



**DAWN:
A SIMULATION MODEL
FOR EVALUATING COSTS AND TRADEOFFS
OF BIG DATA SCIENCE ARCHITECTURES**

AGU Fall Meeting, 2014
San Francisco, CA

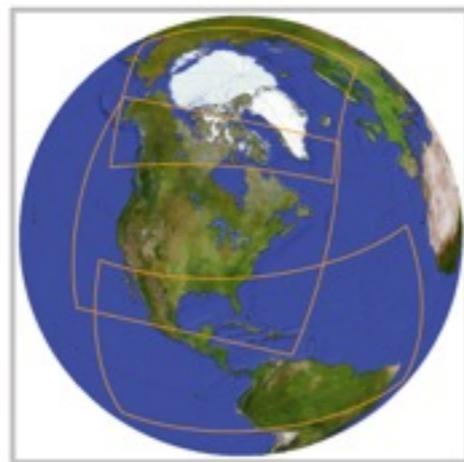
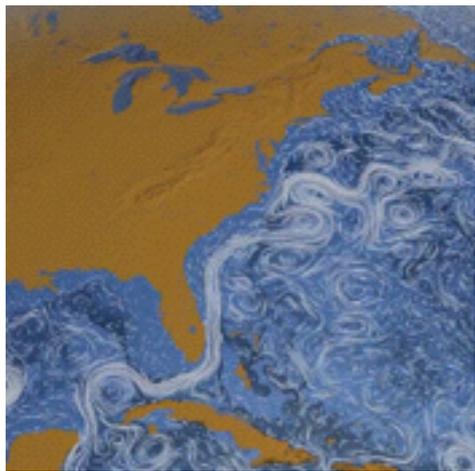
**Luca Cinquini, Lee Kyo
Daniel J Crichton, Amy J Braverman, Thomas Fuchs, Ashish Mahabal and Michael Turmon**

California Institute of Technology & NASA Jet Propulsion Laboratory

Copyright 2014 California Institute of Technology. U.S. Government sponsorship acknowledged.
JPL Unlimited Release Clearance Number: CL#.....

Introduction: Motivation

- Upcoming Big Data deluge in science: the next generation of science models and instruments will increase the size of the current data archives 10x-100x times
 - ▶ Next Climate Model Inter-Comparison Project (CMIP6): 10-20 PB global archive
 - ▶ NASA decadal Earth Observing instruments will collect tens of GB/day
 - ▶ Square Kilometre Array (SKA): 1 EB/day collected, 1.5 EB/year archived



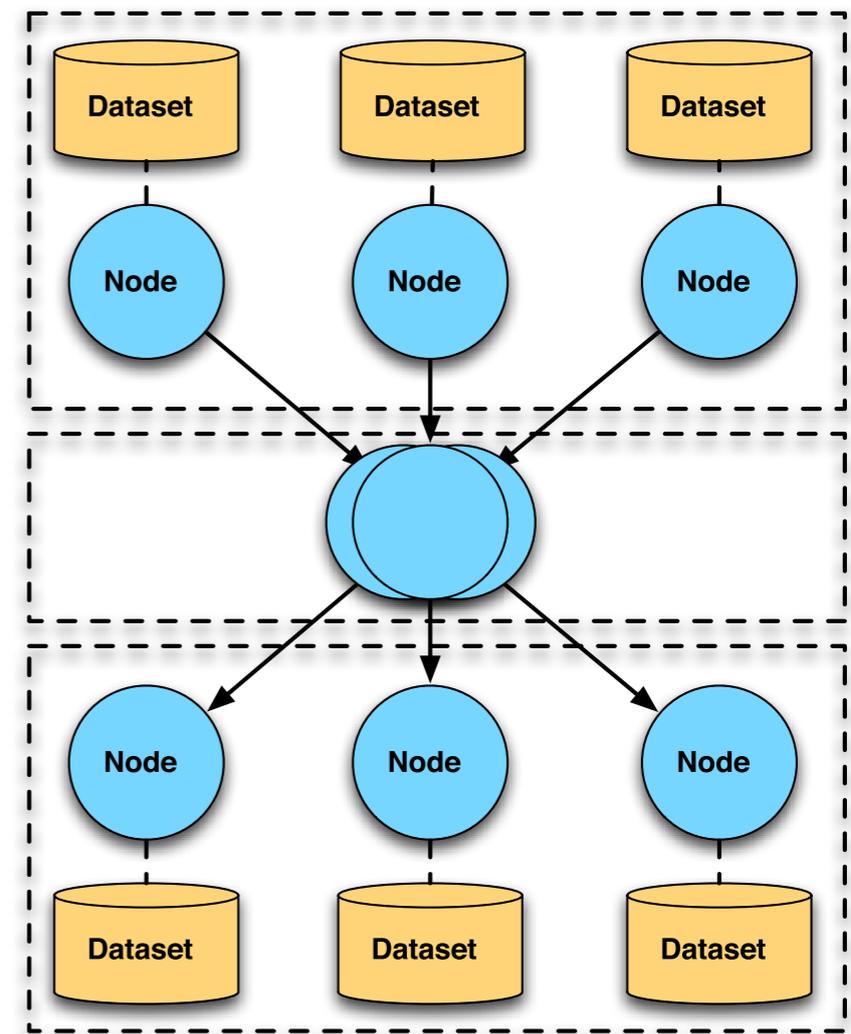
- Effective and timely processing of these enormous data streams will require the design of new system architectures, data reduction algorithms, and evaluation of tradeoffs between data reduction and results uncertainty
- At present, no software tool exists that allows simulation of complex data processing workflows to determine the computational, network and storage resources needed to prepare data for scientific analysis

DAWN: High Level Description



DAWN (Distributed Analytics, Workflows and Numerics) is a model for simulating the execution of data processing workflows on arbitrary data system architectures

- Model Inputs:
 - ▶ Formal representation of system architecture and data processing workflow
 - ▶ Numerical estimates for server capacities, network speed, data volumes, algorithms intensity
- Model Outputs: quantitative evaluation of architecture based on several metrics:
 - ▶ Overall execution time
 - ▶ Separate cumulative computation times, data transfer times
 - ▶ Results uncertainty (if available)
 - ▶ Monetary cost

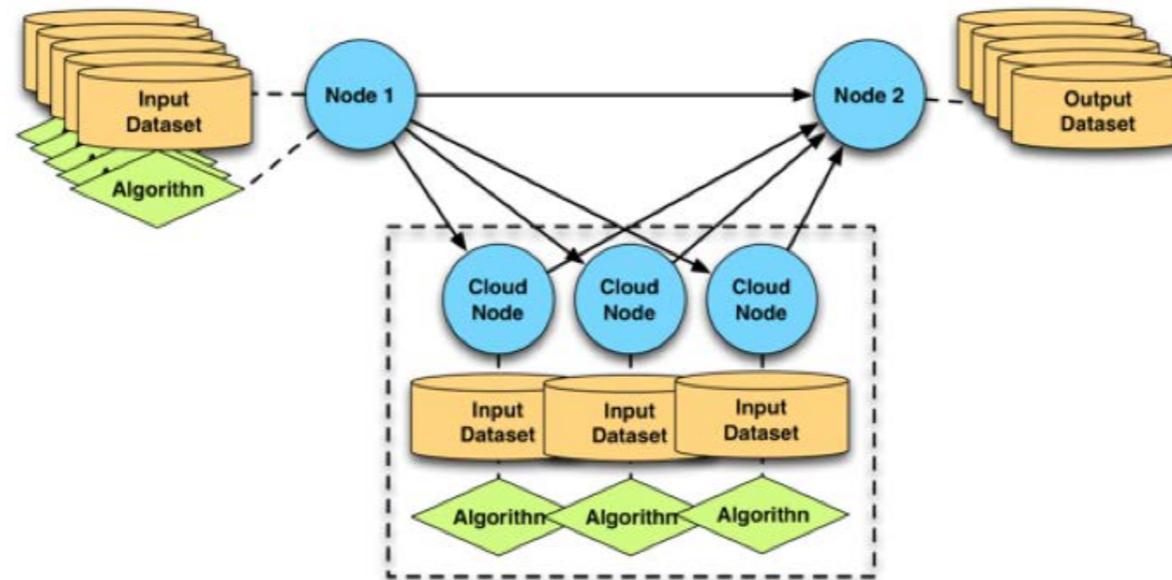


DAWN: High Level Description



- Applications:

- ▶ Select the best system architecture given a fixed set of resources
- ▶ Identify resources needed to process a given data volume/stream in a target time
- ▶ Evaluate tradeoffs between processing reduced data volumes and consequent uncertainty



