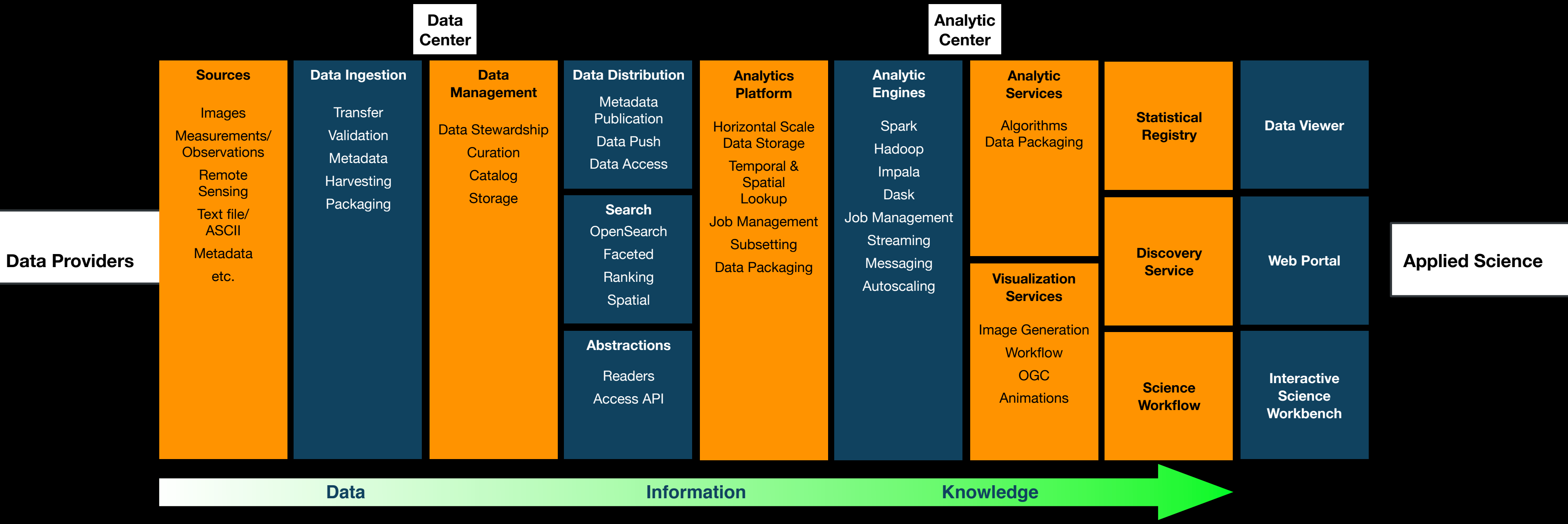


**Observation**

- Traditional methods for data analysis (time-series, distribution, climatology generation) can't scale to handle large volume, high-resolution data. They perform poorly
- Performance suffers when involve large files and/or large collection of files
- A high-performance data analysis solution must be free from file I/O bottleneck

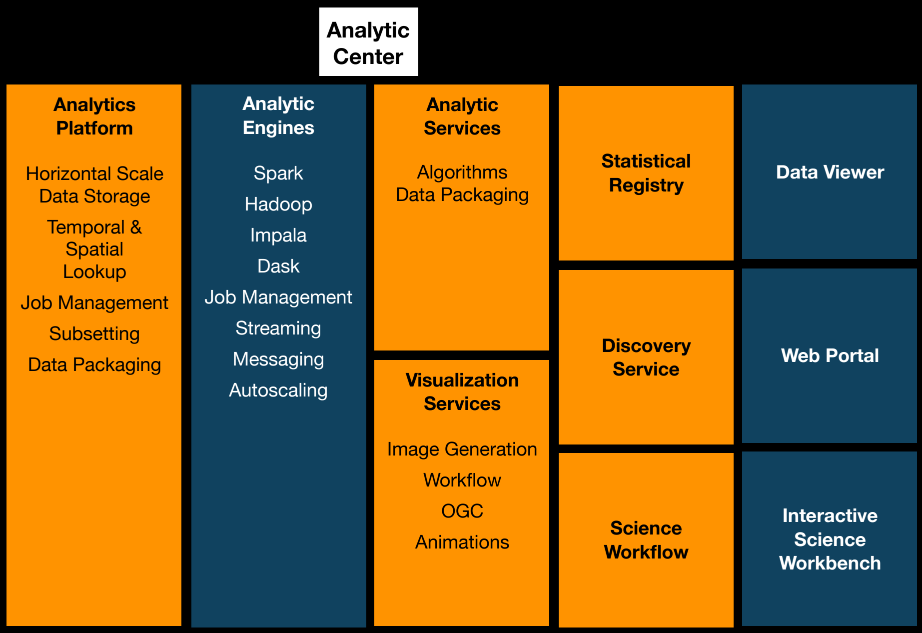


# Enabling Science through Improved Analytics



# Building on the Concept of an Analytic Center

- An **Integrated Data Analytics Center**: an environment for conducting a Science investigation
  - Enables the confluence of resources for that investigation
  - Tailored to the individual study area (ocean, atmospheric, sea level, etc.)
- Harmonizes data, tools and computational resources to permit the research community to focus on the investigation
  - Reduce the data preparation time to something tolerable
  - Catalog of optional resources
  - Semantic-enabled catalog of resources
  - Relevant publications
  - Provide established training data sets of varying resolution
  - Provide effective project confidentiality, integrity and availability
  - Single sign-on and unified financial tracking



# Apache Science Data Analytics Platform (SDAP)

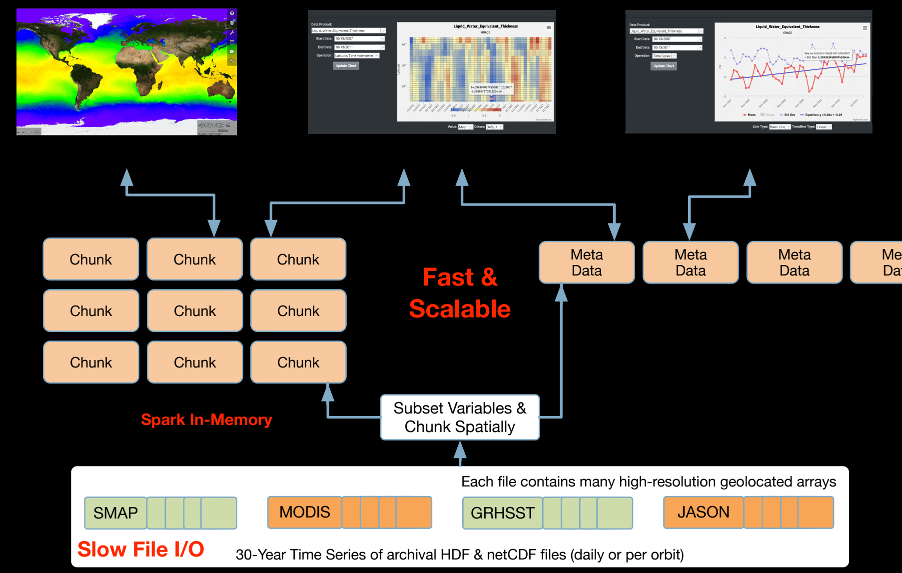
- The **OceanWorks** project establishes an **Integrated Data Analytics Center** at the NASA Physical Oceanography Distributed Active Archive Center (PO.DAAC) for Big Ocean Science
- Focuses on technology integration, advancement and maturity
- Collaboration between JPL, Center for Atmospheric Prediction 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 subsetting
- All code open-sourced as Apache Science Data Analytics Platform (SDAP)

- Entered the Apache Incubator October 2017
- Check us out at http://sdap.apache.org

APACHE INCUBATOR

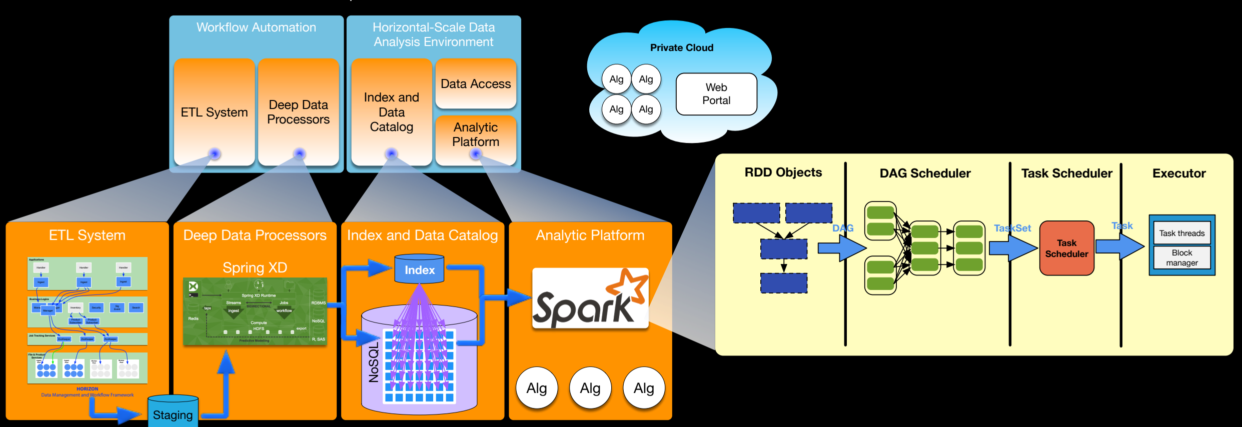
# NEXUS: Scalable Data Analytic Solution

