AIST-14: OceanXtremes
Oceanographic Data-Intensive Anomaly Detection and Analysis Portal

Earth Science Technology Forum

PI: Thomas Huang

Co-Is: Ed Armstrong, George Chang, (Mike) Toshio Chin, and Brian Wilson

Engineers: Kevin Gill, Frank Greguska, Joseph Jacob, and Nga Quach

June 13, 2016

Earth Science Technology Office

OceanXtremes: Oceanographic Data-Intensive Anomaly Detection and Analysis Portal

Objective

Develop an anomaly detection system which identifies items, events or observations which do not conform to an expected pattern

- Mature and test domain-specific, multi-scale anomaly and feature detection algorithms.
- Identify unexpected correlations between key measured variables.

Demonstrate value of technologies in this service:

- Adapted Map-Reduce data mining.
- Algorithm profiling service.
- Shared discovery and exploration search tools.
- Automatic notification of events of interest.

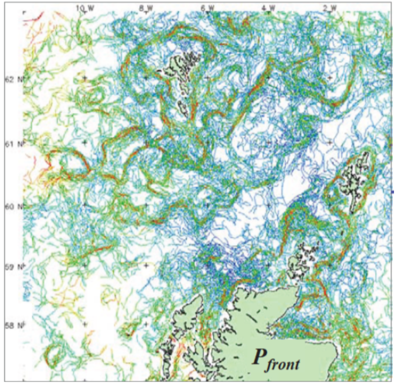


Illustration of future OceanXtremes analysis capability showing sea surface temperature (SST) gradients from AVHRR imagery (warner colors indicate higher gradient persistence)

Approach

- Setup on-premise Cloud environment.
- Select dataset and algorithm for anomaly detection.
- Design and develop OceanXtremes backend.
- Validate OceanXtremes using selected datasets and algorithms.
- Design, develop and integrate web portal to backend system.
- Integrate datacasting and visualization capability.
- Expand the number of datasets and algorithms supported within OceanXtremes
- Conduct end-to-end demonstration.

Key Milestones

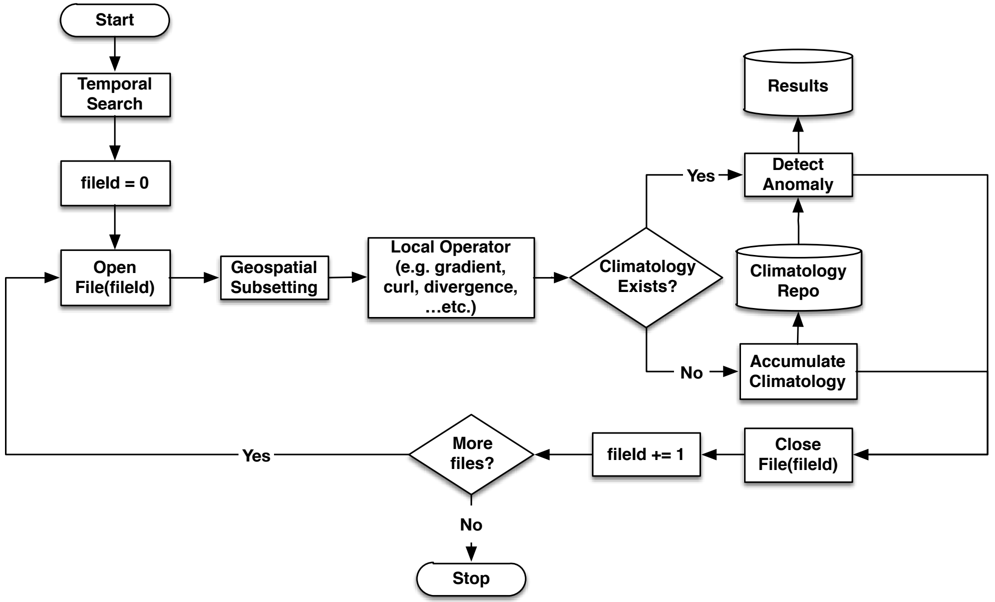
- Complete backend system design 12/15
- Complete testing of backend system 05/16
- Complete web portal design 08/16
- Integrate web portal and backend system 11/16
- Integrate datacasting and visualization capability 02/17
- Collect benchmarking data 04/17
- Conduct end-to-end demonstration 05/17

Co-Is: E. Armstrong, G. Chang, T. Chin, B. Wilson, JPL

TRL_{in} = 2 **$TRL_{current}$ = 3**

Motivation

- Anomaly detection is a process of identifying items, events or observations outside the “norm” or expected patterns
- Current and future oceanographic missions and our research communities present us with challenges to rapidly identify features and anomalies in increasingly complex and voluminous observations
- Typically this is a two-stage procedure
  1. Determine a long-term/periodic mean (“climatology”)
  2. Deviations from the mean are searched. Step 1 could be omitted in cases where a climatology data set already exists.