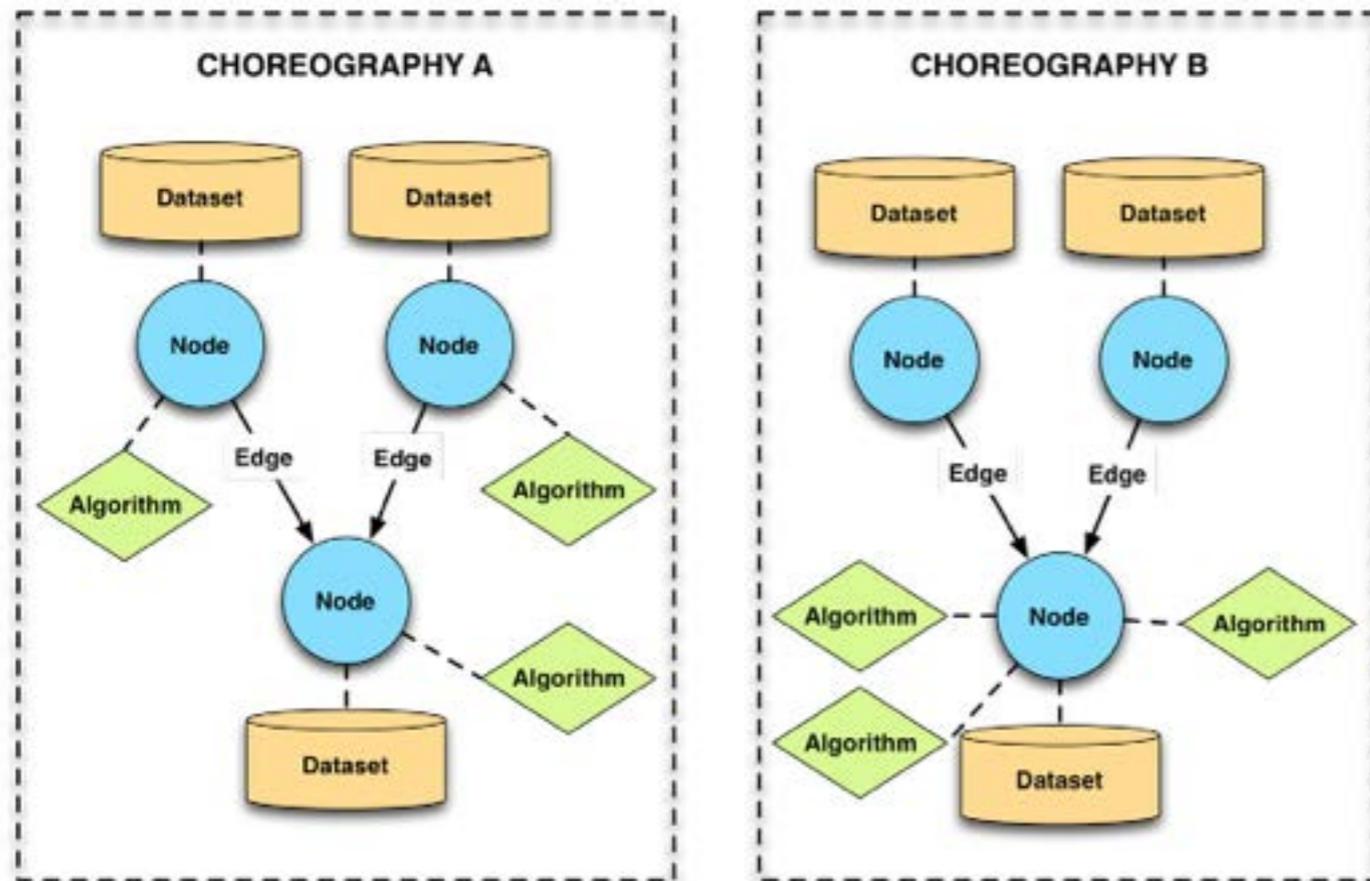
- Main Features:

- ▶ Discipline agnostic: can be applied to any data processing use case
- ▶ Fast: DAWN simulates data processing, does not actually execute it
- ▶ Extensible: users can extend the base framework to provide custom implementations for data processing, data transfer, resource management etc.
- ▶ XML-driven: use cases can be specified as formal XML documents
- ▶ Tunable: may be run multiple times by varying system parameters, or by random sampling to simulate variability of physical system components

DAWN: Main Concepts

DAWN defines the System Architecture as combination of two logical concepts:

- Topology: set of computing servers (“Nodes”) and network connecting them (“Edges”), as well as initial distribution of data or streams to be processed (“Datasets”)
- Workflow (or Choreography): specific sequence of Tasks through which data are processed over the nodes or moved along the Edges, eventually resulting in the generation of target data products



- Different workflows can be run over the same system topology to accomplish the same data processing use case, resulting in different cost and uncertainty
- DAWN overarching goal is to identify the optimal system architecture based on several metrics

DAWN: Main Concepts



DAWN objects are encoded as Python classes that can be optionally extended by the user to provide additional or modified functionality:

- Dataset: container for numerical data
 - ▶ Can be fully allocated as Numpy array
 - ▶ More often, simulated by specifying number and type (e.g. float32) of values
- Node: server processing the data
 - ▶ Defined by processing capacity (number of operations per second)
 - ▶ Processes one job at a time through a Queue
 - ▶ Stores any number of Datasets at arbitrary path locations
- Edge: network connecting two nodes
 - ▶ Defined by data transfer speed as bits per second (bps)
 - ▶ Can be configured to transmit each dataset with stochastically sampled speed
- Algorithm: “step by step procedure for calculations” (Wikipedia)
 - ▶ Ideally, defined by number of operations to process a dataset
 - ▶ Typically, complex algorithms must be benchmarked to obtain time estimate
- Task: single step within a data processing workflow
 - ▶ Two types: “computational task” or “data movement task”
 - ▶ May be executed sequentially or in parallel (if they involve different nodes)

Science Use Cases Summary



Working with domain experts, DAWN was used to analyze and optimize the system architecture of several use cases from disparate scientific disciplines:

- Astronomy:

- ▶ Classification of space objects in Catalina Real-time Transient Survey (CRTS)
- ▶ Event selection and notification by Large Synoptic Array Telescope (LSST) broker

- Medical Science:

- ▶ Data processing pipelines for identification of cancer biomarkers

- Climate Science:

- ▶ Evaluation of CMIP5/IPCC5 climate models by comparison with satellite observations

- Hydrology:

- ▶ Analysis and correlation of multiple radar observations of water resources (expected processing pipeline for NASA-ISRO Synthetic Aperture Radar (NISAR))

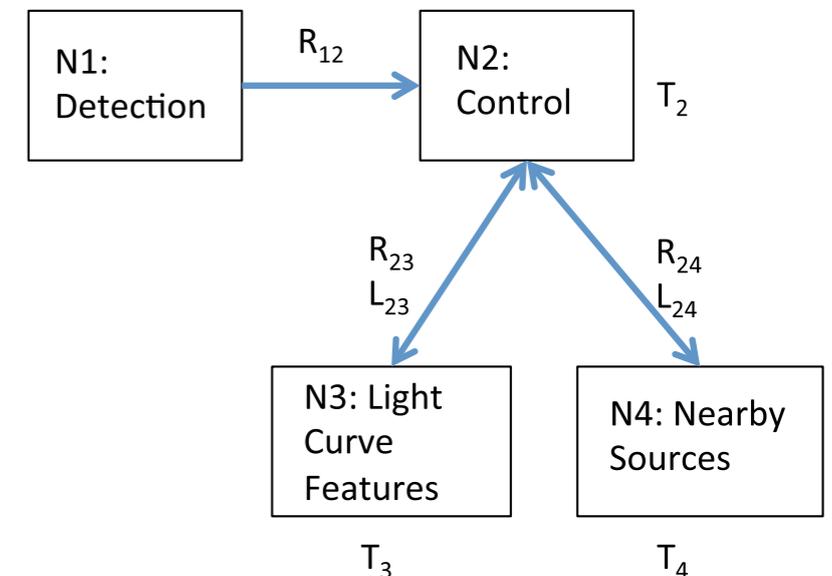
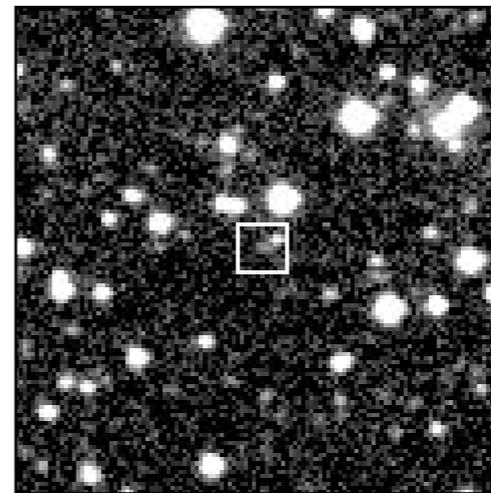
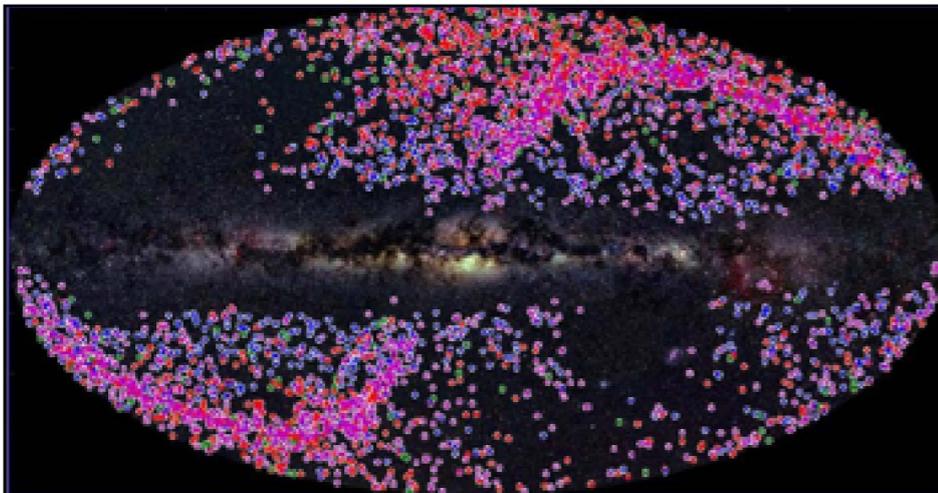
- Generic Cloud:

