National Aeronautics and Space Administration

Server-Side Processing for Large-Scale Data Analytics ESGF Compute Working Team (ESGF-CWT)

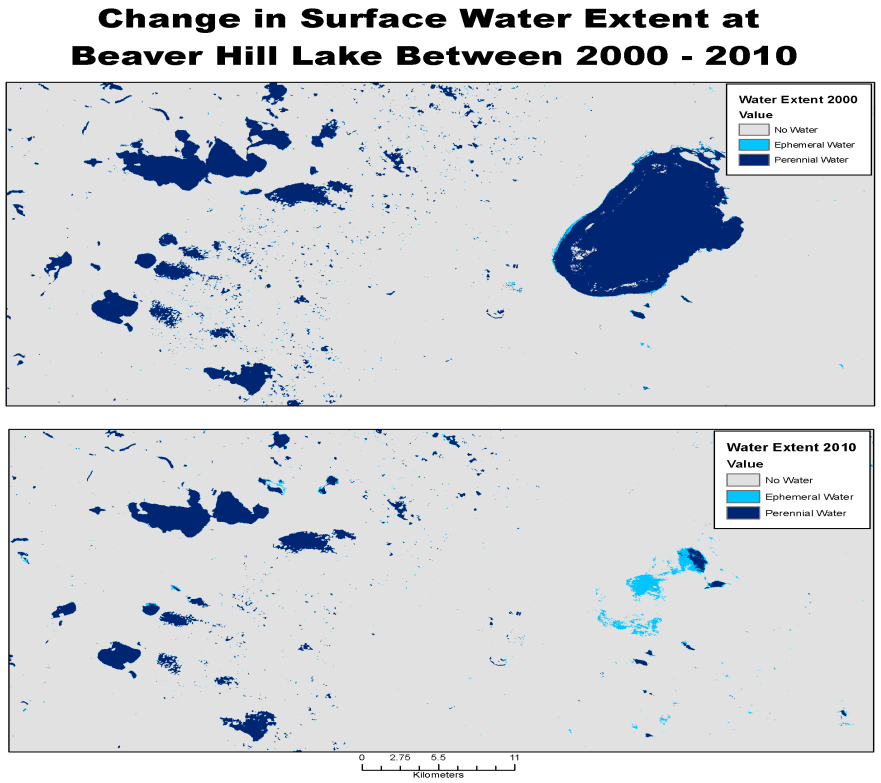
ESGF & UV-CDAT Face-to-Face
December 2014

Daniel Duffy daniel.q.duffy@nasa.gov

NASA Center for Climate Simulation (NCCS)

Goddard Space Flight Center (GSFC)

Arctic Boreal Vulnerability Experiment (ABoVE) Example



- **Tracking the change in surface water over the study region using Landsat**
  - Average across three epochs (1990, 2000, 2010)
  - 25,000 Landsat scenes/~7 TB of data
  - ***Projected time 9 months!***
- **Download the data into our science cloud**
  - 48 virtual machines
  - 6 weeks of data movement and processing
- **Opened up the opportunity to do more processing**
  - Explore the complete Landsat record
  - 100,000 scenes> 20 TB of data

Climate Analytics in ESGF

New methods for climate analytics are needed – access to large distributed data sets through APIs for storage-proximal processing.

- Creation of the ESGF Compute Working Team (ESGF-CWT).
- Co-chaired by Charles Doutriaux and Daniel Duffy
  - International group of ESGF experts (and me for some reason?); meet bi-weekly

High Potential Scientists – The bar is pretty low or "laying on the floor" (Eli Dart)

- Given that 60% to 80% of a project time is spent on data migration and data wrangling, any speed up will be a great improvement
- In other words, if it is going to take 9 months for a researcher to get their science completed, we should not worry (at this time) whether or not our server-side capabilities are going to take 1 hour or 1 day
- Let's get something working and running; we can always optimize later