- 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 using Apache Spark: parallel compute, in-memory map-reduce framework
- Pre-Chunk and Summarize Key Variables
    - Easy statistics instantly (milliseconds)
    - Harder statistics on-demand using Spark (in seconds)
    - Visualize original data (layers) on a map quickly (Cassandra store)
- **Algorithms** – Time Series | Latitude/Time Hovmöller| Longitude/Time Hovmöller| Latitude/Longitude Time Average | Area Averaged Time Series | Time Averaged Map | Climatological Map | Correlation Map | Daily Difference Average

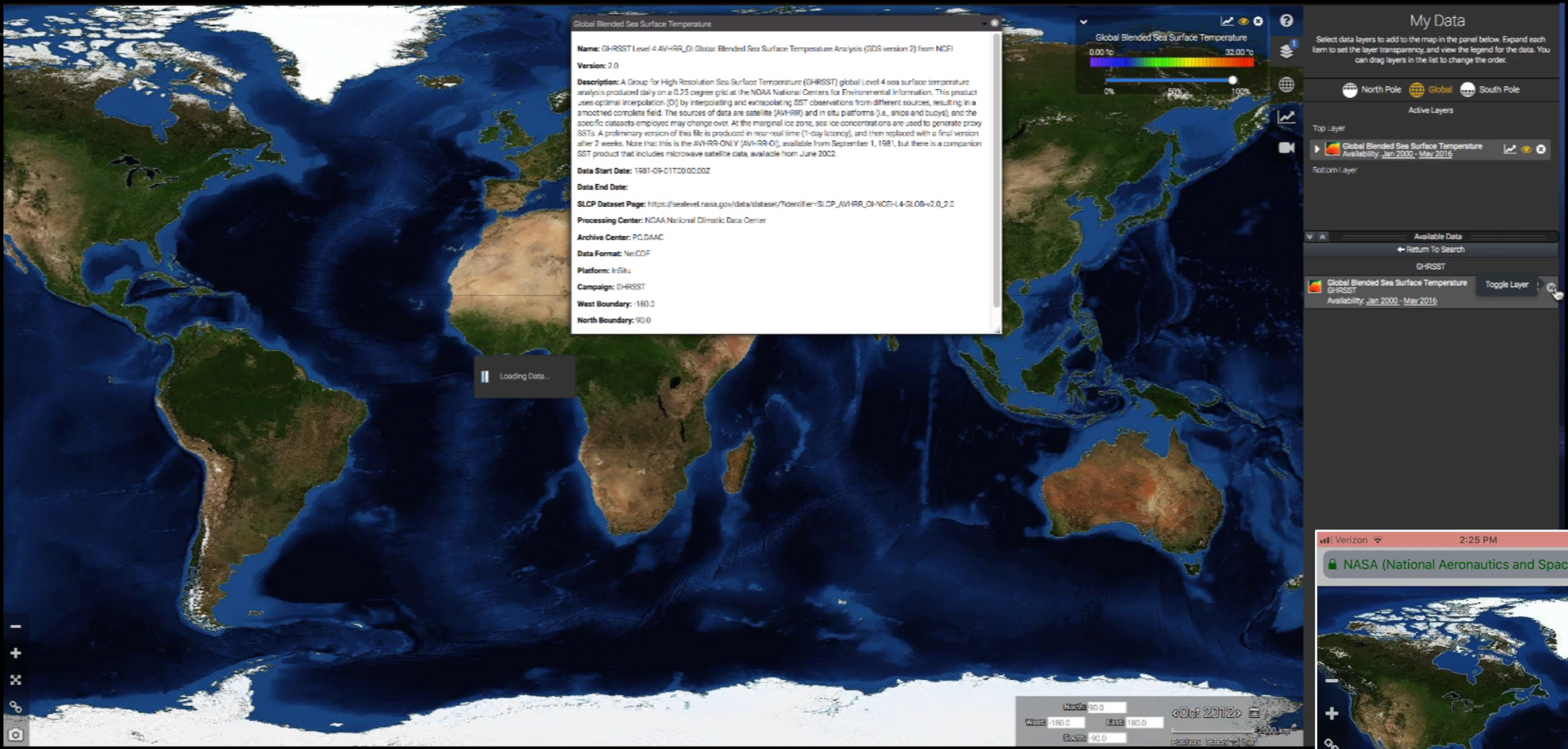



Twin-Database Architecture

NEXUS: The Deep Data Platform



# Oceanographic Analysis On-The-Fly





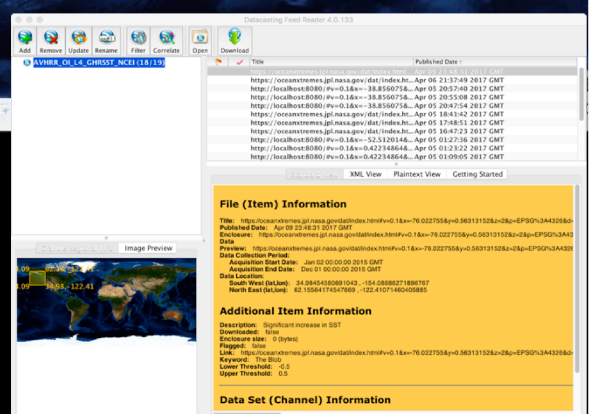
*https://sealevel.nasa.gov/data-analysis-tool*

**Sea Level Change - Data Analysis Tool**

Visualizations | Hydrological Basins | Time Series | Deseason | Data Comparison | Scatter Plot | Latitude/Time Hovmöller | Etc.

# Analyze Ocean Anomaly: "The Blob"


- **Visualize** parameter
- **Compute** daily differences against climatology
- **Analyze** time series area averaged differences
- **Replay** the anomaly and visualize with other measurements
- **Document** the anomaly
- **Publish** the anomaly



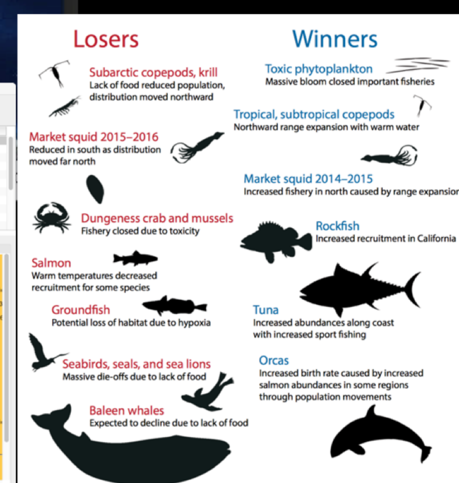
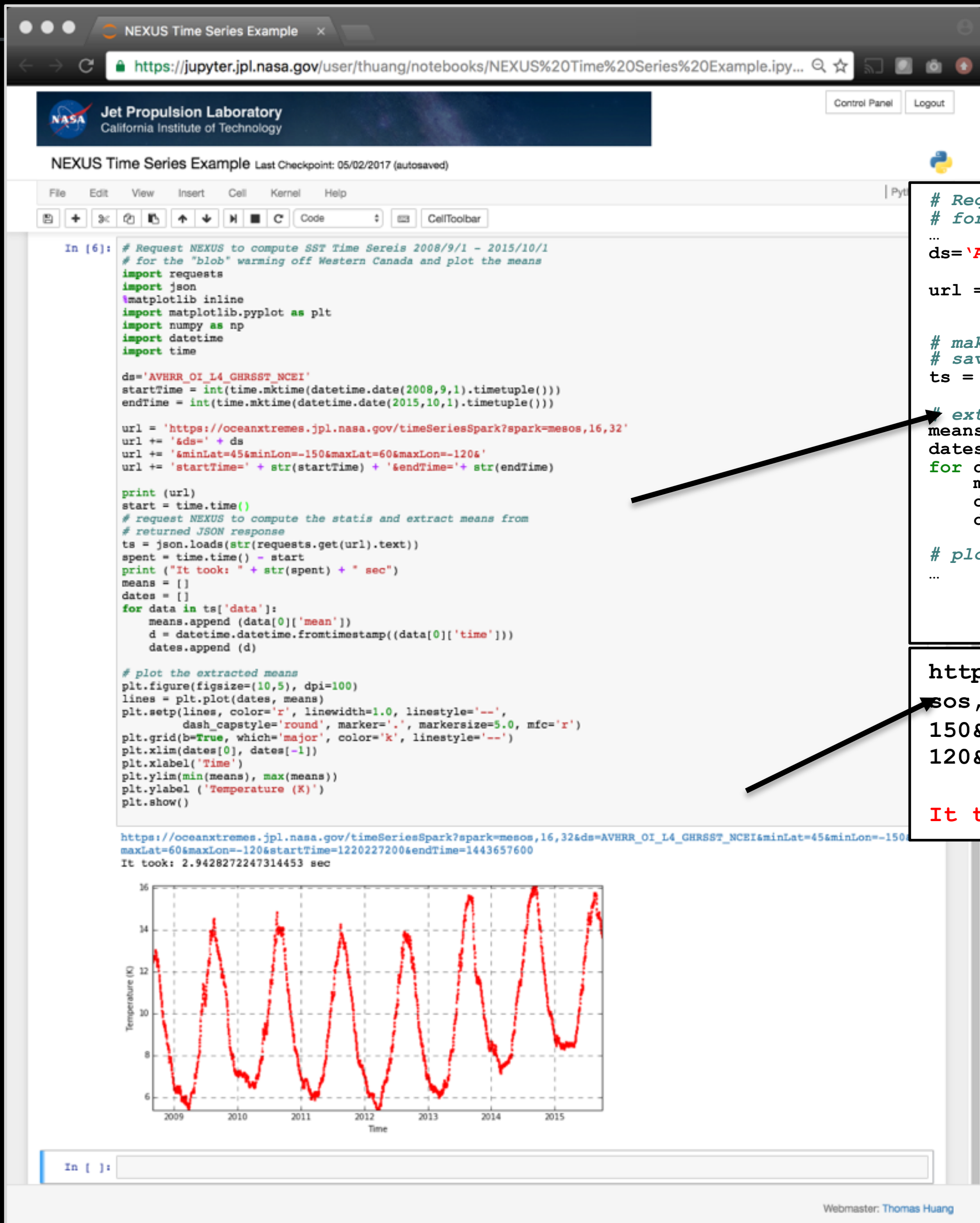