- ▶ Tradeoffs between advantages of distributed processing and cost of data transfer

Astronomy: Classification of Space Objects



- Description: analysis of light curve observations from astronomy survey
 - ▶ See CRTS (“Catalina Real Time Transient Survey”), LSST (“Large Synoptic Array Telescope”)
 - ▶ Light curve data are continuously transferred from Node 1 (Detection) to Node 2 (Control)
 - ▶ Light curve = time series of observation magnitude values: (T_i, V_i)
 - ▶ Each light curve is processed through two subsequent algorithms:
 - * Features Extraction: evaluates approx. 50 parameters per light curve (mean, moments, quantiles, Fourier...)
 - * Classification: uses light curve features to compare data to known objects from database



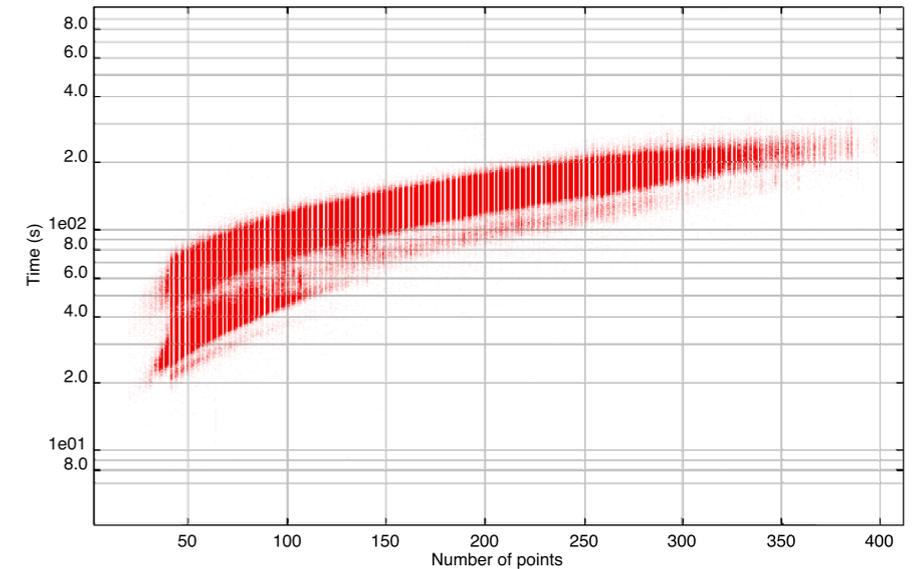
- Question: is there a gain in performance by executing the computation on another node as a function of the data streaming rate and the system parameters (network speed, execution time of algorithms, server computational capacity)

Astronomy: Classification of Space Objects

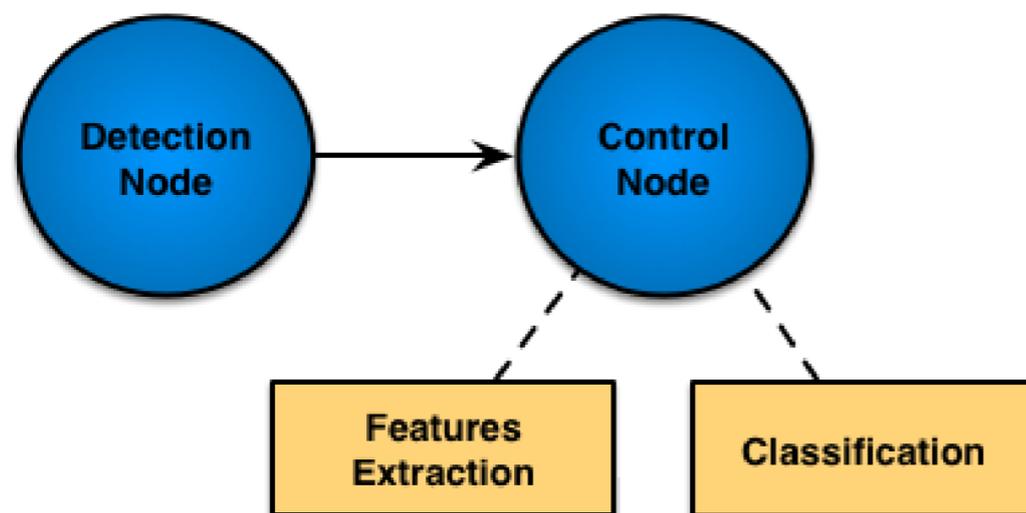


• Simulation:

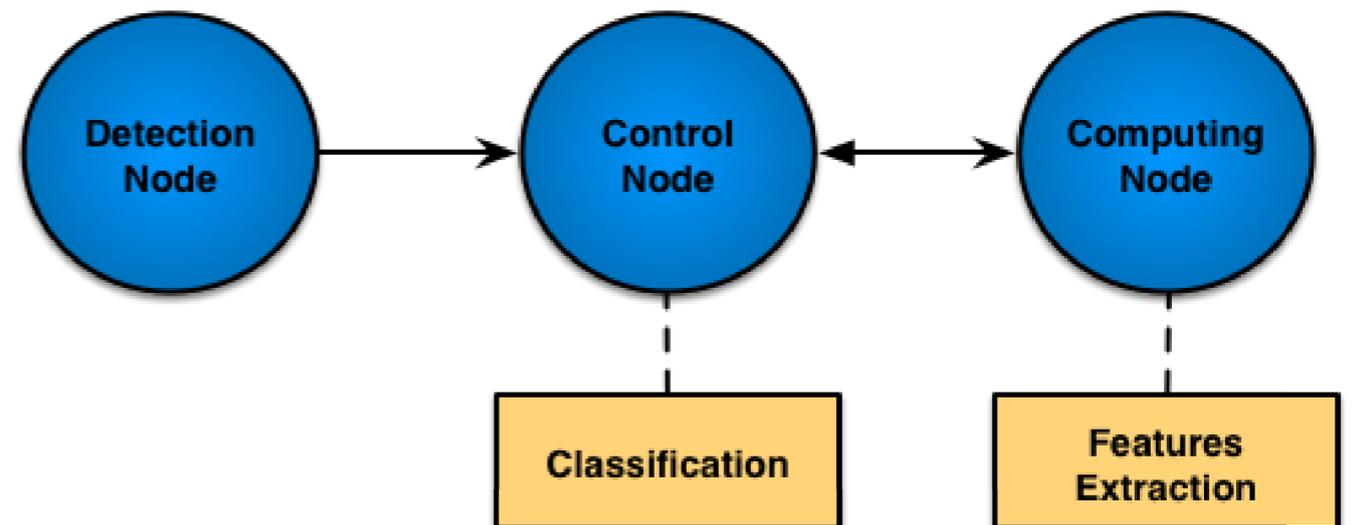
- ▶ Each light curve composed of 240 points (hourly observations for 10 days)
- ▶ Each workflow consisted of sending 10 light curves with given time offset (0-1000s)
- ▶ Features and Classification times set from reasonable estimates (192s, 240s)
- ▶ Centralized Workflow: Features Extraction and Classification algorithms both executed on same node
- ▶ Distributed Workflow: light curves transferred to Computing node to execute Features Extraction, results transferred back to Control node for Classification