Earth Science Technology Office

OceanXtremes Architecture

Xtremes Ingester

Real-time ingestion system

Xtremes Climatology

Batch-oriented climatology computation service

Xtremes Processor

Horizontal-scale system for anomaly computation and detection

Xtremes Analyzer

Webservice to access data and anomalies

Xtremes Visualizer

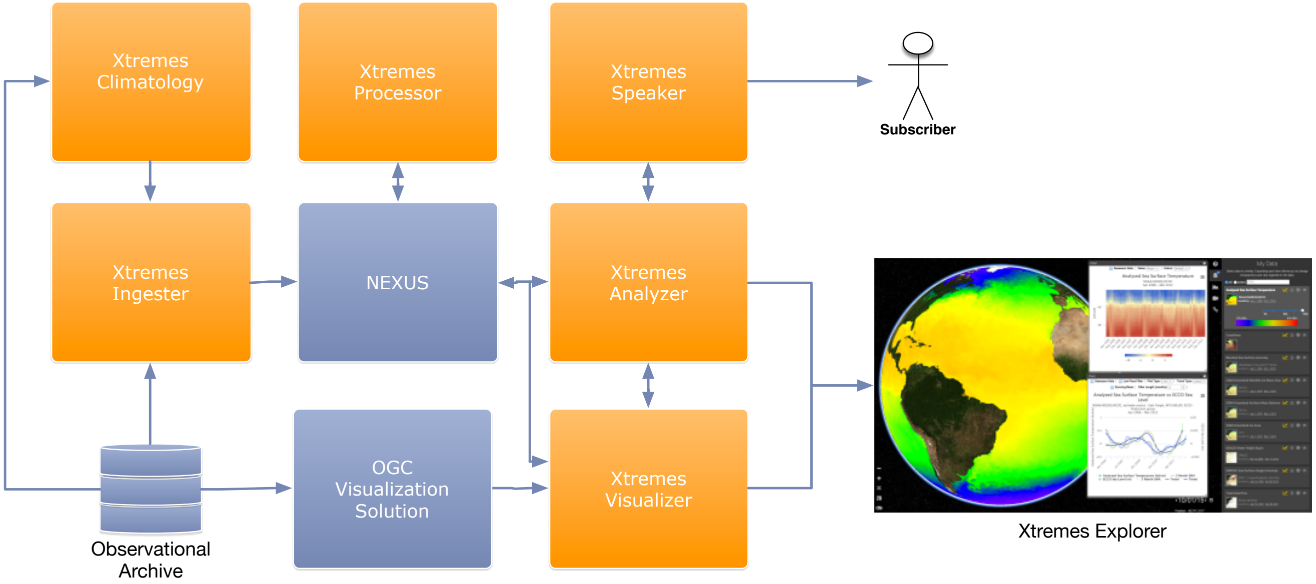
Web service for data visualization

Xtremes Speaker

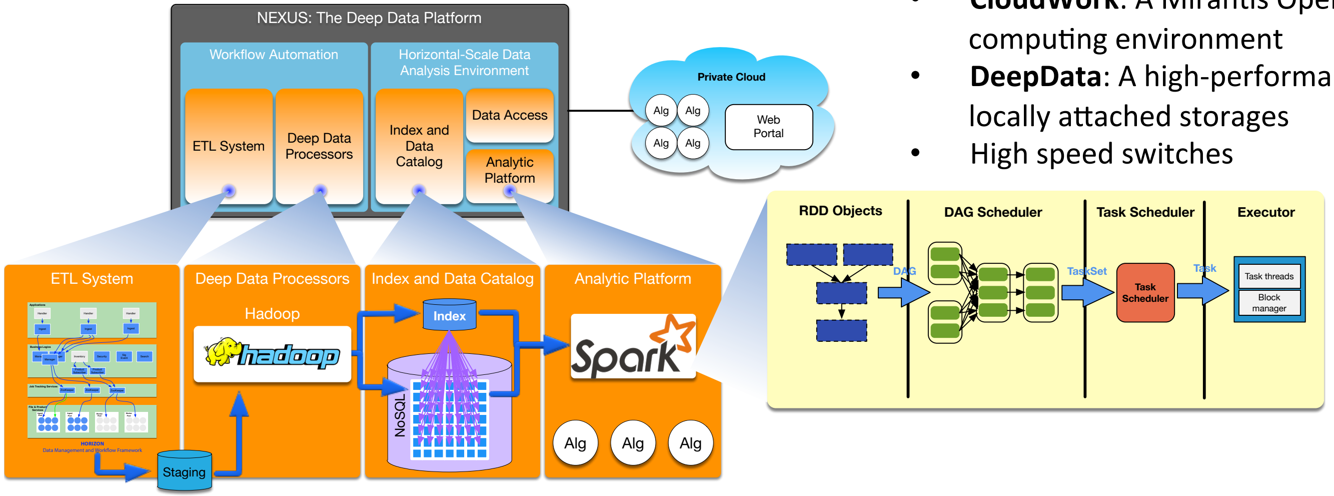
Feed generation and management system

Xtremes Explorer

Web-based data visualization and analysis



Deep Data Computing Environment (DDCE)



DDCE is OceanXtremes' development environment. It consists of

- **CloudWork**: A Mirantis OpenStack private cloud computing environment
- **DeepData**: A high-performance data cluster with locally attached storages
- High speed switches

openstack
CLOUD SOFTWARE

NEXUS Deep Data Analytics: One-Minute Summary

NEXUS is an emerging technology developed at JPL

- A Cloud-based/Cluster-based data platform that performs scalable handling of observational parameters analysis designed to scale horizontally by
  - Leveraging high-performance indexed, temporal, and geospatial search solution
  - Breaks data products into small chunks and stores them in a Cloud-based data store

Data Volumes Exploding

- NISAR & SWOT missions coming
- File I/O is slow

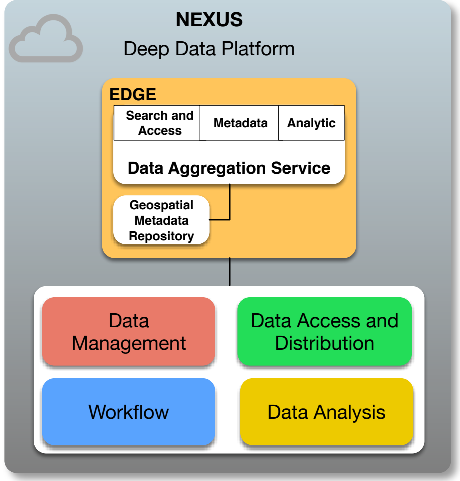
Scalable Store & Compute is Available

- NoSQL cluster databases
- Parallel compute, in-memory map-reduce
- Bring Compute to Highly-Accessible Data