Figure from Cavole L. M., et al. (2016). "Biological Impacts the 2013–2015 Warm-Water Anomaly in the Northeast Pacific: Winners, Losers, a the Future." Oceanography 29.

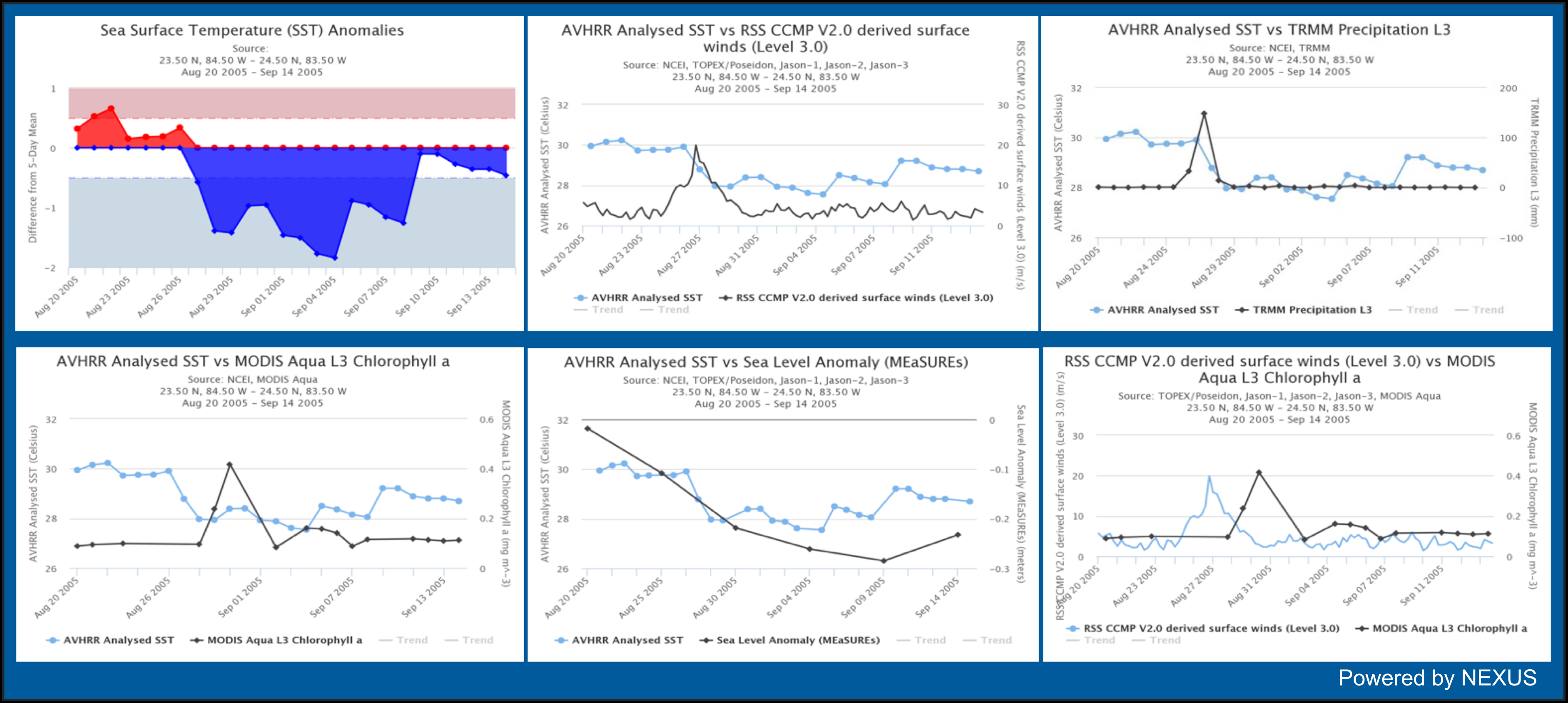
# Enable Scientific Analyses without File Download




```
# Request NEXUS to compute SST Time Series 2008/9/1 - 2015/10/1
# for the "blob" warming off Western Canada and plot the means
…
ds='AVHRR_OI_L4_GHRSST_NCEI'

url = … # construct the webservice URL request


# make request to NEXUS using URL request
# save JSON response in local variable
ts = json.loads(str(requests.get(url).text))

# extract dates and means from the response
means = []
dates = []
for data in ts['data']:
    means.append (data[0]['mean'])
    d = datetime.datetime.fromtimestamp((data[0]['time']))
    dates.append (d)

# plot the result
…
```

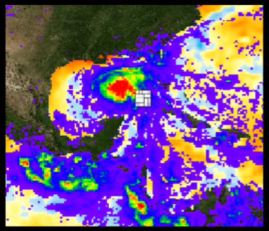
```
https://oceanxtremes.jpl.nasa.gov/timeSeriesSpark?spark=mesos,16,32&ds=AVHRR_OI_L4_GHRSST_NCEI&minLat=45&minLon=-150&maxLat=60&maxLon=-120&startTime=1220227200&endTime=1443657600

It took: 2.9428272247314453 sec
```

# Hurricane Katrina Study




Hurricane Katrina passed to the southwest of Florida on Aug 27, 2005. The ocean response in a 1 x 1 deg region is captured by a number of satellites. The initial ocean response was an immediate cooling of the surface waters by 2 ℃ that lingers for several days. Following this was a short intense ocean chlorophyll bloom a few days later. The ocean may have been "preconditioned' by a cool core eddy and low sea surface height.

The SST drop is correlated to both wind and precipitation data. The Chl-A data is lagged by about 3 days to the other observations like SST, wind and precipitation.

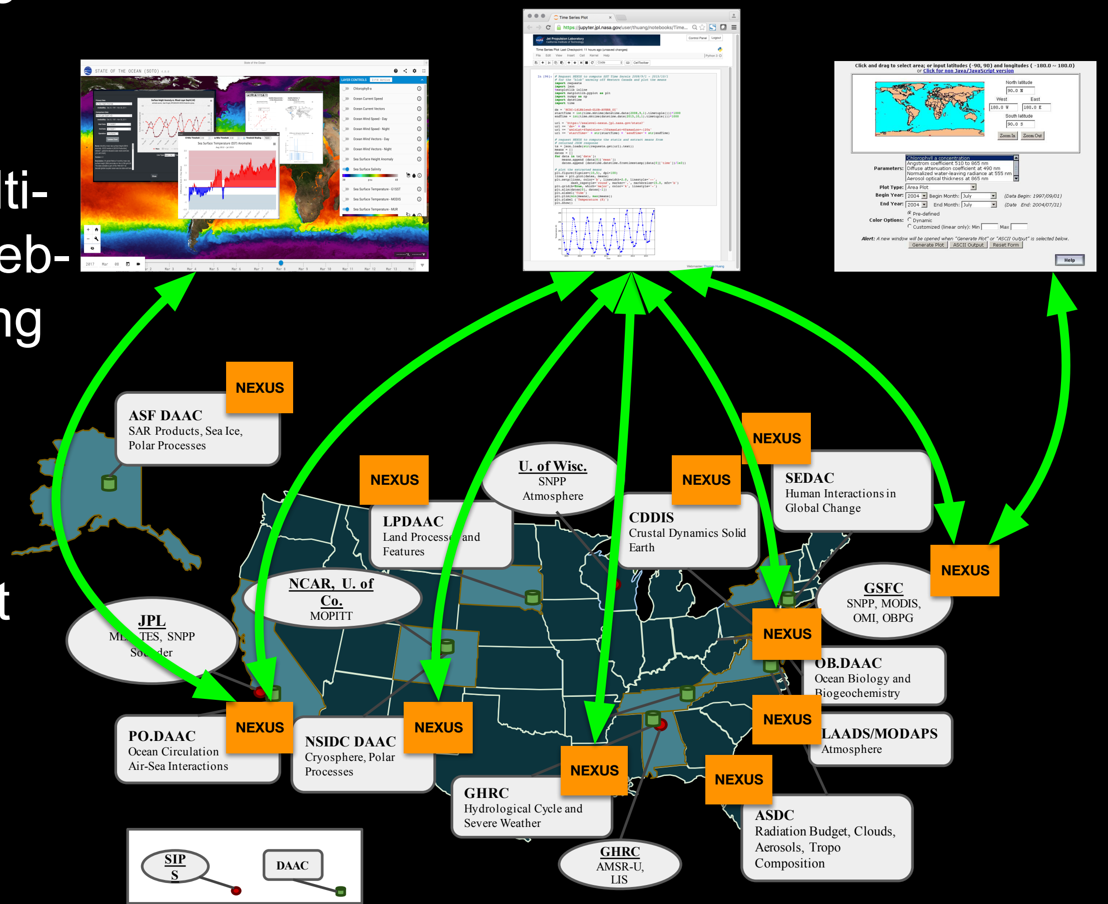


Hurricane Katrina TRMM overlay SST Anomaly


*A study of a Hurricane Katrina–induced phytoplankton bloom using satellite observations and model simulations*
Xiaoming Liu, Menghua Wang, and Wei Shi
JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 114, C03023, doi:10.1029/2008JC004934, 2009