Execution time for Features Extraction



Centralized Workflow



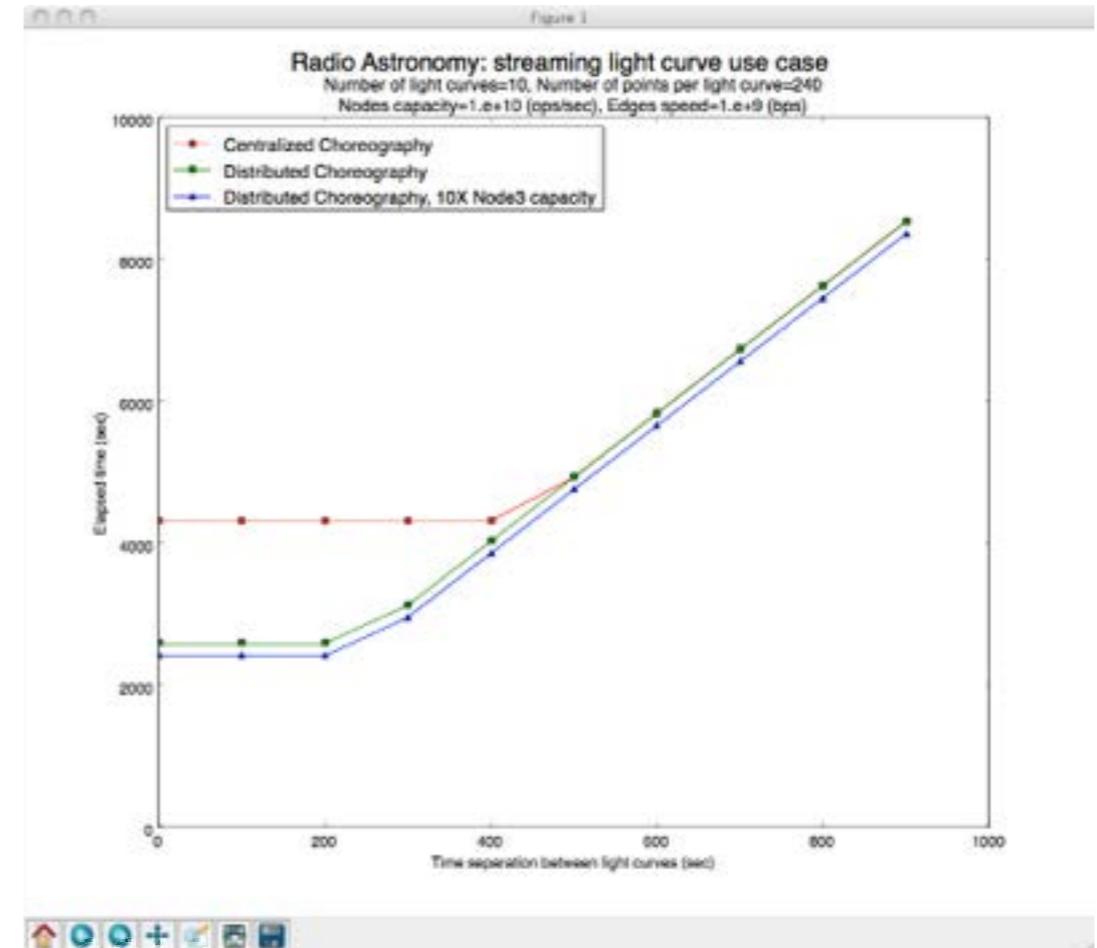
Distributed Workflow

Astronomy: Classification of Space Objects



- Results: use case is dominated by processing time and data streaming frequency, not by data transfer times
 - ▶ Distributed choreography (green) will always be faster than Centralized choreography (red) because the two computations are executed on separate machines...
 - ▶ ...but, the gains disappear at larger transmission offset (= slower frequencies) when both machines wait for the next light curve to process
 - ▶ For the Distributed choreography, increasing the computational capacity of Node 3 by 10X yields only a slight gain in overall performance, as the overall elapsed time is dominated by the Classification algorithm

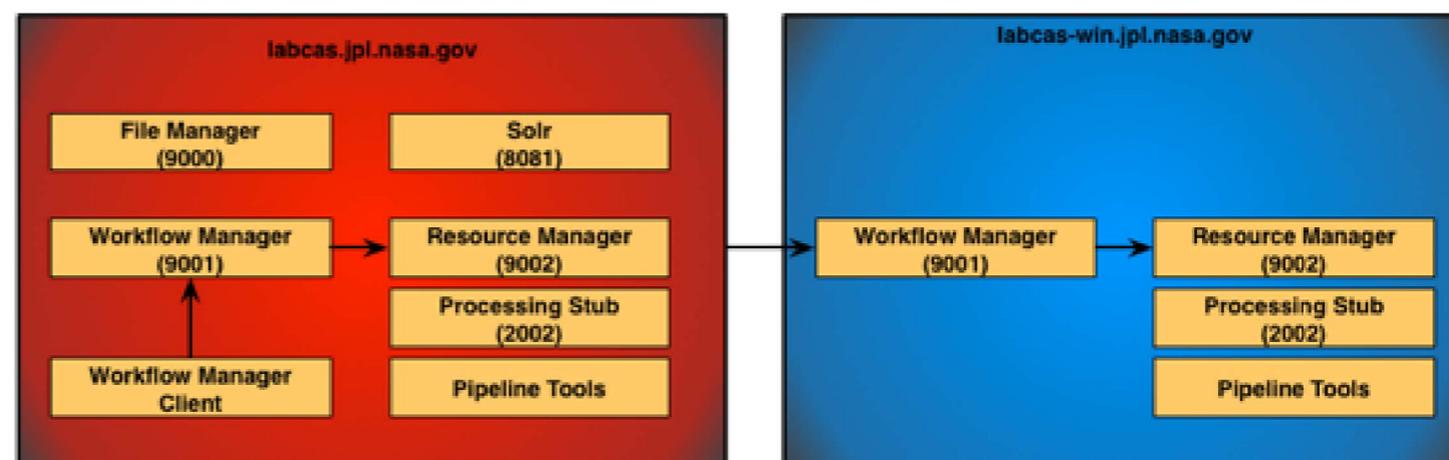
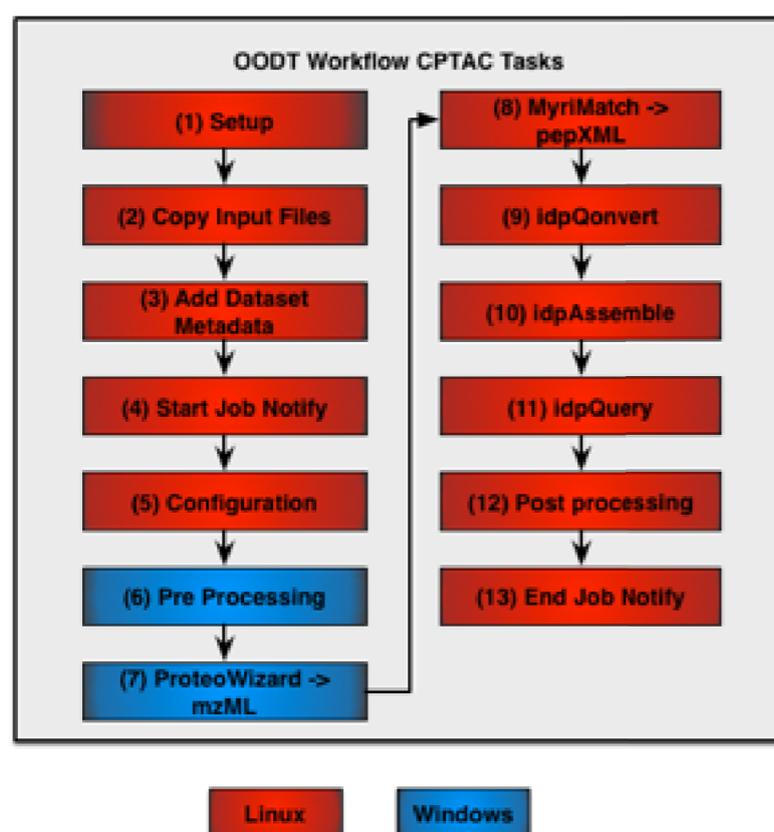
- Conclusions: besides the somewhat obvious conclusions, the model provides a quantitative estimation of the system performance as a function of several parameters, and allows to identify the data streaming rate over which the distributed workflow is more efficient.