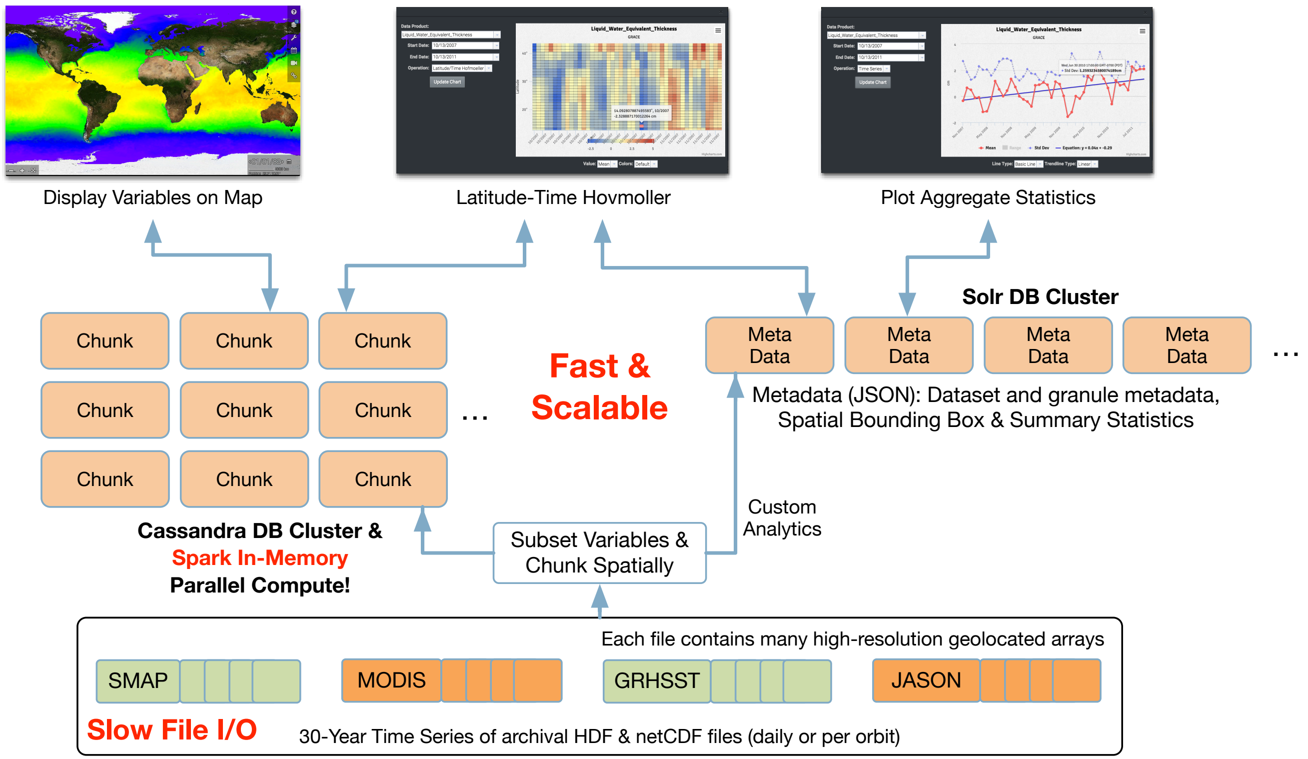
Pre-Chunk and Summarize Key Variables

- Easy statistics instantly (milliseconds)
- Harder statistics on-demand (in seconds)
- Visualize original data (layers) on a map quickly

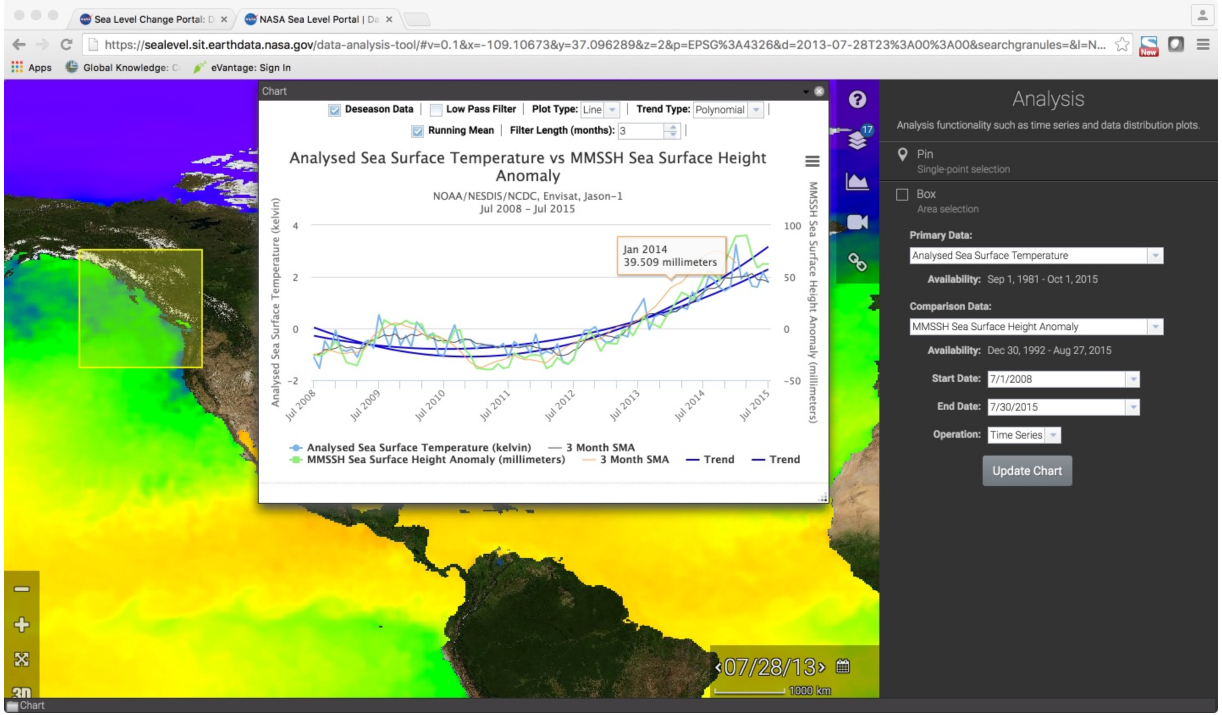
A growing collection of data analytic microservices