# Multi-Variable Analysis

- Public accessible RESTful analytic APIs where computation is next to the data
- NEXUS as the analytic engine infused and managed by the DAACs on the Cloud
- Researchers can perform multi-variable analysis using any web-enabled devices without having to download files
- Reduce unnecessary egress charges
- An architecture to enable next generation of scientific applications



# Supported Datasets

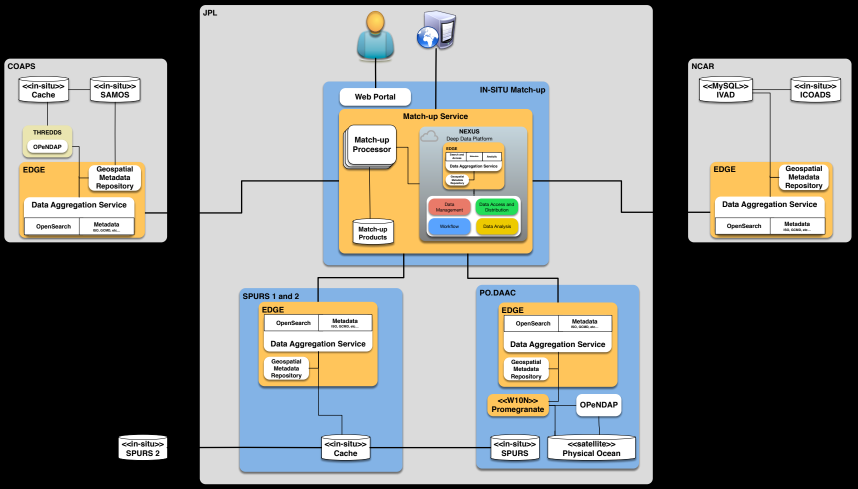
- **Atmosphere**
  - MODIS Aqua Daily L3 Atmospheres, Collection 6, variable Aerosol Optical Depth 550 nm (Dark Target) (MOD08_D3v6)
  - MODIS Terra Daily L3 Atmospheres, Collection 6, variable Aerosol Optical Depth 550 nm (Dark Target) MOD08_D3v6)
  - MODIS Aqua Monthly L3 Atmospheres, Collection 6, variable Aerosol Optical Depth 550 nm (Dark Target) (MOD08_D3v6)
  - MODIS Terra Monthly L3 Atmospheres, Collection 6, variable Aerosol Optical Depth 550 nm (Dark Target) MOD08_D3v6)
- **Chlorophyll**
  - MODIS Aqua Level 3 Global Daily Mapped 4 km Chlorophyll a
- **Estimating the Circulation and Climate of the Ocean (ECCO)**
  - Monthly Mean Version 4 release 2 – Net Surface Fresh-Water Flux, Net Surface Heat Flux, Mixed-Layer Depth, Bottom Pressure, SEAICE Fractional Ice-Covered Area, Free Surface Height Anomaly, SEAICE Effective Snow Thickness, Total Heat Flux, Total Salt Flux
  - Monthly Mean Version 4 release 1 – Net Surface Fresh-Water Flux, Net Surface Heat Flux, Mixed-Layer Depth, Ocean Bottom Pressure, SEAICE Fractional Ice-Covered Area, Free Surface Height Anomaly, SEAICE Effective Snow Thickness, Actual Sublimation Freshwater Flux, Total Heat Flux, Total Salt Flux
- **Gravity**
  - Center for Space Research (CSR) GRACE RL05 Mascon Solutions
  - JPL GRACE Mascon Ocean, Ice, and Hydrology Equivalent Water Height RL05M.1 CRI filtered Version 2
- **Ocean Temperature**
  - GHRSST Level 4 MUR Global Foundation Sea Surface Temperature Analysis (v4.1)
  - GHRSST Level 4 AVHRR_OI Global Blended Sea Surface Temperature Analysis (GDS version 2) from NCEI
  - MODIS Aqua Level 3 SST Thermal IR Daily 4km Nighttime v2014.0
  - MODIS Aqua Level 3 SST Thermal IR Daily 4km Daytime v2014.0

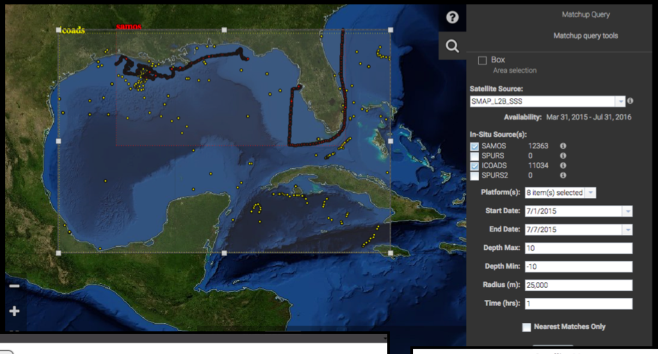
# Supported Datasets

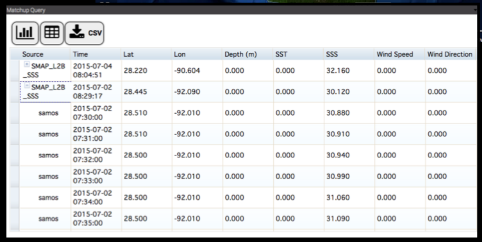
- **Salinity**
  - JPL SMAP Level 2B CAP Sea Surface Salinity V2.0 Validated Dataset
  - JPL SMAP Level 3 CAP Sea Surface Salinity Standard Mapped Image Monthly V3.0 Validated Dataset
- **Sea Surface Height Anomalies (SSHA)**
  - JPL MEaSUREs Gridded Sea Surface Height Anomalies Version 1609
- **Wind**
  - Cross-Calibrated Multi-Platform Ocean Surface Wind Vector L3.0 First-Look Analyses
- **Precipitation (non-ocean data)**
  - TRMM (TMPA) Precipitation L3 1 day 0.25 degree x 0.25 degree V7 (TRMM_3B42_Daily) at GES DIS
  - TRMM (TMPA-RT) Precipitation L3 1 day 0.25 degree x 0.25 degree V7 (TRMM_3B42_RT) at GES DISC
- **In Situ**
  - Shipboard Automated Meteorological and Oceanographic System (SAMOS)
  - International Comprehensive Ocean-Atmosphere Data Set (ICOADS) Release 3, Individual Observations
  - Salinity Process in the Upper Ocean Regional Study – 1 (SPURS1)
  - Salinity Process in the Upper Ocean Regional Study – 2 (SPURS2)
  - Global gridded NetCDF Argo only dataset produced by optimal interpolation (salinity variables)
  - Global gridded NetCDF Argo only dataset produced by optimal interpolation (temperature variables)

# In-situ Satellite Data Matchup

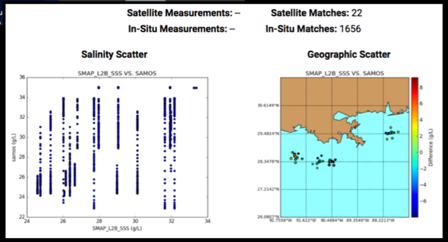
- Distributed Oceanographic Matchup Service (DOMS)
- Typically data matching is done using one-off programs developed at multiple institutions
- A primary advantage of DOMS is the reduction in duplicate development and man hours required to match satellite/in situ data
  - Removes the need for satellite and in situ data to be collocated on a single server
  - Systematically recreate matchups if either in situ or satellite products are re-processed (new versions), i.e., matchup archives are always up-to-date.
- In situ data nodes at JPL, NCAR, and FSU operational.
- Provides data querying, subset creation, match-up services, and file delivery operational.
- Plugin architecture for in situ data source using EDGE, a open source implementation of Open Search