Medical Science: Identification of Cancer Biomarkers

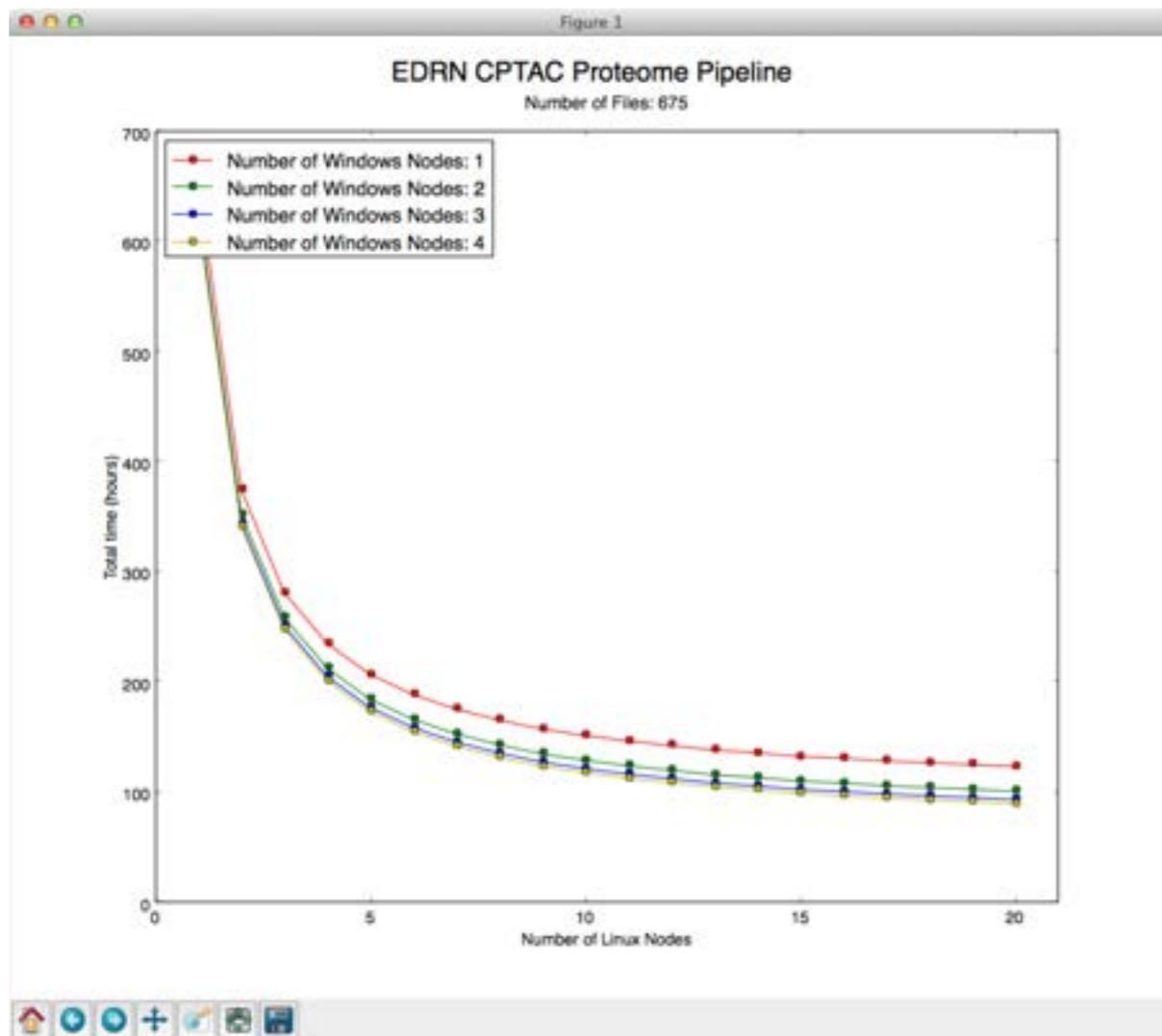


- Description: execution of intensive data processing pipelines for identification of cancer biomarkers
 - ▶ Part of Early Detection Research Network (EDRN) project funded by National Cancer Inst.
 - ▶ CPTAC (Clinical Proteomics Tumor Analysis Consortium) data processing pipeline:
 - * Sequence of 13 processing tasks running on both Linux and Windows servers,
 - includes discipline specific algorithms: ProteoWizard, MyriMatch, idpQonvert,...
 - processes 675 data files
 - * Instrumented as single workflow run by OODT (Object Oriented Data Technology)
 - * Takes 50+ days when running all tasks sequentially on 1 Linux + 1 Windows server



- Question: what is the performance gain by running the pipeline on a system of servers, and executing some of the tasks in parallel ?

- Simulation: DAWN was used to simulate execution of CPTAC pipeline on the Cloud
 - ▶ Time estimates for processing tasks obtained from benchmarking the actual algorithms
 - ▶ DAWN workflow simulated the execution of the most intensive processing task (MyriMatch) in parallel on a cluster of available processing nodes

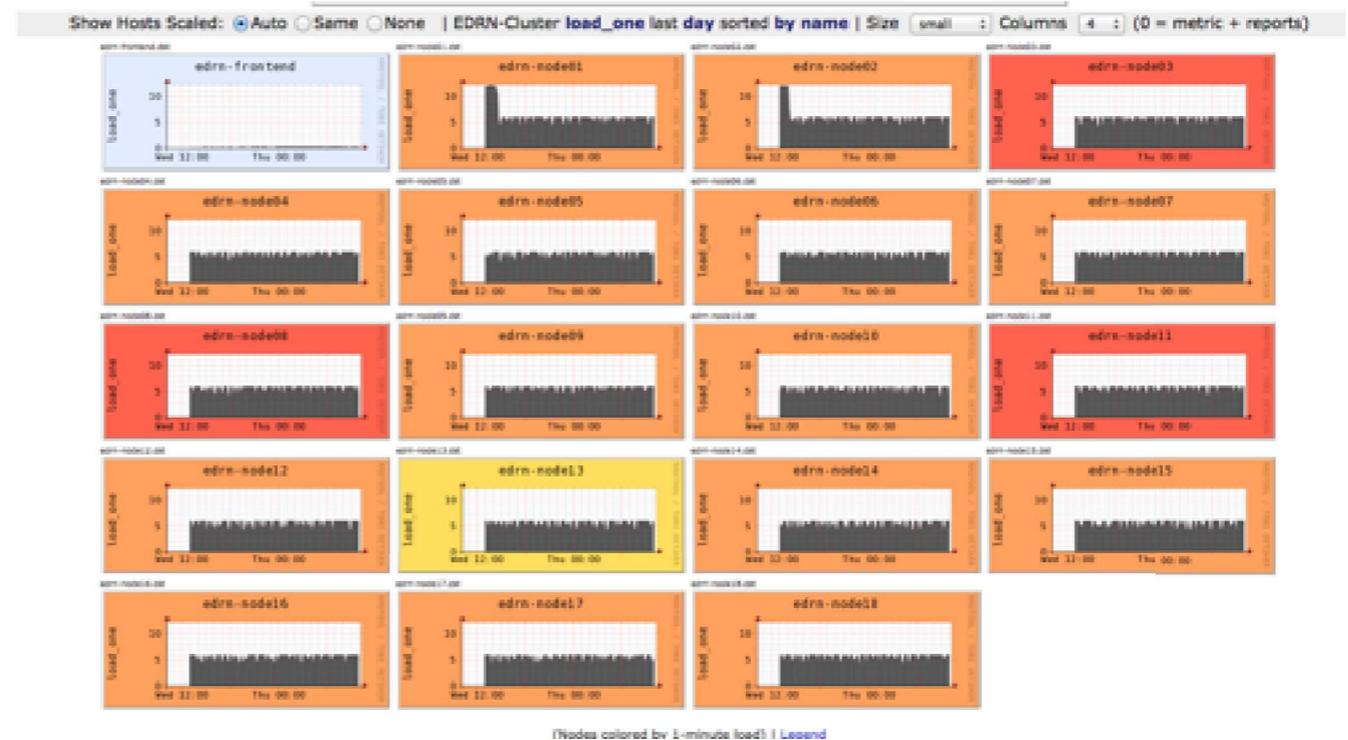
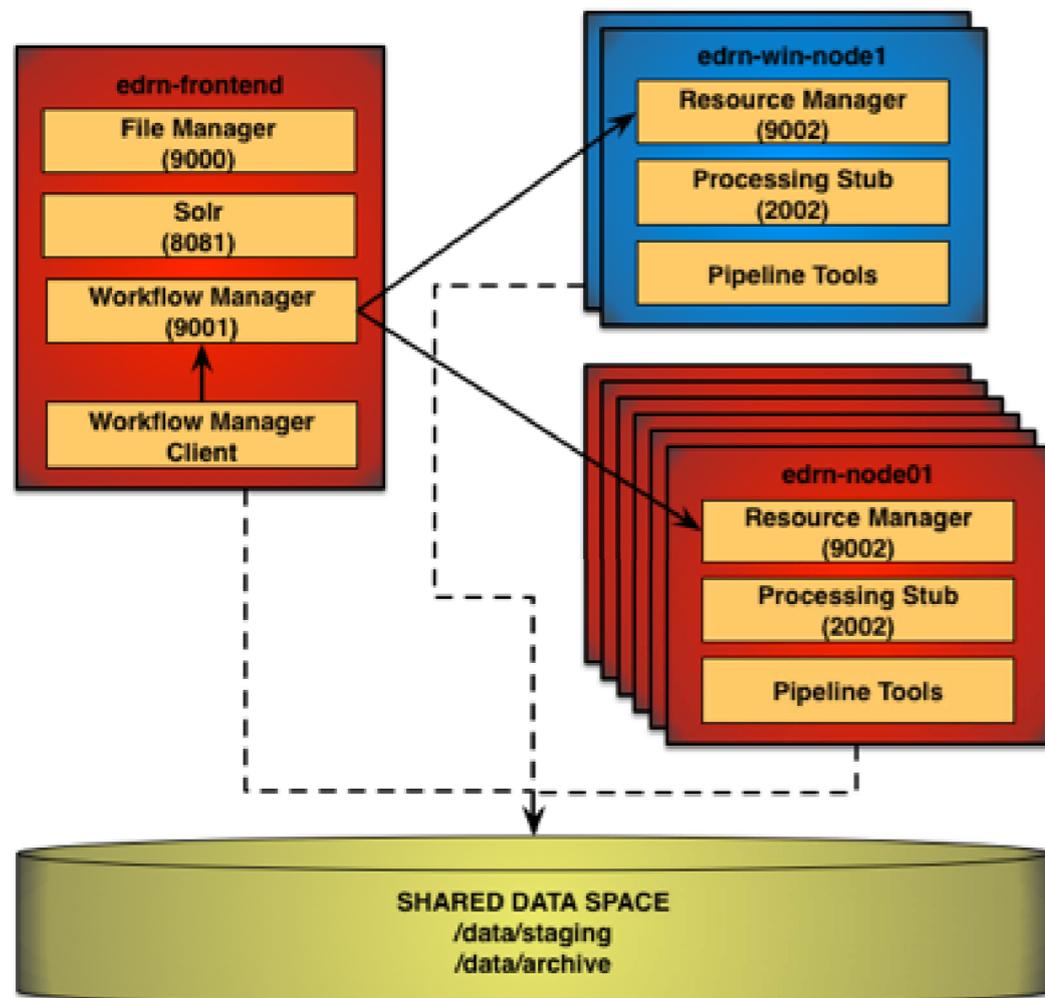