Analytics & Summarization of Stack



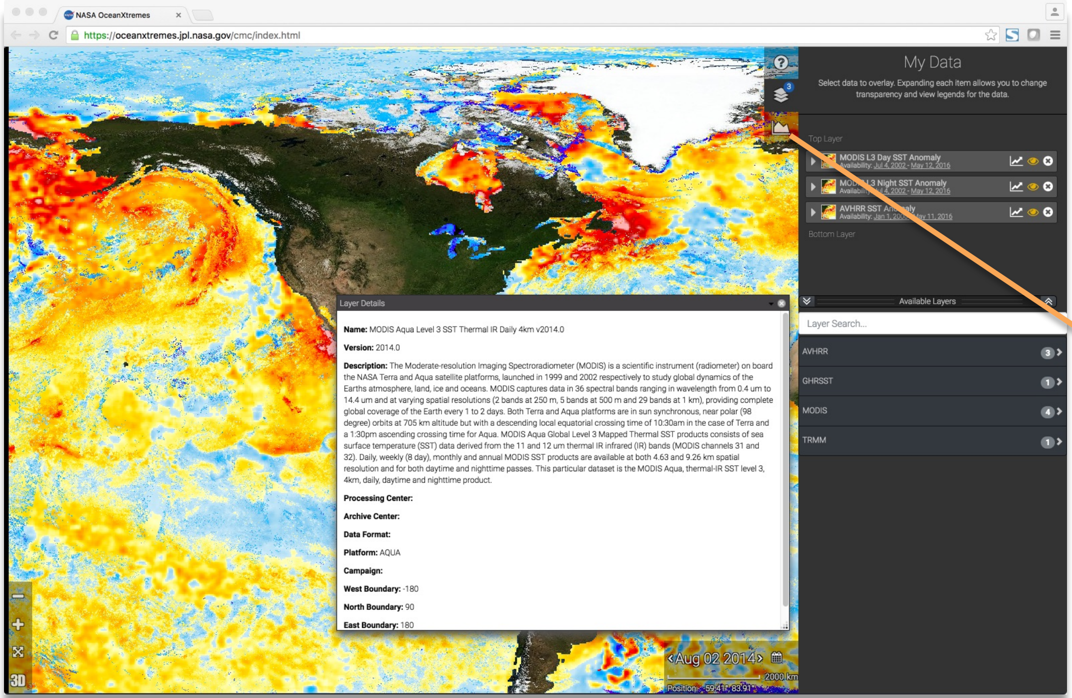
Enable Ocean Science



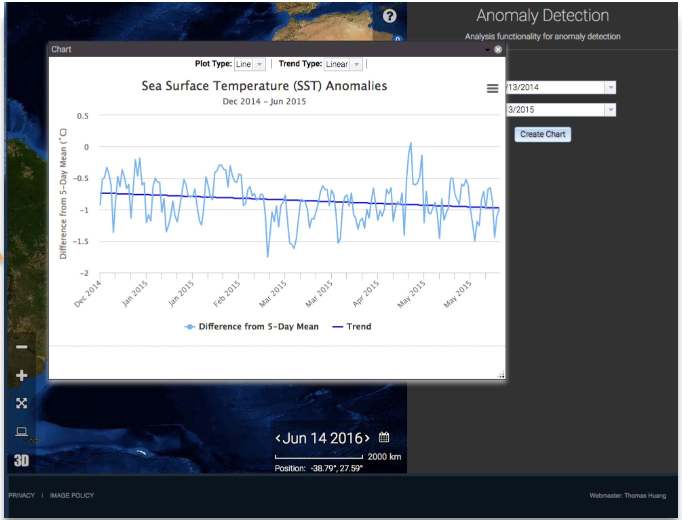
"The Blob is a result of a high pressure system that has parked itself in the Gulf of Alaska for the past few years that has driven the polar jet stream north into northern Canada and then it plunged rapidly out of northern Canada into the American Midwest and northeast. And so the result was hot dry winters on the west coast, and fierce winters with heavy snow pack in the Midwest." – Bill Patzert, NASA/JPL

Earth Science Technology Office

Xtremes Explorer

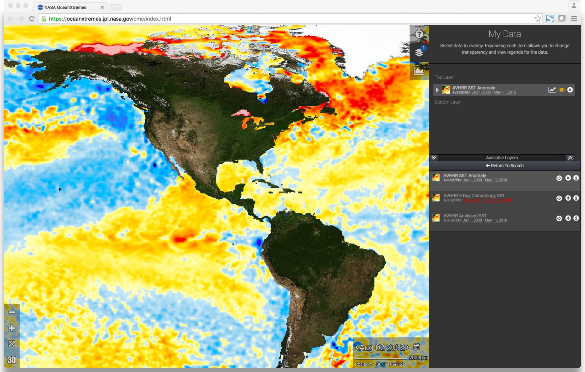


High Resolution Data Visualization for the Web

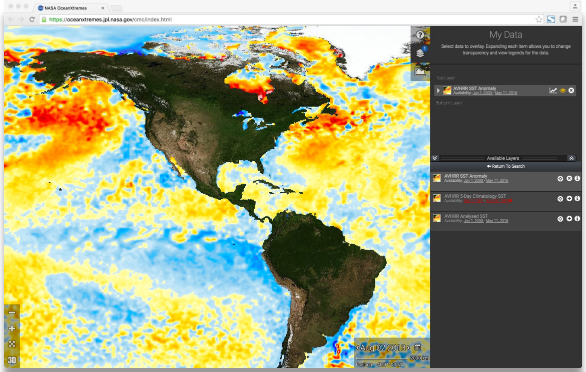


Data Analysis Workbench

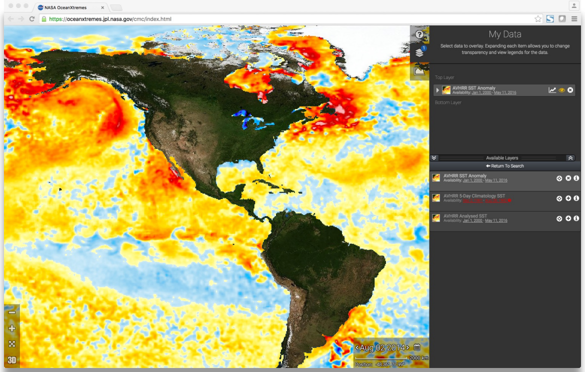
Daily Anomalies



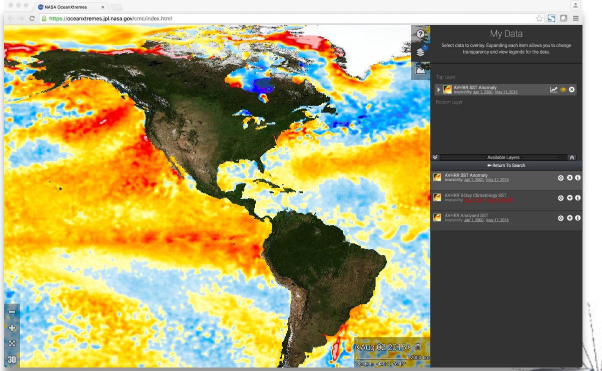
Aug 02, 2012



Aug 02, 2013



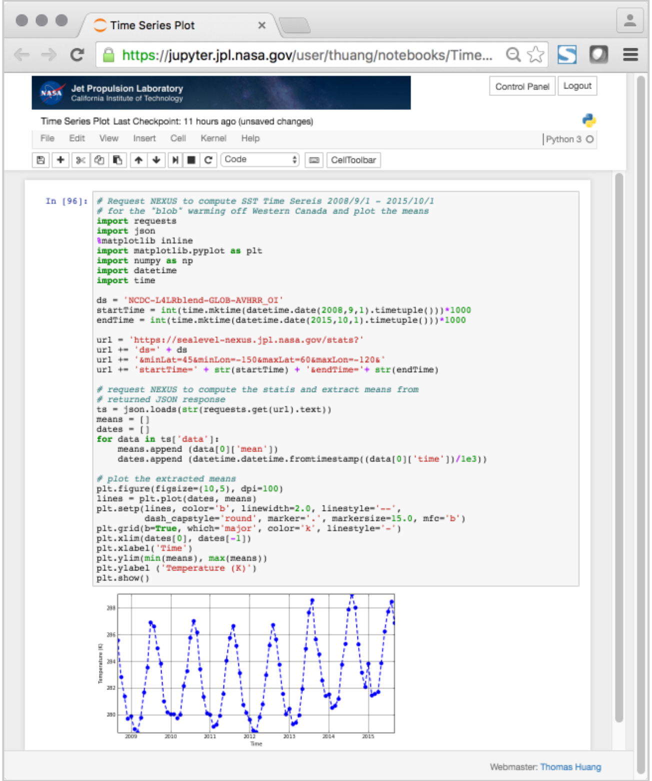
Aug 02, 2014



Aug 02, 2015

The Notebook

Interact with OceanXtremes using Jupyter Notebook



- **`/capabilities`**: list of capabilities
- **`/chunks`**: list data chunks by location, time, and datasets
- **`/correlationMap`**: Correlation Map
- **`/datainbounds`**: Matchup operation to fetch values from dataset within geographic bounds
- **`/datapoint`**: Matchup operation to fetch value at lat/lon point
- **`/dailydifferenceaverage`**: Daily difference average
- **`/latitudeTimeHofMoeller`**: Latitude Time Hovmoeller
- **`/list`**: list available datasets
- **`/longitudeLatitudeMap`**: Longitude Latitude Map
- **`/longitudeTimeHofMoeller`**: Longitude Time Hovmoeller
- **`/stats`**: Statistics (standard deviation, count, min/max, time, mean)