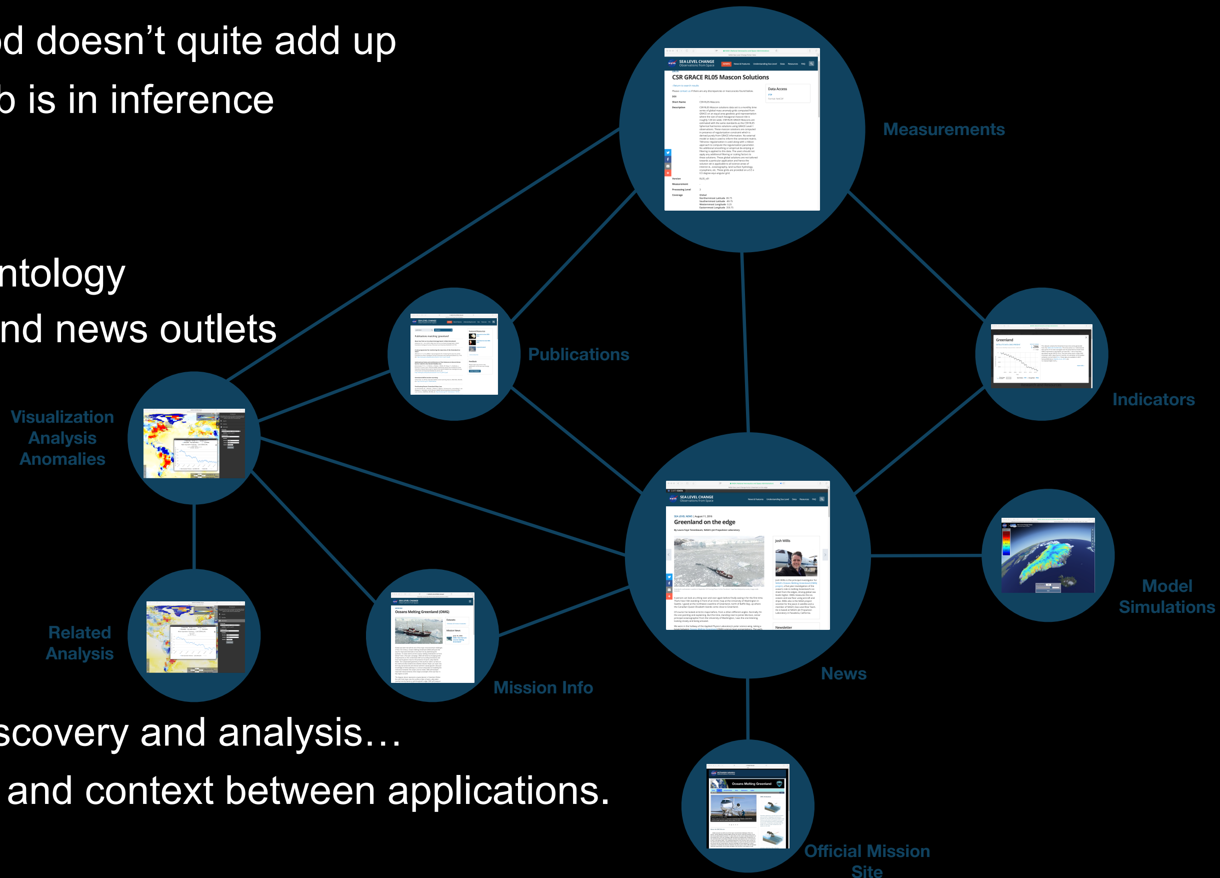


Matchup Query

| Source | Time | Lat | Lon | Depth (m) | SST | SSS | Wind Speed | Wind Direction |
|---|---|---|---|---|---|---|---|---|
| SMAP_L2B_SSS | 2015-07-04 08:04:51 | 28.220 | -90.604 | 0.000 | 0.000 | 32.160 | 0.000 | 0.000 |
| SMAP_L2B_SSS | 2015-07-02 08:29:17 | 28.445 | -92.090 | 0.000 | 0.000 | 30.120 | 0.000 | 0.000 |
| samos | 2015-07-02 07:30:00 | 28.510 | -92.010 | 0.000 | 0.000 | 30.880 | 0.000 | 0.000 |
| samos | 2015-07-02 07:31:00 | 28.510 | -92.010 | 0.000 | 0.000 | 30.910 | 0.000 | 0.000 |
| samos | 2015-07-02 07:32:00 | 28.500 | -92.010 | 0.000 | 0.000 | 30.940 | 0.000 | 0.000 |
| samos | 2015-07-02 07:33:00 | 28.500 | -92.010 | 0.000 | 0.000 | 30.990 | 0.000 | 0.000 |
| samos | 2015-07-02 07:34:00 | 28.500 | -92.010 | 0.000 | 0.000 | 31.060 | 0.000 | 0.000 |
| samos | 2015-07-02 07:35:00 | 28.500 | -92.010 | 0.000 | 0.000 | 31.090 | 0.000 | 0.000 |




# Tackling Information Discovery and Retrieval

- **Search** is looking for something you expect to exist
  - Information tagging
  - Indexed search technologies like Apache Solr or ElasticSearch
  - The solution is pretty straightforward
- **Discovery** is finding something new, or in a new way
  - This is non-trivial
  - Traditional ontological method doesn't quite add up
  - The strength of semantic web is in inference
  - Need method involves
    - Dynamic data ranking
    - Dynamic update to the ontology
    - Mining user interaction and news outlets
- **Relevancy** is
  - Domain-specific
  - Personal
  - Temporal
  - Dynamic
- **Interoperability** is
  - Enabling linkage between discovery and analysis…
  - Smart handoff of information and context between applications.

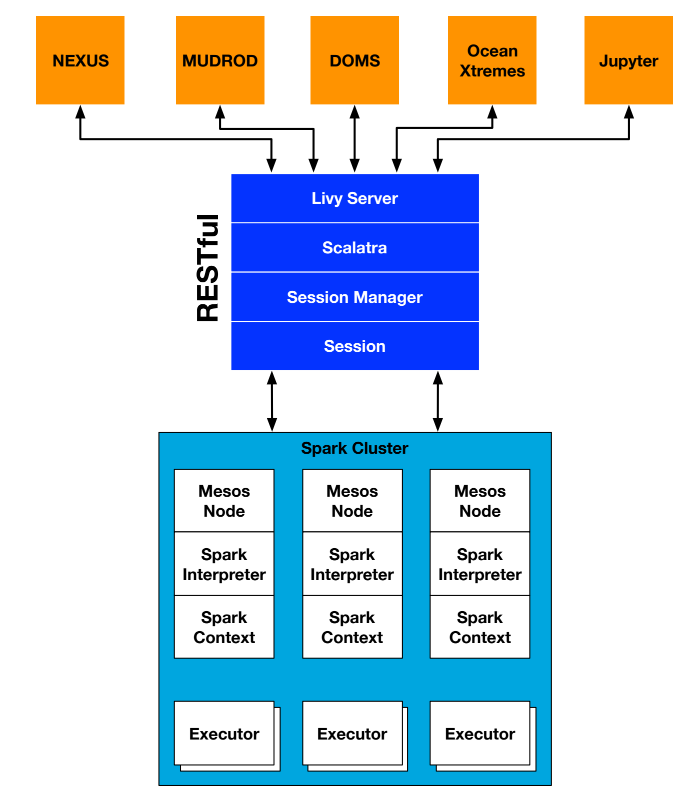


Jet Propulsion Laboratory
California Institute of Technology

# Recent Developments

# Architecture for Apache Spark Resource Sharing

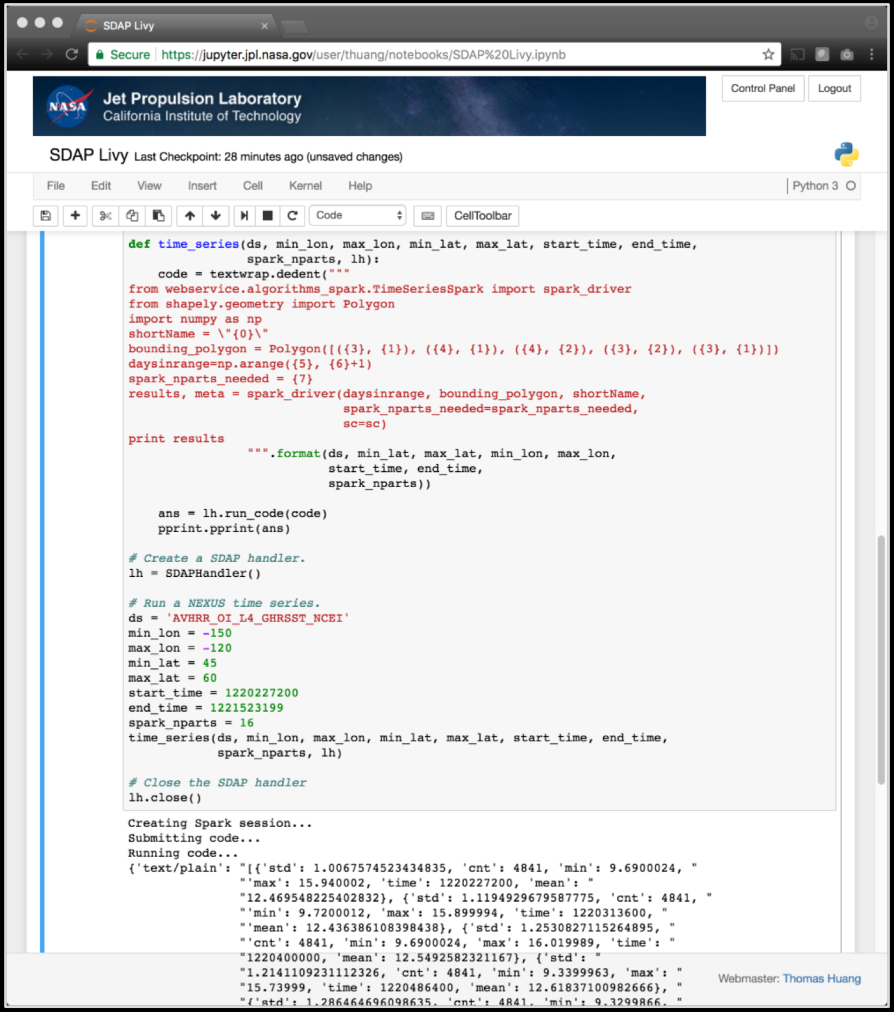
- Apache Spark has become the de facto framework for many data analytics problems. Spinning new Spark cluster for each service is undesirable
- Too many cluster and very costly, since Apache Spark recommends high memory machine instances
- Looking at the Amazon's EMR model. It is designed to be a job execution solution, and the jobs could from different applications
- Apache Livy provides a RESTful interface to Apache Spark cluster. It is a drop-in service to enable applications to interact with Spark cluster using RESTful API.
- Spark-enabled applications can use Livy to interface with a shared Spark cluster