- Preliminary Results:
 - ▶ Clear reduction in overall elapsed time when MyriMatch step is executed on multiple nodes
 - ▶ Efficiency gain levels off around 15 nodes
 - ▶ Another gain is obtained by using 2 Windows nodes instead of 1
 - ▶ Full pipeline should complete in 4-5 days

Medical Science: Identification of Cancer Biomarkers



- Conclusions: following DAWN analysis, we setup an internal Cloud composed of:
 - ▶ 1 front-end Linux server (hosting common services and driving submission of workflow)
 - ▶ 2 back-end Windows servers (to execute Windows specific tasks)
 - ▶ 18 back-end Linux servers (to execute MyriMatch tasks in parallel)
 - ▶ OODT software stack + custom algorithms automatically replicated to all nodes
 - ▶ Start/stop services on all nodes simultaneously

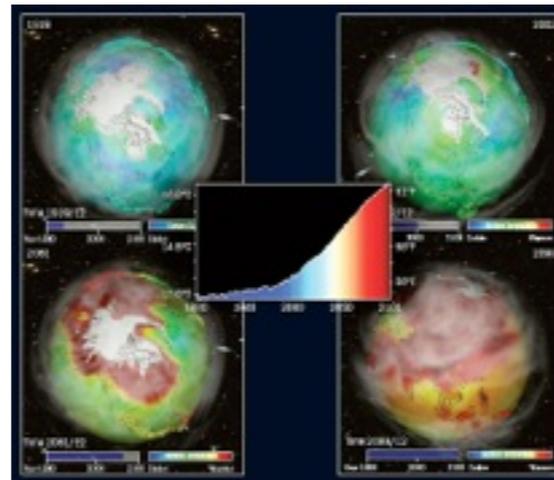
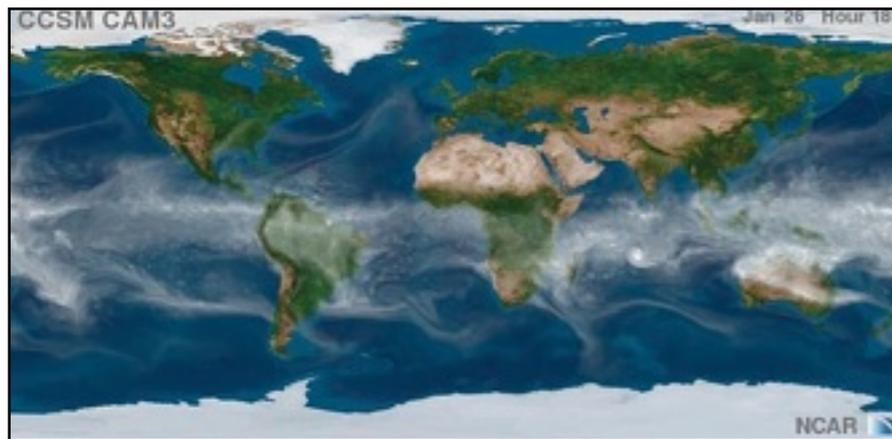


- ▶ MyriMatch sub-workflow completed in 1 day!
 - ✳ Factor of 3 improvement by using more powerful hardware (6 core servers instead of 2)
 - ✳ Factor of 18 by executing task in parallel

Climate Science: Comparison of Models & Observations



- Description: predictions for climate change are generated by running sophisticated coupled climate models on a set of agreed-upon scenarios, and combining the results
 - ▶ Process is coordinated by World Climate Research Programme (WGCM), reports authored by International Panel on Climate Change (IPCCC)
 - ▶ Models are developed and run by groups at different institutions around the world, produce large volumes of output data that are stored on distributed servers
 - *1.5 PB for current CMIP5/IPCC5, expected 10-20 PB for upcoming CMIP6/IPCC6
 - ▶ Access and analysis of model output requires large resources for moving, storing and processing the data
 - ▶ Overall reliability of climate change prediction could be improved by “scoring” models according to how well they reproduce observations for recent past

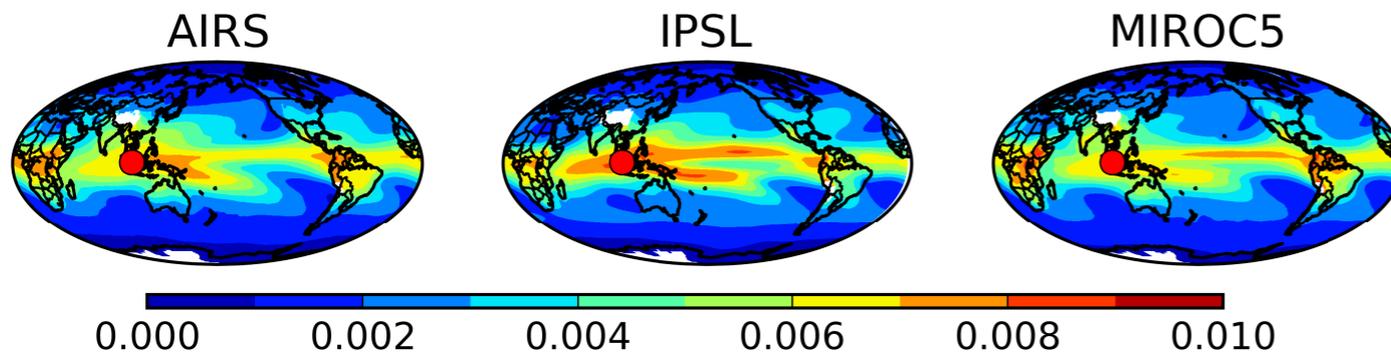


- Question: what is the most efficient architecture for processing large volumes of data from distributed servers, and the tradeoff between executing the analysis on reduced data volumes, and the resultant uncertainty of results ?