Data Tiling Scheme

- Pre-processing occur during ETL phase
- Breaking geospatial arrays into small geo-addressable data chunks (or partitions)
- Tile → small → in memory processing
- All spatial indexes are managed by Apache Solr

Tiling Algorithm

$$c = \textit{Number of tiles desired}$$
$$d = \textit{Number of dimensions}$$
$$L_d = \textit{Length of dimension } d$$
$$S_d = \textit{Step size for dimension } d$$
$$S_d = \left\lfloor \frac{L_d}{\sqrt[d]{c}} + \frac{1}{2} \right\rfloor$$

MUR Data in 0.01 degrees, Tiles 2.5° x 5°

$$c = 5184$$
$$d = 2$$
$$L_{latitude} = 17999$$
$$L_{longitude} = 36000$$
$$S_{latitude} = \left\lfloor \frac{17999}{\sqrt[2]{5184}} + \frac{1}{2} \right\rfloor = \left\lfloor 249.986111111 + \frac{1}{2} \right\rfloor = 250$$
$$S_{longitude} = \left\lfloor \frac{36000}{\sqrt[2]{5184}} + \frac{1}{2} \right\rfloor = \left\lfloor 500 + \frac{1}{2} \right\rfloor = 500$$

Real-time Ingestion Solution

A real-time data ingestion system

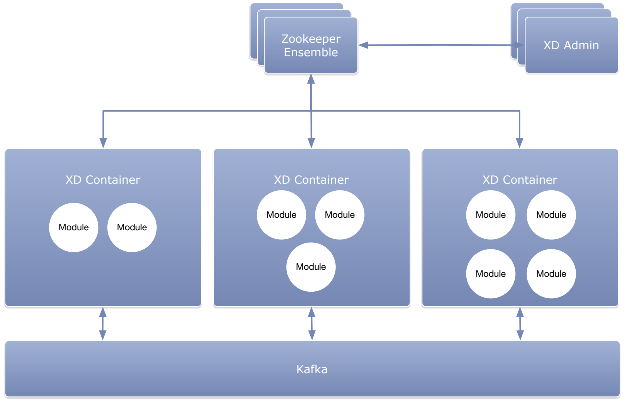
1. Data discovery
2. Metadata extraction
3. Data partition (tiles)
4. Pre-compute metrics
5. Register to NEXUS

Core components

- Admin
- Containers
- High-performance message broker
- Distributed synchronization service

Deployed under OpenStack Cloud

18 virtual instances



Real-time Data Ingestion Architecture

High-level Ingestion Workflow

Investigated Parallel Performance

Four technologies:

Multicore on single node	8 core = 8-way parallelism
PySpark on YARN scheduler	8 nodes x 4 cores on each 32-way parallelism
PySpark on Mesos scheduler	
DPark on Mesos (fastest)	

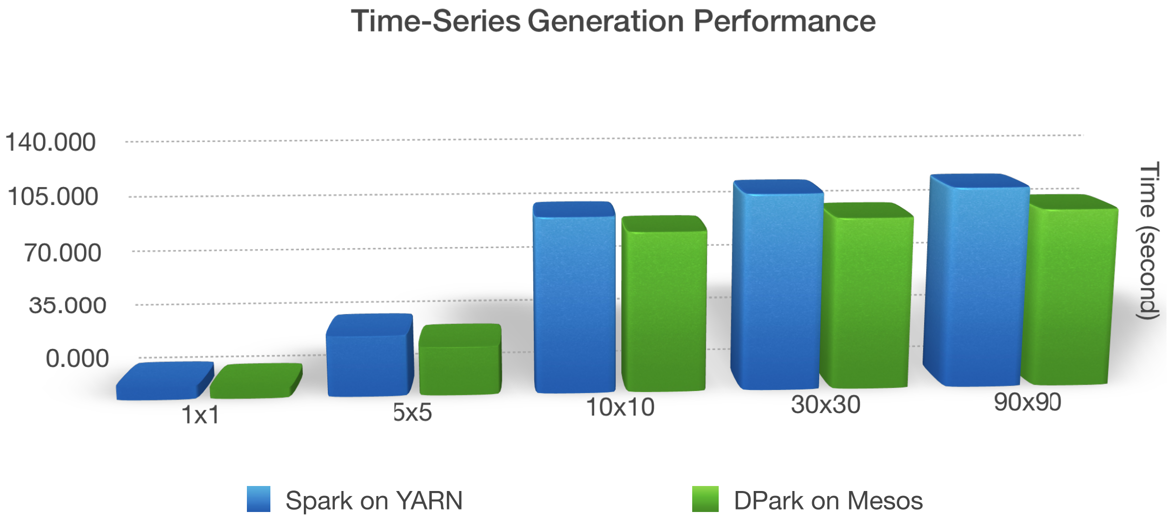
Multiple runs over different numbers of tiles

- Query for tiles that intersect a user-chosen lat/lon rectangle and time range
- Multiple rectangles: 1, 5, 10, 30, and 90 degree lat/lon boxes

Vary number of partitions to keep cores busy (> 2-3X)

- 32, 64, 128, 256 (128 best, 256 saturates)

Performance Benchmark



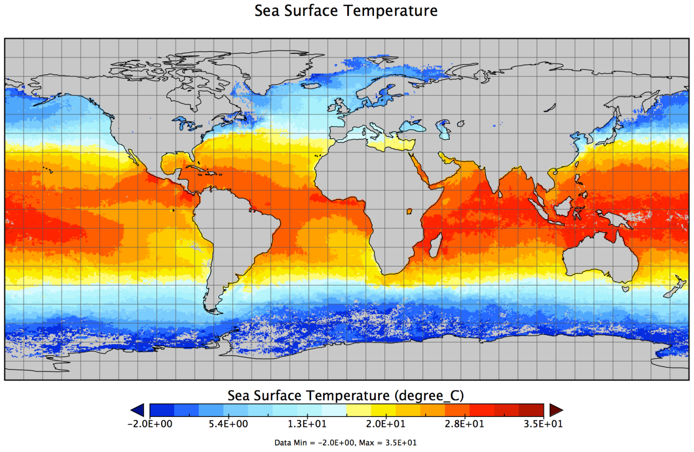
Environment	1x1	5x5	10x10	30x30	90x90
Spark on YARN	7.562	36.663	106.101	118.678	121.306
DPark on Mesos	5.638	29.353	96.799	103.839	107.826

DPark on Mesos is fastest and scales with # of tiles

- Mesos vs. YARN: shorter startup time, faster task scheduling
- DPark: no data movement between python runtime and JVM
- As # of tiles grows, 7 nodes x 4 cores all kept busy.