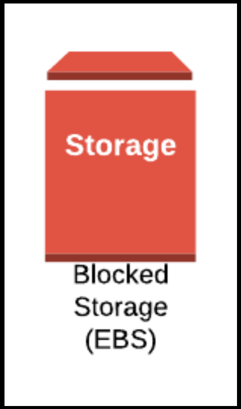

Architecture for Sharing Spark Cluster

# Option for Clients to Push Code

- Using RESTful API, researchers can now push ad hoc map-reduce algorithm in Python to execute by our Spark cluster
- It provides a flexible environment for researchers to experiment with their algorithms and our data, without having to deal with the complexity of Cloud and job management



# Addressing Storage Issues for Analytical Data




$0.045/GB-month
**$47,186/PB-month**

v.s.


$0.021/GB-month
**$22,020/PB-month**

NEXUS supports 2 types of cloud storage

- Blocked storage (e.g. Amazon EBS), Attach to computing node. Generally faster
  - Cassandra and ScyllaDB
- Object storage (e.g. Amazon S3), Independent storage service. Highly scalable

For a data center, not all data need to be served on fast storage, Object storage provides a better, scalable alternative

# Stream-based Ingestion Workflow Architecture

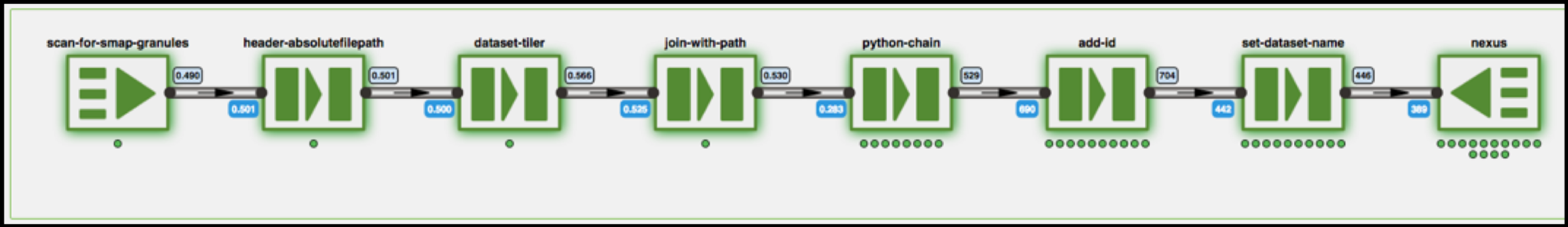
- Data streaming architecture
  - Applications are connected to form ingestion streams
  - Configurable to handle different datasets
    - Multiple tiling algorithms
    - Support L2 swath data
    - Support gridded data
  - Scalable across compute resources
  - Resilient to failure
  - ESDS-RFC-028v1.1

```
def filter_empty_tiles(self, tile):
  # Only supply data if there is actual values in the tile
  if tile.data.size - numpy.count_nonzero(numpy.isnan(tile.data)) > 0:
    yield tile.data
  else:
    print "Discarding data %s from %s because it is empty" % (tile.section_spec, tile.granule)
```

Pluggable validation checkers

```
def transform(self, tile):

  # Subtract 360 if the longitude is greater than 180
  tile.data.longitudes[longitudes > 180] -= 360

  yield tile.data
```

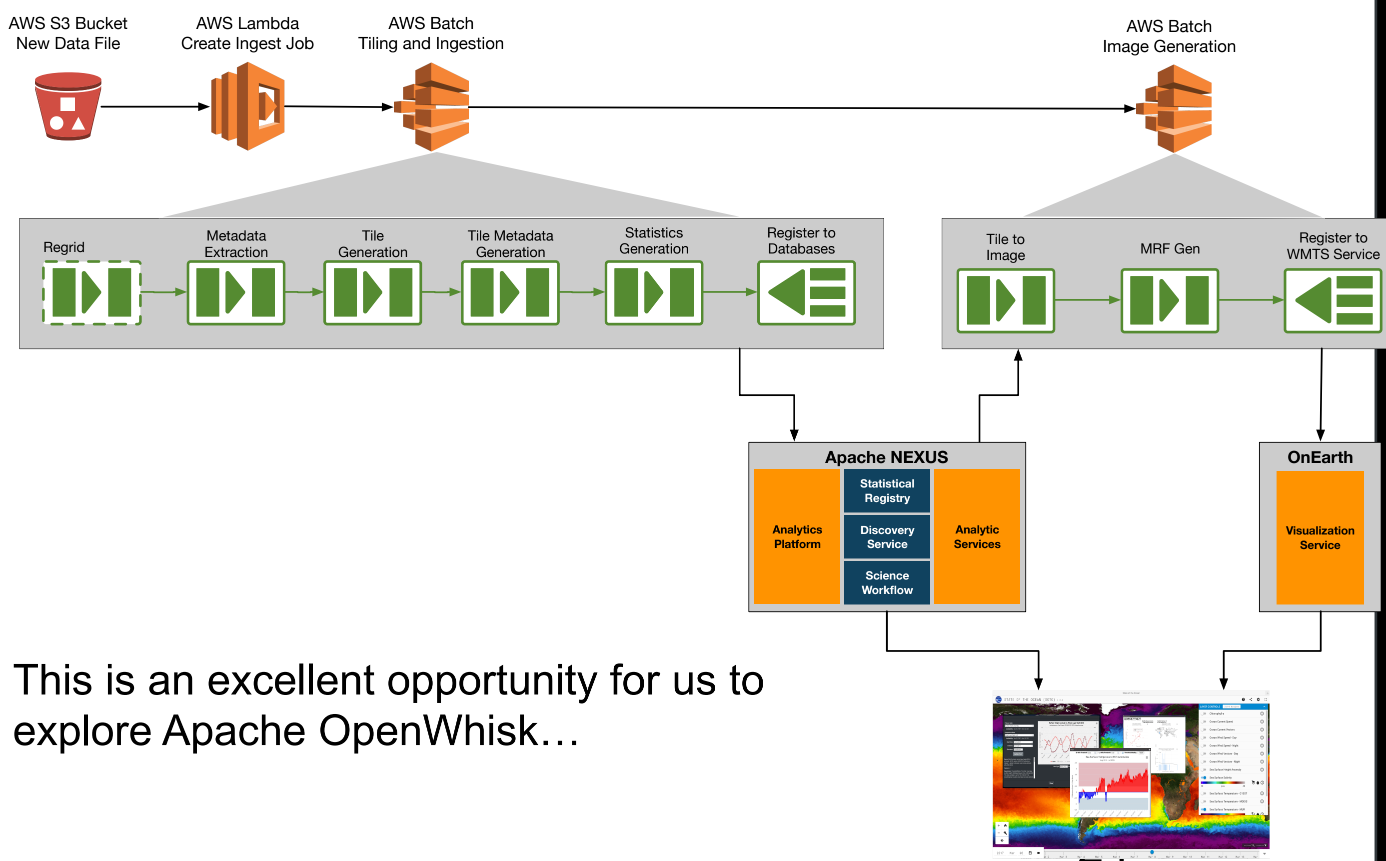
Data translation




Stream for AVHRR_OI-NCEI-L4-GLOB-v2.0 Sea Surface Temperature

# Serverless ingestion

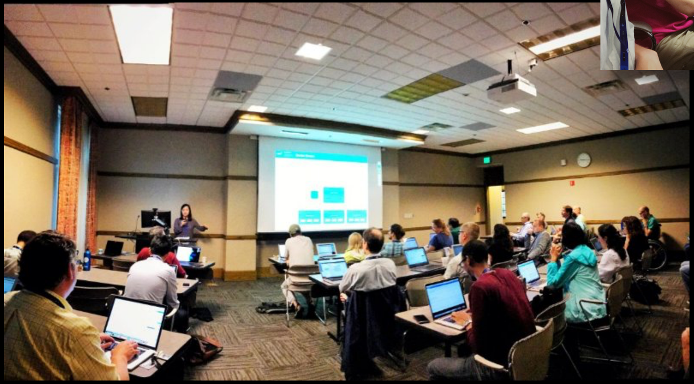
- The original ingestion workflow requires several EC2 instances and attached EBS storage. Amazon bills active EC2 and storage even if there is no data to be processed
- Cost-saving serverless architecture using AWS S3 Bucket trigging AWS Lambda job to create AWS Batch jobs