ESGF-CWT Process and Progress

Charge of the ESGF-CWT

- Develop general APIs for exposing ESGF distributed compute resources (such as clusters, cloud servers and HPC) to multiple analysis tools
- Note that the charge is not to develop the server-side processing capabilities, but rather focus on the API

Process

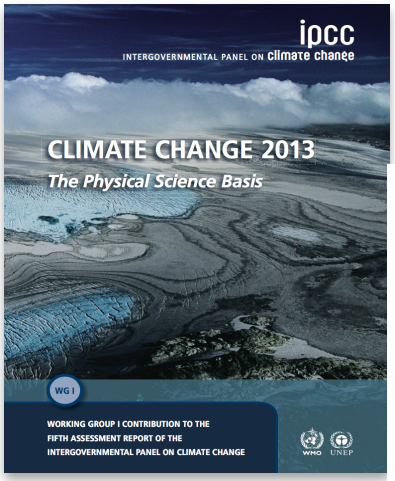
- Started out with a use case
- Worked through the use case to frame our thoughts and requirements
- Used the Goddard Climate Data Services (CDS) API and server-side processing capability as a driver for the conversation
- Compared and contrasted different viable APIs to settle in on a consensus

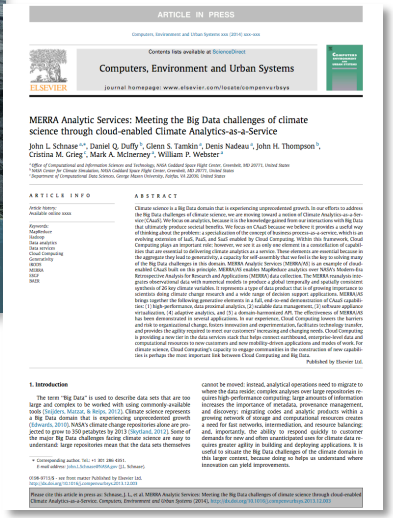
Please Note – This talk will highlight an implementation of server-side processing. This is being used to illustrate the power of server-side analytics; not a pitch for everyone to adopt this approach.

Climate-Analytics-as-a-Service (CAaaS)

How much climate data?

- NASA MERRA Reanalysis Collection ~200 TB
- Total data holdings of the NASA Center for Climate Simulation (NCCS) is ~40 PB
- Intergovernmental Panel on Climate Change Fifth Assessment Report (AR5) ~2-5 PB
- AR6 estimated to be 5x (?) AR5
- Reference: *MERRA Analytic Services: Meeting the Big Data challenges of climate science through cloud-enabled Climate Analytics-as-a-Service,* Schnase, et. al, Computers, Environment, and Urban Systems
  - doi:10.1016/j.compenvurbsys.2013.12.003
  - Work funding by NASA under the CMAC Program (T. Lee)





Components of a CAaaS

MERRA Reanalysis

Data

Relevance and Collocation

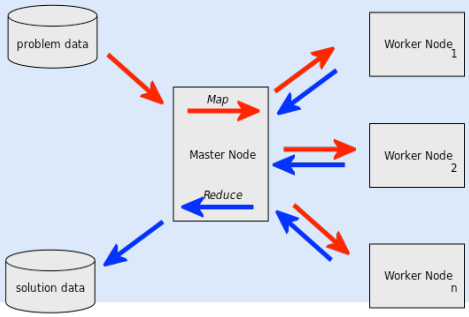
Data have to be significant, sufficiently complex, and physically or logically co-located to be interesting and useful …

High-Performance Compute/Storage Fabric

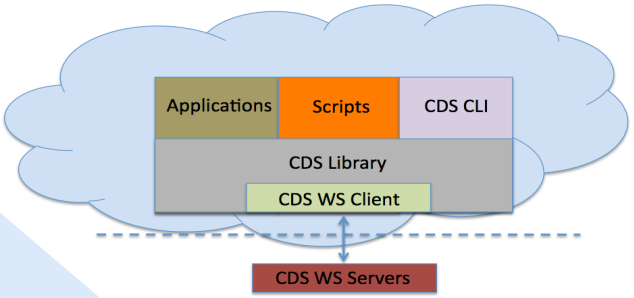
Storage-proximal analytics with simple canonical operations