Climatology Algorithm: Gaussian Interpolation

- Armstrong, E. and J. Vaquez-Cuervo, A New Global Satellite-Based Sea Surface Temperature Climatology, *Geophysical Research Letters* Volume 28, No. 22, Pages 4199-4202, November 15, 2001
- A time/space Gaussian interpolation to generate global sea surface temperature climatology
- The Fortran-based implemented was ported to execute on the Deep Data Computing Cluster
- Python wrapper is being implemented to simplify integration into Xtremes Climatology
- Allow users to rapidly create regional and custom period climatologies for SST, wind etc.



SST - 4km
Climatology 2002 - 2016

GEOPHYSICAL RESEARCH LETTERS, VOL. 28, NO. 22, PAGES 4199-4202, NOVEMBER 15, 2001

A New Global Satellite-Based Sea Surface Temperature Climatology

Edward M. Armstrong and Jorge Vazquez-Cuervo

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

Abstract. A new approach to the generation of a global sea surface temperature (SST) climatology from satellite data is presented. This work is an extension of *Casey and Cornillon* [1999] who demonstrated the overall superiority of constructing a global climatology using exclusively advanced very high resolution radiometer (AVHRR) Pathfinder satellite SST data vs. blended in situ/satellite data. In this implementation, a global pentad (five day) climatology was derived from daily 9 km AVHRR Pathfinder SST data through Gaussian interpolation and averaging. Performance of this climatology with respect to the Casey 9 km pentad satellite and Reynolds 1° monthly climatologies was then investigated by examining the standard deviation of the anomaly data set constructed by subtracting climatological SST observations from co-located long-term in situ SST observations. In all areas examined this new climatology, hereafter referred to as the JPL pentad climatology, demonstrated modest improvements over the other climatologies.

Introduction

The need for accurate SST climatology is well-known and important to programs such as the Intergovernmental Panel on Climate Change (IPCC) whose investigations attempt to identify and predict anthropogenic-induced global warming from observed and computer modeled data. Recently, *Casey and Cornillon* [1999] created an entirely satellite-based climatology that typically "outperformed" the Reynolds 1°, Global Sea-Ice and SST (GISST) 1°, 1994 World Ocean Atlas (WOA94) 1° and GOSTA 5° SST climatologies that are derived from blended in situ/satellite or in situ only data. The performance evaluation methods used long-term in situ observations from the 1994 World Ocean Atlas (WOA94) and Comprehensive Ocean- Atmosphere Data Set (COADS) to construct temperature anomaly data sets formed by subtracting the climatological SST from co-located in situ SST (either WOA94 or COADS). A performance criterion was then based on the standard deviations of the anomaly temperature data sets, with the lowest standard deviation (σ) indicating the climatology best able to represent SST variability and therefore most suitable for detecting global temperature trends and reducing climatic noise. The Casey climatology was generated from daily 9 km Pathfinder AVHRR daytime and nighttime imagery from 1985-1997 (13 year baseline) through a pixel-by-pixel averaging approach of the entire time series after applying a "cloud erosion" filter.

In this study, a different approach to the nearly identical Pathfinder AVHRR data set used to derive the Casey climatology was taken (some of the Pathfinder v4.0 and v4.1 interim algorithm data used in the Casey climatology have been reprocessed with the v4.1 algorithm). No cloud erosion filtering was performed and daily files were Gaussian interpolated/averaged to climatological pentad periods. The climatology generated was tested against the Casey and Reynolds climatologies both globally, by 10° latitude bands, and for smaller high variability regions using the criterion of the minimization of anomaly SST standard deviation.

Methods and Data

The JPL pentad climatology was derived from 9 km Pathfinder SST satellite data [*Kilpatrick et al.*, 2001] from 1985-1999 using only high quality ("best pixel") SST. These data are equivalent to Pathfinder "all pixel" data with cloud flag values of 4 or higher (cloud flags range from 0-7). For each year, Gaussian interpolation was applied to spatially and temporally interpolate daily day and night data into separate pentads on a 9 km grid. The Gaussian function was of the form:

$$e^{(-0.6931*(((x-x_o)/x_h)^2+((y-y_o)/y_h)^2+((t-t_o)/t_h)^2))} \qquad (1)$$

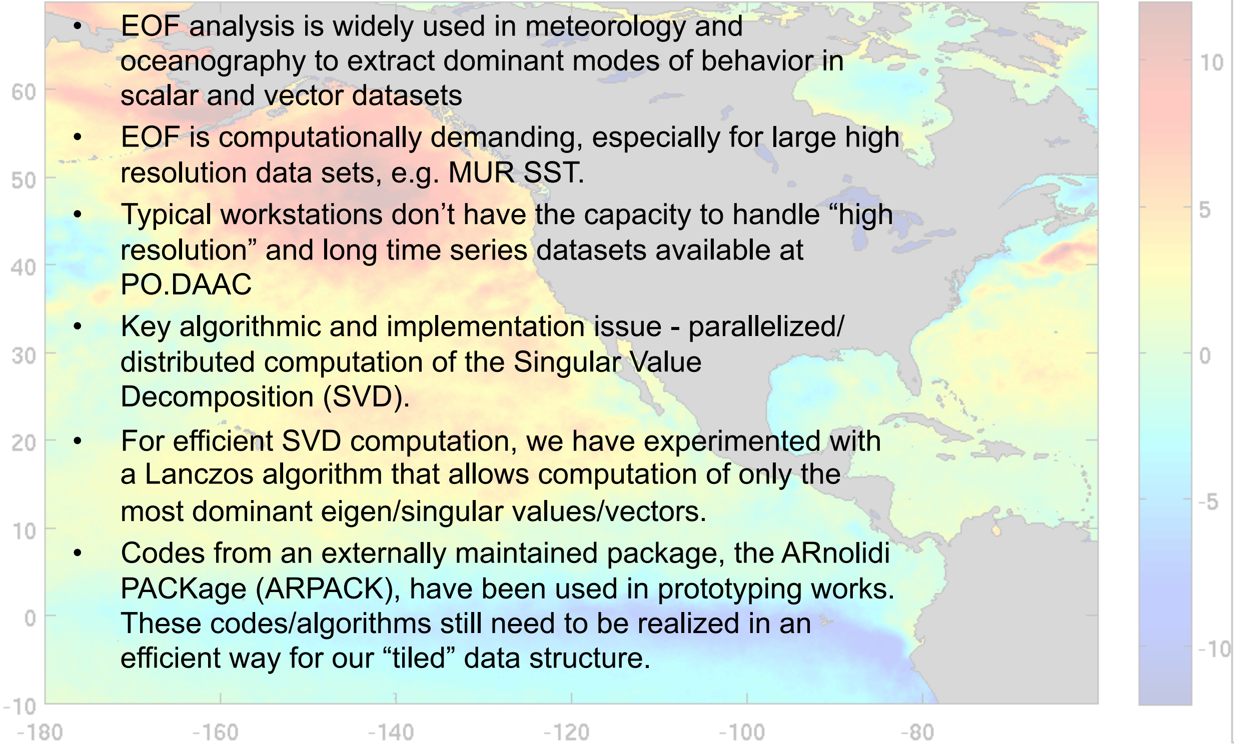
Here x and y refer to spatial points (degrees longitude and latitude, respectively), while $x_\circ$ and $y_\circ$ represent the spatial grid centers to interpolate to. t refers to the temporal period (day) and $t_\circ$ refers to the pentad to interpolate to. The x_h, y_h, and t_h parameters are the e-folding scales of the Gaussian function (i.e., the "half width" of the distribution). The spatial values, x_h and y_h, were taken as $1/e$ of the autocorrelation distributions determined from analyses of the high variability Gulf Stream and eastern equatorial Pacific regions (not shown). In both regions this was about two 9 km pixels or about 0.17° (at the equator) with the final value chosen to be 0.15°. The temporal scale, t_h, was conservatively chosen to be 1 day.