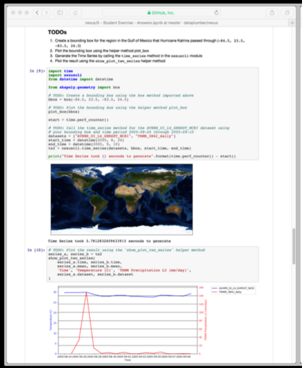



This is an excellent opportunity for us to explore Apache OpenWhisk…

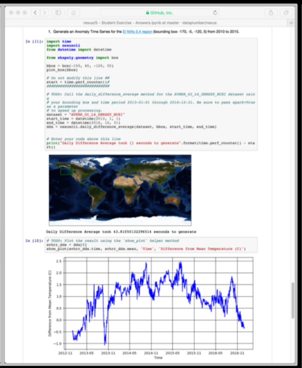
# Community Development and Support

- Develop in the open
- Working with the Apache Incubator
- Target Apache top-level project by Summer 2019.
- Public hands-on workshops
- Organize technical sessions at conferences
- Invited speaker and panelist
- Lead Editor: 2018 Wiley Book on **Big Earth Data Analytics in Earth, Atmospheric and Ocean Sciences**

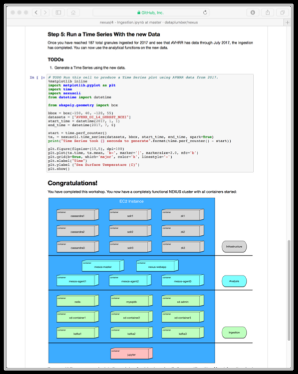




Analyze Hurricane Katrina by comparing SST and TRMM time series



Generate daily difference average "The Blob" is an oceanographic anomaly



Each participant deployed 3 computing clusters, a total of 24 containers on EC2

# Summary

- Traditional method for scientific research (search, download, local number crunching) is unable to keep up… put simply it is unsustainable.
- How much speed and storage can you afford?
- Think beyond archive and file downloads
- Investment in data and computational sciences
- Data Centers might want to be in the business of Enabling Science!
- Connected information enables discovery
- Community developed solution through open sourcing
- Thanks to the NASA ESTO/AIST and Sea Level Rise programs, and the NASA ESDIS project
- OceanWorks infusion 2018 – 2019
  - Watch for changes to the Sea Level Change Portal
    - Even faster analysis capabilities
    - More variety of measurements – satellites, in situ, and models
    - Event more relevant recommendations
  - NASA's Physical Oceanography Distributed Active Archive Center (PO.DAAC)
    - More than just pretty pictures. DAAC applications will have new analytic capabilities.
- Lead Editor: 2018 Wiley Book on **Big Earth Data Analytics in Earth, Atmospheric and Ocean Sciences**


**Thomas Huang**
Jet Propulsion Laboratory
California Institute of Technology

**JPL Team**
Ed Armstrong, Frank Greguska, Joseph Jacob, Lewis McGibbney,
Nga Quach, Vardis Tsontos, and Brian Wilson

**Florida State University Team**
Shawn Smith, Mark A. Bourassa, Jocelyn Elya

**National Center for Atmospheric Research Team**
Steve J. Worley, Tom Cram, Zaihua Ji

**George Mason University Team**
Chaowei (Phil) Yang, Yongyao Jiang, and Yun Li

**Thank you very much**

**Questions?**

Catch us on the mailing lists

dev@sdap.apache.org

Contact me directly at

Lewis.j.mcgibbney@jpl.nasa.gov
lewismc@apache.org

# Acknowledgments

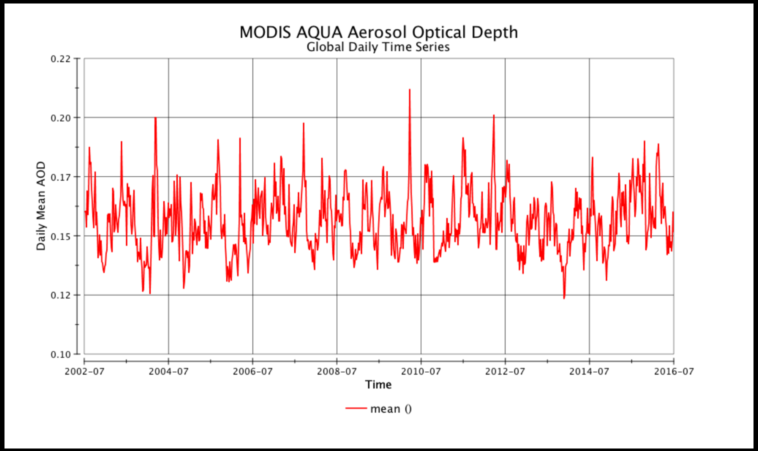
Funding support under the NASA Advanced Information Systems and Technology (AIST) program OceanWorks project (PI T. Huang, JPL)

# Backup

# Benchmarking

# NEXUS Performance: GIOVANNI (v4) vs. Custom Spark vs. AWS EMT

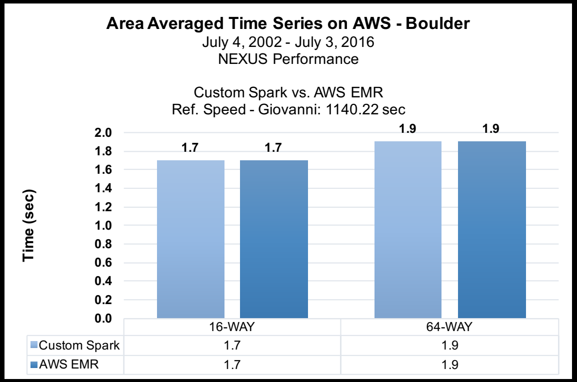
**Dataset**: MODIS AQUA Daily
**Name**: Aerosol Optical Depth 550 nm (Dark Target) (MYD08_D3v6)
**File Count**: 5106
**Volume**: 2.6GB
**Time Coverage**: July 4, 2002 – July 3, 2016

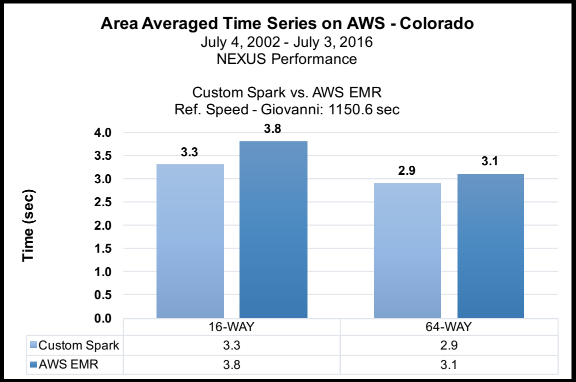


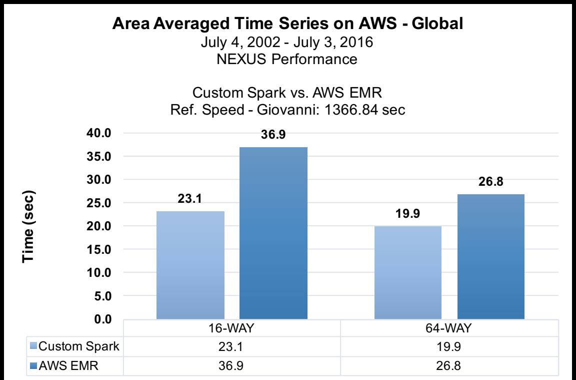

**Giovanni**: A web-based application for visualize, analyze, and access vast amounts of Earth science remote sensing data without having to download the data.