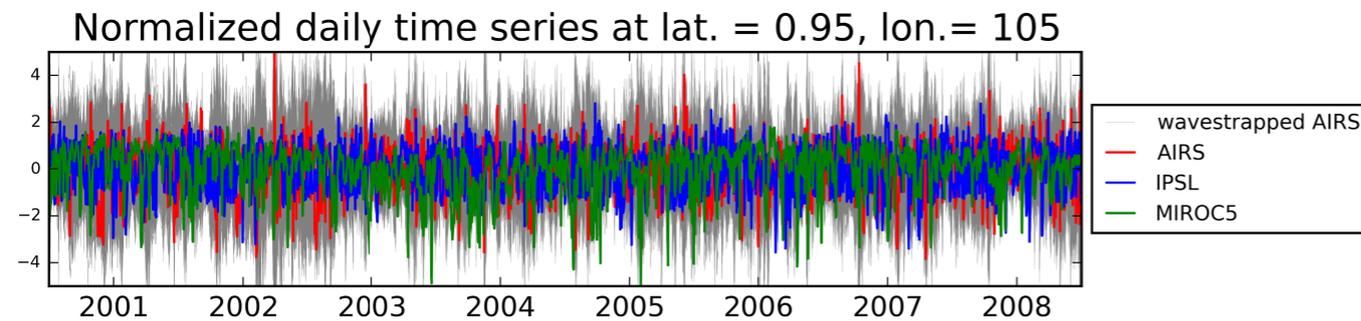
Climate Science: Specific Use Case



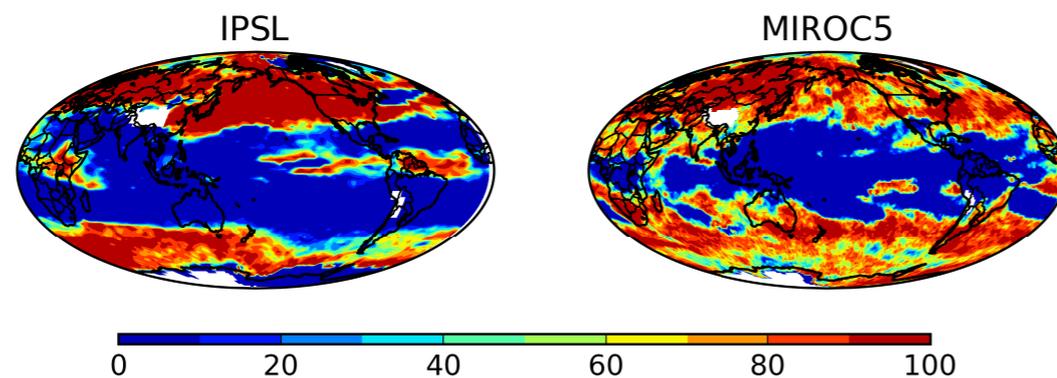
- Specific Use Case: comparison of specific humidity at 700hPa between:
 - ▶ 2 CMIP5 models: IPSL (96x96deg), MIROC5 (128x256deg)
 - ▶ AIRS satellite observations, (already) re-gridded to the models resolution
 - ▶ Daily data for 8 years (2003-2010) - (time series of 2920 points)



Observations, model datasets on original grids



Distances between observed and model time series at one grid cell



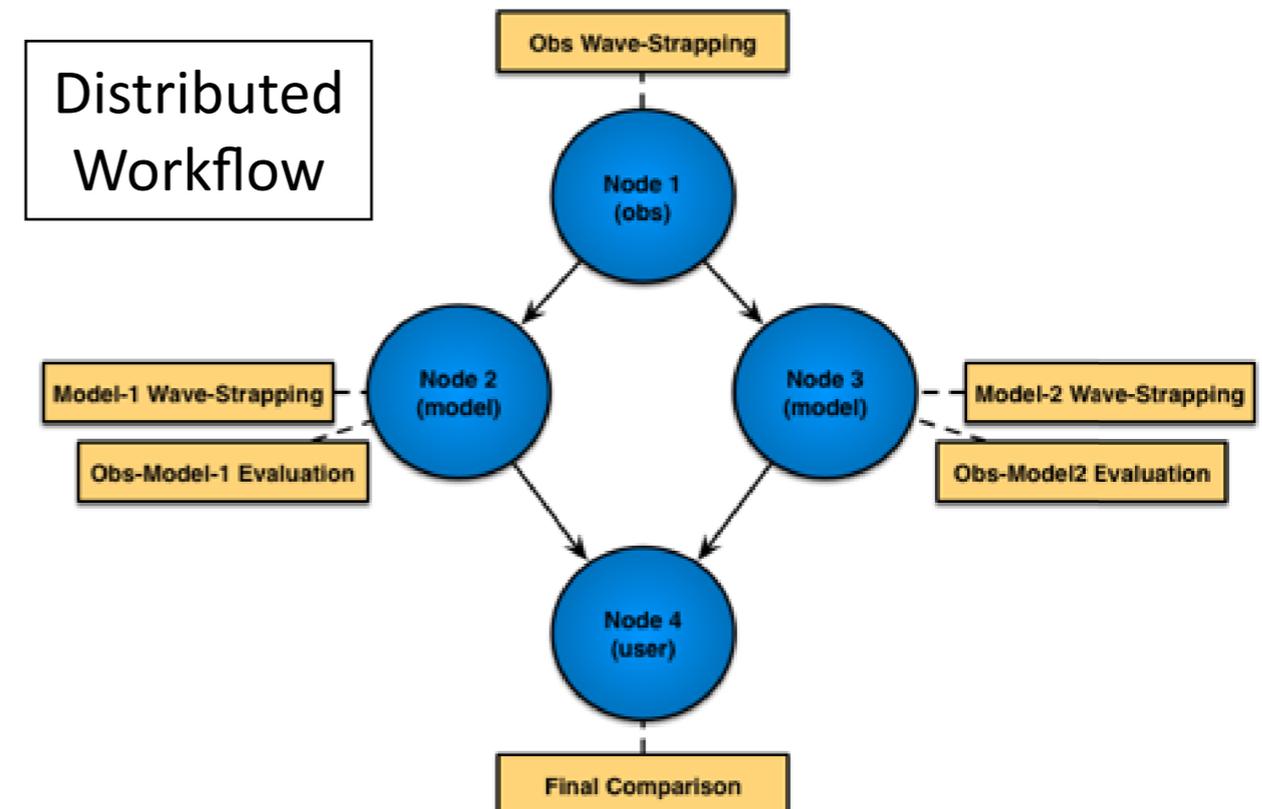
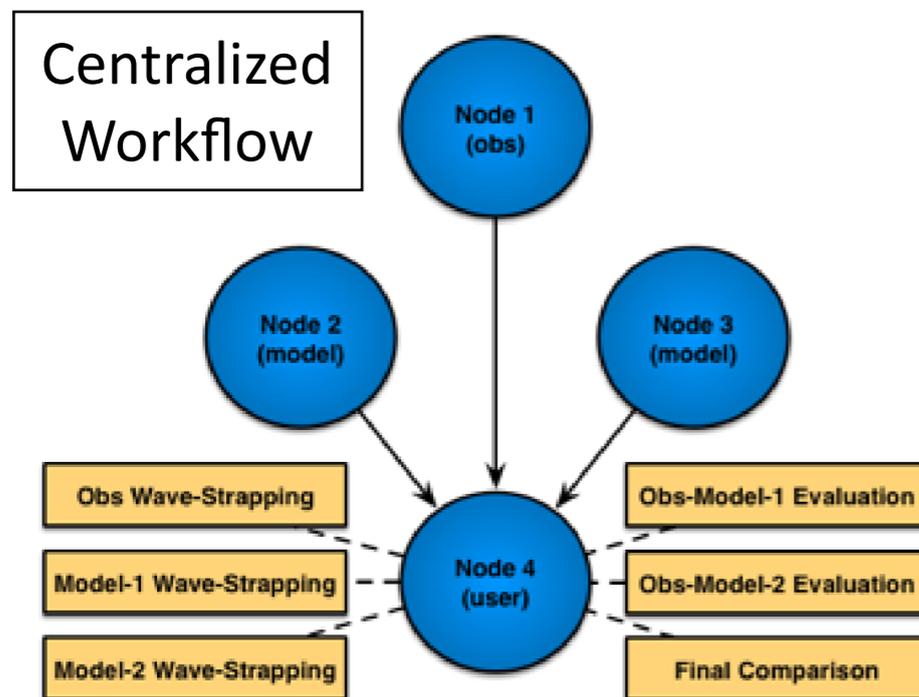
Global evaluation metric (0-100%) for all times, considering statistical uncertainty of the time series

*Climate Use Case analysis performed by Lee Kyo, Amy Braverman.
Many Thanks! For more information, see Amy's poster.*

Climate Science: DAWN Simulation



- Simulation: DAWN was used to compare performance & tradeoffs of 2 architectures:
 - ▶ Centralized workflow: all datasets are transferred from remote servers to user node where analysis takes place
 - *Traditional workflow for climate model evaluation
 - ▶ Distributed workflow: observations are separately transferred to models node, where partial analysis takes place, then partial results are transferred to user node for final comparison
 - *Optionally, sub-sets of data can be used for model evaluation: tradeoff between efficiency and uncertainty