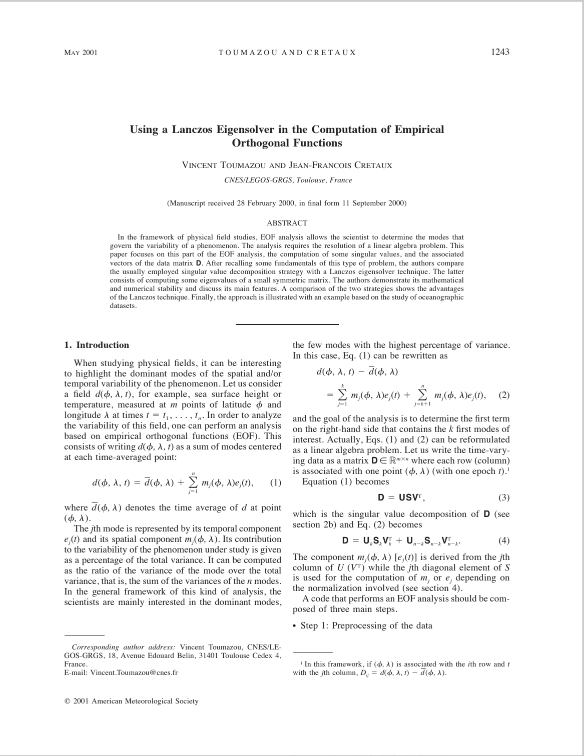
The final step averaged the individual yearly interpolated day/night pentads into the climatological pentads (e.g., all thirty pentad 70 maps from 1985-1999 were averaged to create climatological pentad 70). A noteworthy difference between the Casey and JPL climatologies was with regard to clouds. For the Casey climatology, a "cloud erosion" filter was applied to the Pathfinder images that cloud flagged SST values in the immediate (one pixel) vicinity of Pathfinder detected clouds (in "best pixel" imagery). However, we believe the Pathfinder cloud flagging in the "best pixel" imagery is already conservative since each pixel must pass a strict hierarchy of cloud contamination criteria including the stringent cloud test that is arranged as a decision tree of various tests and is unique for each AVHRR satellite [*Kilpatrick et al.*, 2001]. The JPL climatology was also created with two sep-

Paper number 2001GL013316.
0094-8276/01/2001GL013316$05.00

Algorithm: Empirical Orthogonal Function (EOF)

- EOF analysis is widely used in meteorology and oceanography to extract dominant modes of behavior in scalar and vector datasets
- EOF is computationally demanding, especially for large high resolution data sets, e.g. MUR SST.
- Typical workstations don't have the capacity to handle "high resolution" and long time series datasets available at PO.DAAC
- Key algorithmic and implementation issue - parallelized/ distributed computation of the Singular Value Decomposition (SVD).
- For efficient SVD computation, we have experimented with a Lanczos algorithm that allows computation of only the most dominant eigen/singular values/vectors.
- Codes from an externally maintained package, the ARnolidi PACKage (ARPACK), have been used in prototyping works. These codes/algorithms still need to be realized in an efficient way for our "tiled" data structure.





Using a Lanczos Eigensolver in the Computation of Empirical Orthogonal Functions

VINCENT TOUMAZOU AND JEAN-FRANCOIS CRETAUX

CNES/LEGOS-GRGS, Toulouse, France

(Manuscript received 28 February 2000, in final form 11 September 2000)

ABSTRACT

In the framework of physical field studies, EOF analysis allows the scientist to determine the modes that govern the variability of a phenomenon. The analysis requires the resolution of a linear algebra problem. This paper focuses on this part of the EOF analysis, the computation of some singular values, and the associated vectors of the data matrix **D**. After recalling some fundamentals of this type of problem, the authors compare the usually employed singular value decomposition strategy with a Lanczos eigensolver technique. The latter consists of computing some eigenvalues of a small symmetric matrix. The authors demonstrate its mathematical and numerical stability and discuss its main features. A comparison of the two strategies shows the advantages of the Lanczos technique. Finally, the approach is illustrated with an example based on the study of oceanographic datasets.

1. Introduction

When studying physical fields, it can be interesting to highlight the dominant modes of the spatial and/or temporal variability of the phenomenon. Let us consider a field $d(\phi, \lambda, t)$, for example, sea surface height or temperature, measured at m points of latitude ϕ and longitude λ at times $t = t_1, \ldots, t_n$. In order to analyze the variability of this field, one can perform an analysis based on empirical orthogonal functions (EOF). This consists of writing $d(\phi, \lambda, t)$ as a sum of modes centered at each time-averaged point:

$$d(\phi, \lambda, t) = \overline{d}(\phi, \lambda) + \sum_{j=1}^{n} m_j(\phi, \lambda)e_j(t), \quad (1)$$

where $\overline{d}(\phi, \lambda)$ denotes the time average of d at point (ϕ, λ).

The jth mode is represented by its temporal component $e_j(t)$ and its spatial component $m_j(\phi, \lambda)$. Its contribution to the variability of the phenomenon under study is given as a percentage of the total variance. It can be computed as the ratio of the variance of the mode over the total variance, that is, the sum of the variances of the n modes. In the general framework of this kind of analysis, the scientists are mainly interested in the dominant modes, the few modes with the highest percentage of variance. In this case, Eq. (1) can be rewritten as

$$d(\phi, \lambda, t) - \overline{d}(\phi, \lambda) = \sum_{j=1}^{k} m_j(\phi, \lambda)e_j(t) + \sum_{j=k+1}^{n} m_j(\phi, \lambda)e_j(t), \quad (2)$$

and the goal of the analysis is to determine the first term on the right-hand side that contains the k first modes of interest. Actually, Eqs. (1) and (2) can be reformulated as a linear algebra problem. Let us write the time-varying data as a matrix $\mathbf{D} \in \mathbb{R}^{m \times n}$ where each row (column) is associated with one point (ϕ, λ) (with one epoch t).[1]

Equation (1) becomes

$$\mathbf{D} = \mathbf{USV}^{\mathrm{T}}, \quad (3)$$

which is the singular value decomposition of **D** (see section 2b) and Eq. (2) becomes

$$\mathbf{D} = \mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^{\mathrm{T}} + \mathbf{U}_{n-k}\mathbf{S}_{n-k}\mathbf{V}_{n-k}^{\mathrm{T}}. \quad (4)$$

The component $m_j(\phi, \lambda)$ $[e_j(t)]$ is derived from the jth column of U (V^{T}) while the jth diagonal element of S is used for the computation of m_j or e_j depending on the normalization involved (see section 4).