- Represents current state of data analysis technology, by processing one file at a time
- Backed by the popular NCO library.  Highly optimized C/C++ library

**AWS EMR**: Amazon's provisioned MapReduce cluster



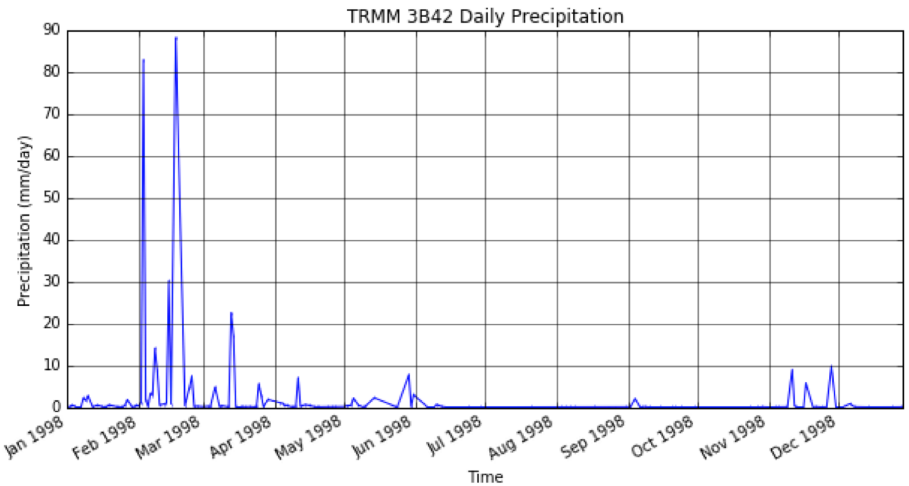

Algorithm execution time. Excludes Giovanni's data scrubbing processing time

# Analysis Examples

# Times Series

In [16]:
```
import time
import nexuscli
import shapely.wkt
from datetime import datetime

from shapely.geometry import box

nexuscli.set_target("https://oceanworks.jpl.nasa.gov")

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')
```

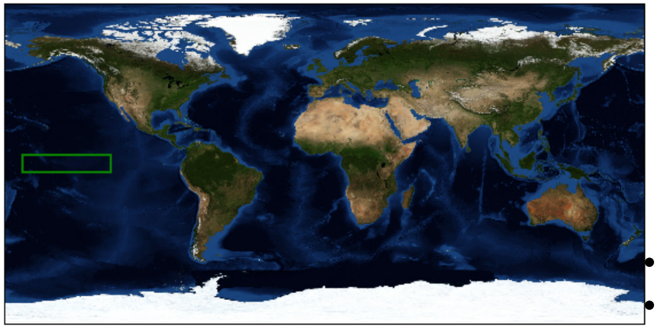
```
Time Series took 0.4333303924649954 seconds to generate
```



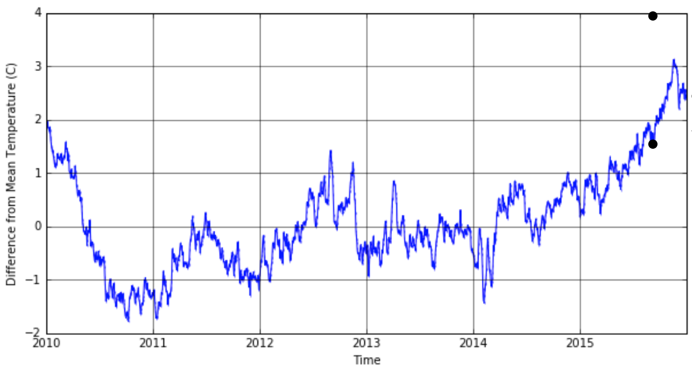

- Time series of precipitation in LA County
- Length of 1 year (1998)

# Difference from Mean

```
In [17]: import time
         import nexuscli
         from datetime import datetime

         from shapely.geometry import box

         # Bounding box for the El Nino 3.4 Region
         bbox = box(-170, -5, -120, 5)
         plot_box(bbox)

         start = time.perf_counter()

         # Time range of interest
         dataset = "AVHRR_OI_L4_GHRSST_NCEI"
         start_time = datetime(2010, 1, 1)
         end_time = datetime(2015, 12, 31)
         # Call server
         dda = nexuscli.daily_difference_average(dataset, bbox, start_time, end_time)

         print("Daily Difference Average took {} seconds to generate".format(time.perf_counter() - start))

         avhrr_dda = dda[0]
         # Plot results!
         show_plot(avhrr_dda.time, avhrr_dda.mean, 'Time', 'Difference from Mean Temperature (C)')
```



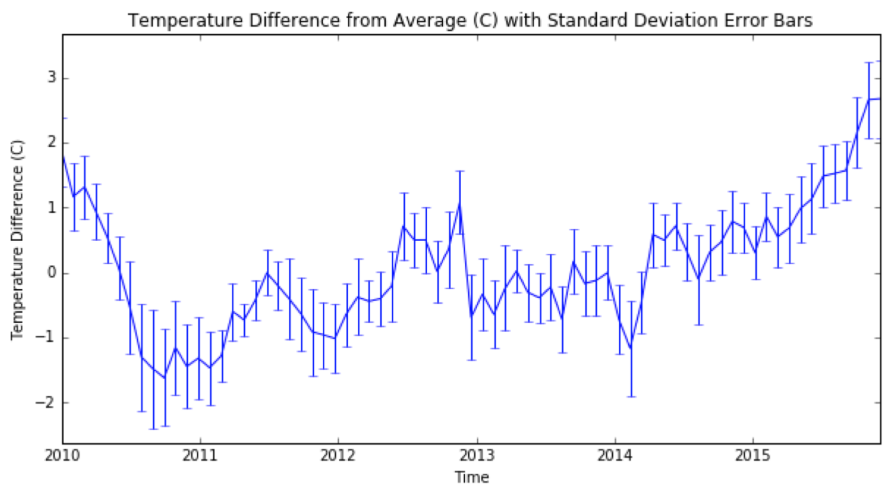
```
Daily Difference Average took 57.92217040248215 seconds to generate
```




- El Nino 3.4 region
- On this plot, average temperature is 0
- Considered El Nino/La Nina event if difference from average is +/- 0.5˚c
- Very strong El Nino/La Nina signals shown here

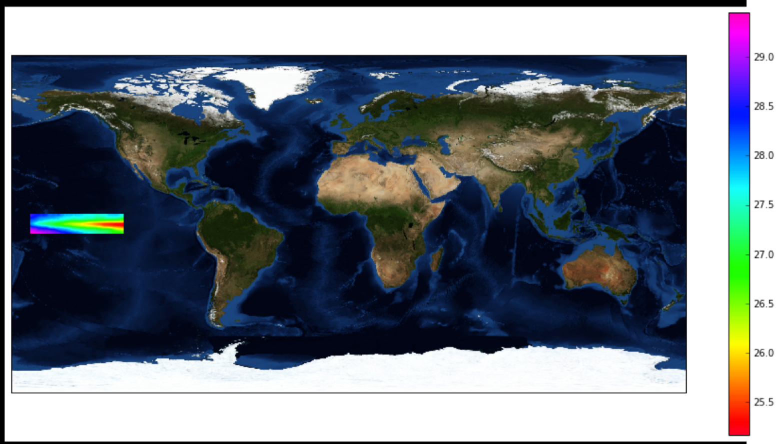
# Difference from Mean with Error Bars

In [32]:
```
# Sampling every 30th data point to reduce plot noise
means, dates, st_ds = avhrr_dda.mean[0::30], avhrr_dda.time[0::30], avhrr_dda.standard_deviation[0::30]

# Plot the extracted means
plt.figure(figsize=(10,5), dpi=100)
lines = plt.errorbar(dates, means, st_ds)
plt.xlim(dates[0], dates[-1])
plt.xlabel('Time')
plt.ylim(min(means)-1, max(means)+1)
plt.ylabel ('Temperature Difference (C)')
plt.title('Temperature Difference from Average (C) with Standard Deviation Error Bars', fontsize=12)
plt.show()
```




- Same as previous but
  - Standard deviation shown as error bars
  - Only every 30$^{th}$ data point is plotted (instead of daily)

# Time Average Map




```
In [2]: import requests
        import json
        import datetime
        import time
        import numpy
        %matplotlib inline
        import matplotlib as mpl
        import matplotlib.pyplot as plt
        from matplotlib.pyplot import imshow
        from shapely.geometry import box
        from mpl_toolkits.basemap import Basemap

        epoch = datetime.datetime.utcfromtimestamp(0)

        # Build the HTTP request
        host = "https://oceanworks.jpl.nasa.gov"

        ds='AVHRR_OI_L4_GHRSST_NCEI'
        bbox = box(-170, -5, -120, 5) #minx, miny, maxx, maxy
        date_format = '%Y-%m-%dT%H:%M:%SZ'
        startTime = int((datetime.datetime(2016,1,1) - epoch).total_seconds())
        endTime = int((datetime.datetime(2016,12,31) - epoch).total_seconds())
        request = "{}/timeAvgMapSpark?ds={}&startTime={}&endTime={}" \
                    "&minLon={}&minLat={}&maxLon={}&maxLat={}&spark=local,16,32" \
            .format(host, ds, startTime, endTime, *bbox.bounds)
        print (request)

        # Send request to server
        response = requests.get(request).json()

        # Parse the response and create an image
        lons = [point['lon'] for point in response['data'][0]]
        lats = [a_list[0]['lat'] for a_list in response['data']]

        my_list = numpy.ndarray((len(lats), len(lons)))
        for x in range(0, len(lats)):
            for y in range(0, len(lons)):
                my_list[x][y] = response['data'][x][y]['avg']

        fig, ax1 = plt.subplots(figsize=(16,8), dpi=100)
        im = ax1.pcolormesh(lons, lats, my_list, vmin=my_list.min(),vmax=my_list.max(), cmap='gist_rainbow')
        fig.colorbar(im, ax=ax1)

        map = Basemap()
        map.bluemarble(scale=0.5)

        https://oceanworks.jpl.nasa.gov/timeAvgMapSpark?ds=AVHRR_OI_L4_GHRSST_NCEI&startTime=1451606400&endTime=1483142400&mi
        nLon=-170.0&minLat=-5.0&maxLon=-120.0&maxLat=5.0&spark=local,16,32
```

- Same region as previous slide
- Shows average values across time range