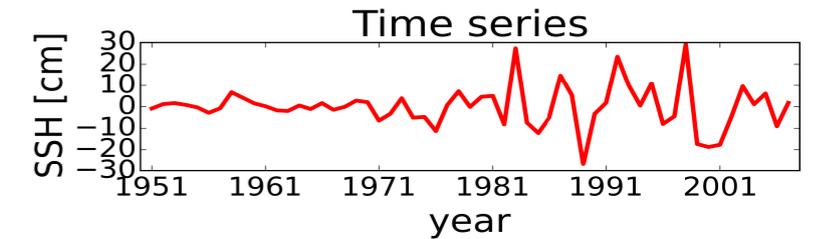


Climate Science: Statistical Analysis

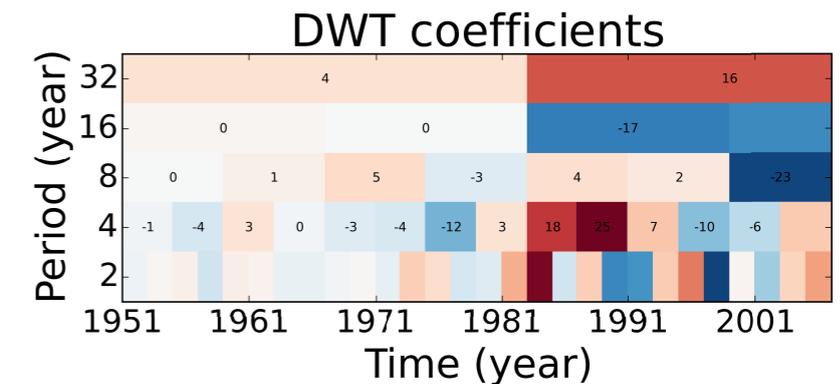


Models & observations were compared with a statistical technique that estimates the uncertainty of a time series by combining “Wave-Strapping” & “Wild Bootstrapping”:

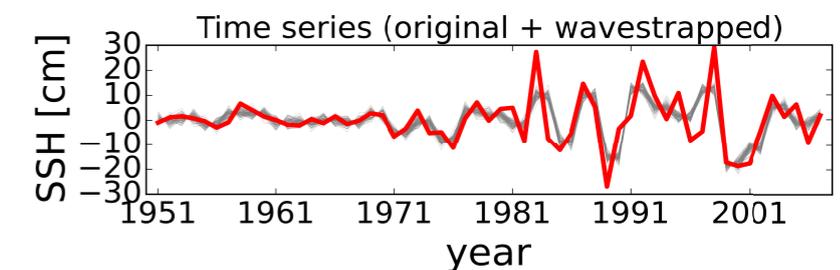
- Observed or simulated time series of key meteorological variables show autocorrelation structure with long memory (which do not quickly decay to zero as time span increases)
- “Wave-Strapping”: Discrete Wavelet Transformation (DWT) is applied to the original time series to produce a set of uncorrelated coefficients [Percival et al, 2000]
- “Wild Bootstrapping”: the DWT coefficients were multiplied by random numbers $N(0,1)$ to create a new time series with random perturbations [Wu, 1986]
- Reconstructed time series are compared for each grid cell to produce an evaluation metric
- When studying large-scale climate variability (e.g.: ENSO), it is possible to wave-strap only certain levels of DWT coefficients (i.e. filter out some levels) to reduce data size



1. Apply DWT to a time series



2. Multiply DWT coefficients by random variables



3. Apply inverse DWT to the perturbed coefficients

Climate Science: Benchmarking the Real Workflows



Working with scientists, we benchmarked the execution of the real analysis workflows using a system of 3 JPL servers (where datasets are stored) + user laptop, and used the measured times for data transfer, computation as DAWN configuration parameters

<u>Centralized Workflow</u>	Real	DAWN
Data Transfer to User Node	00:09:25	00:09:48
Observations Wavestrapping	00:23:30	00:22:30
Model Evaluation	00:37:00	00:34:10
Total	01:09:58	1:06:28

<u>Distributed (no compression)</u>	Real	DAWN
Observations Wavestrapping	01:10:00	01:10:00
Data Transfer to Model Nodes	00:01:06	00:01:24
Model Evaluation	01:23:52	01:23:50
Data Transfer to User Node	00:00:11	00:00:00.1
Total	02:35:28	02:35:34

Results

- Analysis is dominated by computation times (in particular, for higher resolution grid)
- For this topology, centralized workflow is more efficient because user laptop is more powerful than the remote servers, despite the slower network to the user node. But:
 - ▶ In an enterprise environment, remote servers should be more powerful
 - ▶ Storage on user laptop is limited
 - ▶ Real execution did not run any tasks in parallel (not even for distributed workflow)

Climate Science: Idealized Use Case

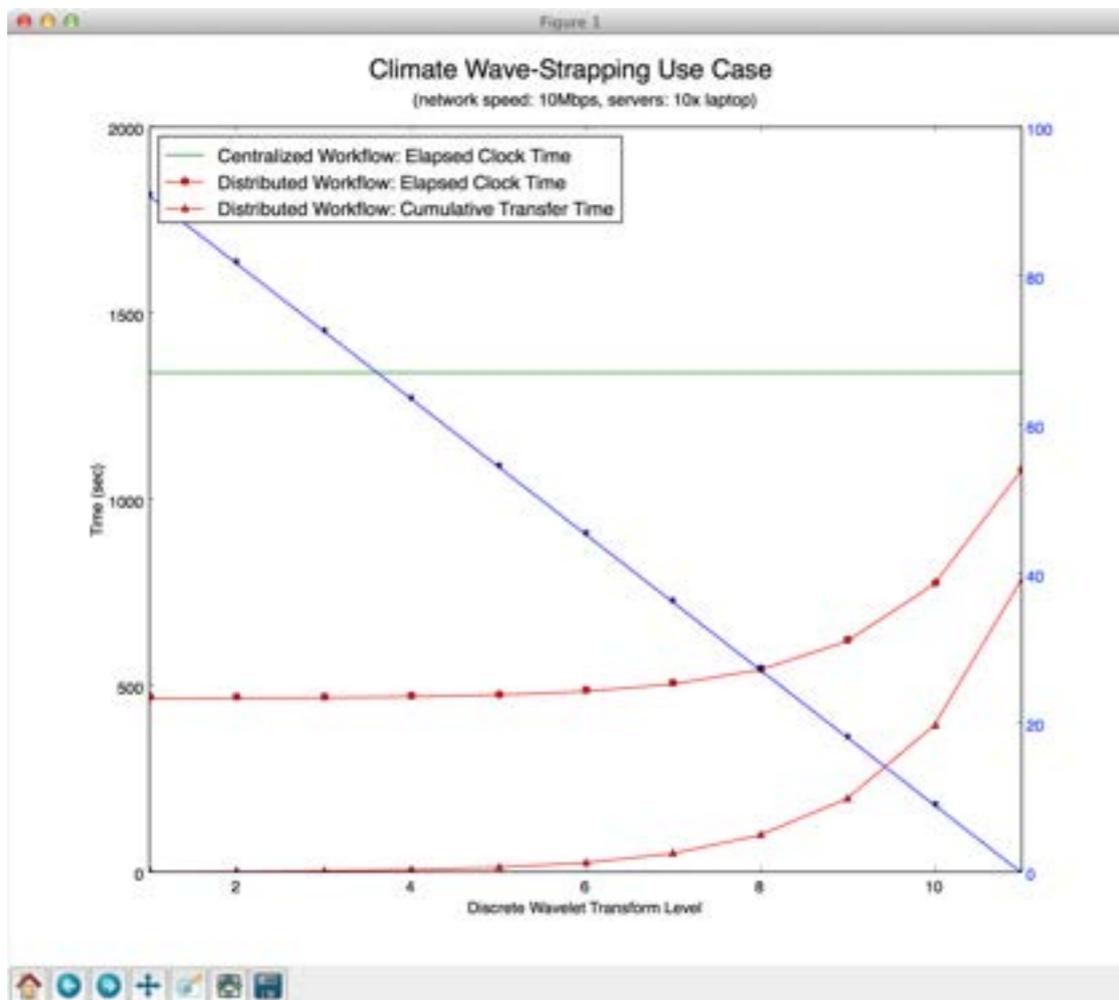


DAWN was used to simulate an idealized use case - closer to enterprise-level analysis:

- Assume all servers with same computation capacity (10x user laptop)
- Assume all network edges with same speed: 10 Mbps (between CMIP5 data servers)
- Execute data transfers and computations in parallel, whenever possible
- Investigate performance, uncertainty as function of DWT reduction level
- Infer size of datasets from number of filtered DWT coefficients

Conclusions

- Data transfer times become significant
- Distributed architecture is more efficient:
 - ▶ Less data transferred over network
 - ▶ Computations are executed in parallel
 - ▶ But still dominated by high res grids
- Advantage in using compressed datasets
- Uncertainty estimate may be used to identify the optimal DWT reduction level



Analysis and prediction of Climate change would greatly benefit from establishing a distributed computational infrastructure (see ESGF).

Summary and Future Work



Summary

- JPL is developing a model (DAWN) for simulating arbitrary data intensive system architectures
- This model has been proven to be applicable to the analysis of use cases from different scientific disciplines, yielding useful insight for optimizing the data processing pipelines
- Our goal is to develop DAWN into a powerful and easy to use tool for the Big Data era in scientific research

Future Work

- Enable DAWN to simulate sub-workflows
- On-demand invocation for dynamic allocation of resources by workflow engines (such as OODT Workflow Manager)
- Provide documentation, example use cases
- Possibly, implement a user interface to generate and run workflows
- Make available as Open Source software on GitHub