A code that performs an EOF analysis should be composed of three main steps.

- Step 1: Preprocessing of the data

Corresponding author address: Vincent Toumazou, CNES/LEGOS-GRGS, 18, Avenue Edouard Belin, 31401 Toulouse Cedex 4, France.
E-mail: Vincent.Toumazou@cnes.fr

[1] In this framework, if (ϕ, λ) is associated with the ith row and t with the jth column, $D_{ij} = d(\phi, \lambda, t) - \overline{d}(\phi, \lambda)$.

Data Sources

Phenomenon	Dataset	Key Variables	Time Range	Data Mining Operators Needed
El Nino genesis, anomaly detection and characterization in different regions (3.4 vs 4). Coastal upwelling	CCMP L4	Wind	1987-2015	Anomaly calculation from fixed or on-the-climatology, Threshold detection. Variance characterization
	Integrated Altimeter L4	SSH	1992-2013	
	MODIS Aqua/Terra L3	SST	2000-present	
	AVHRR_OI L4	SST	1982-present	
	MUR L4	SST	2002-present	
El Nino and other teleconnections. Regional correlations	CCMP L4	Wind	1987-2015	Cross correlations. Covariabilty and EOFs.
	Integrated Altimeter L4	SSH	1992-2013	
	MODIS Aqua/Terra L3	SST	2000-present	
	AVHRR_OI L4	SST	1982-present	
	MUR L4	SST	2002-present	
	Aquarius L3	Salinity	2011-present	
	MODIS Aqua L3	Chl A	2002-present	

Earth Science Technology Office

Data Sources

Phenomenon	Dataset	Key Variables	Time Range	Data Mining Operators Needed
Upwelling. Hurricane genesis	CCMP L4	Wind	1987-2015	Divergence and curl.
	MODIS Aqua/Terra L3	SST	2000-present	
	AVHRR_OI L4	SST	1982-present	
	MUR L4	SST	2002-present	
Gradients, edges, and eddy detection	MODIS Aqua/Terra L3	SST	2000-present	Matched filter (e.g., Sobel operator). First derivatives.
	MUR L4	SST	2002-present	
	MODIS Aqua L3	Chl A	2002-present	
Trends. Basin scale variability	CCMP L4	Wind	1987-2015	Regression, Polynomial fits. Variance.
	Integrated Altimeter L4	SSH	1993-2013	
	MUR L4	SST	2002-present	

Engagements

April 2015: PO.DAAC User Working Group

June 2015/2016: Earth Science Technology Forum

July 2015: ESIP Federation Summer Meeting

October 2015: IEEE Big Data Conference

December 2015: American Geophysical Union Fall Meeting

January 2016: ESIP Federation Winter Meeting

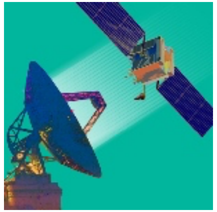
February 2016: Ocean Sciences Meeting

March 2016: Ground System Architectures Workshop

March 2016: PO.DAAC User Working Group

July 2016: ESIP Federation Summer Meeting

October 2016: International Conference on Marine Data and Information Systems – Gdanski, Poland



Near-term Plan

Data Services

- Xtremes Ingester
  - Improve tiling performance and additional tile-level stats
- Xtremes Processors
  - MapReduce framework
  - Automatic detection workflow
- Xtremes Analyzer: Search and metadata capabilities
- Xtremes Speaker: Datacasting feed management
- Docker deployment process

Science and Algorithms

- Catalog know anomalies (e.g. El Nino, hurricane, etc)
- Empirical Orthogonal Function (EOF)

Web Portal

- More visualizations
- Anomaly search
- User-defined anomaly detection

Datasets

- MODIS Terra L3 – SST and Chl A
- CCMP L4 - Wind
- Integrated Altimeter L4 - SSH
- Aquarius L3 - Salinity

Spark and Resource Management

Issue: From our benchmark comparison, we have concluded the common Spark + YARN combination, while it is faster than Hadoop, the bridge to PySpark with YARN don't yield the desired performance.

PySpark is a python wrapper on Spark, which is implemented in Scala (Java). Data is being copied between Java memory space to Python memory space. Python, because of numpy and scipy, is still the leading programming language for scientific programing

YARN got popular with Hadoop in the Cloudera distribution. It works well with Hadoop, but we discovered the scheduling overhead with YARN is less than desirable.

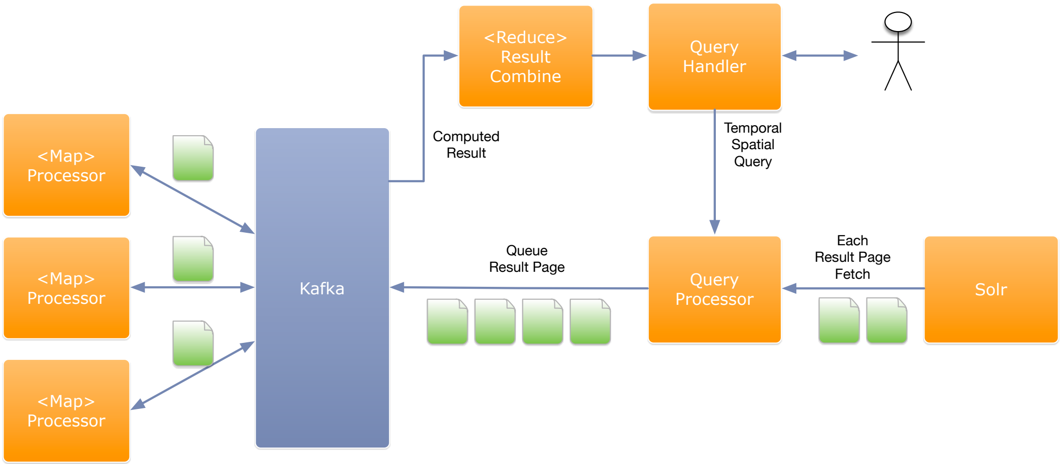
DPark is pure python implementation of Spark. Our benchmarking shows DPark + Mesos out performs PySpark + YARN by ~30%.

Near Term Plan

High-throughput Distributed Processing

Issue: Result retrieval from Solr could create huge performance bottleneck. What happen when a temporal-spatial query returns 1M matches. Current implementation fetches all 1M matches before start processing.

A new high-throughput distributed processing framework is developed to farm jobs for each Solr page fetch. It frees the system from high memory utilization and also increase parallelism, which yields faster response.



Questions?