Data do not move, analyses need horsepower, and leverage requires something akin to an analytical assembly language …

MERRA Analytic Services



Climate Data Services API



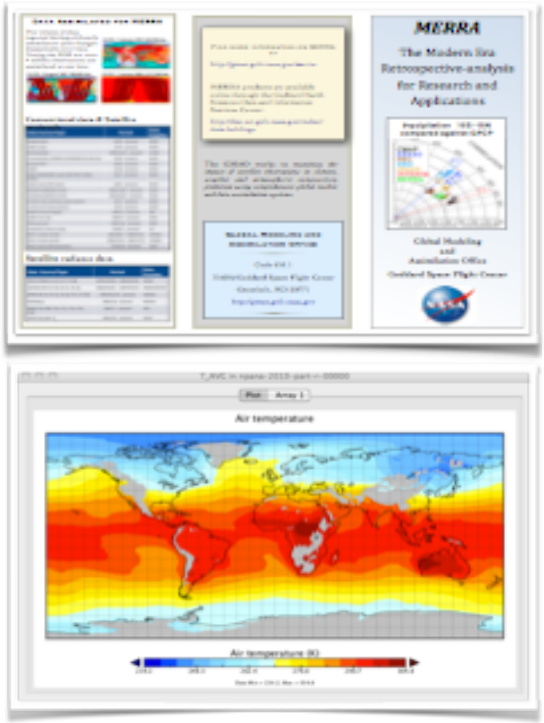
Exposure

Convenient and Extensible

Capabilities need to be easy to use and facilitate community engagement and adaptive construction …

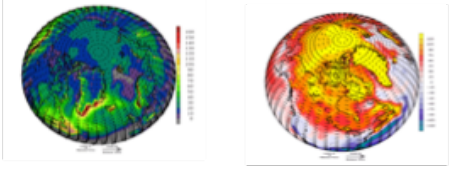
MERRA Data Set

MERRA Reanalysis



Modern Era-Retrospective Analysis for Research and Applications

- Source: Global Modeling and Assimilation Office (GMAO)
- Input: 114 observation types (land, sea, air, space) into "frozen" numerical model. (~4 million observations/day)
- Output: a global temporally and spatially consistent synthesis of 26 key climate variables. (~418 under the hood.)
- Spatial resolution: 1/2° latitude × 2/3° longitude × 42 vertical levels extending through the stratosphere.
- Temporal resolution: 6-hours for three-dimensional, full spatial resolution, extending from 1979-Present.
- ~ 200 TB, but MERRA II is on the way …



ESGF MERRA published variables

CMIP5	MERRA	Units	Description(Long Name)
rlus	rlus	W m-2	Surface Upwelling Longwave Radiation
rlut	lwtup	W m-2	TOA Outgoing Longwave Radiation
rlutcs	lwtupclr	W m-2	TOA Outgoing Clear-Sky Longwave Radiation
rsds	swgnt	W m-2	Surface Downwelling Shortwave Radiation
rsdscs	swgdnclr	W m-2	Downwelling Clear-Sky Shortwave Radiation
rsdt	swtdn	W m-2	TOA Incident Shortwave Radiation
rsut	swtdn??	W m-2	TOA Outgoing Shortwave Radiation
clt	cldtot	%	Total Cloud Fraction
pr	prectot	kg m-2 s-1	Precipitation
cl	cloud	%	Cloud Area Fraction
evspsbl	evap	kg m-2 s-1	Evaporation
hfls	eflux	W m-2	Surface Upward Latent Heat Flux
hfss	hflux	W m-2	Surface Upward Sensible Heat Flux
hur	rh	%	Relative Humidity
hus	qv	v	Specific Humidity
prc	preccon	kg m-2 s-1	Convective Precipitation
prsn	precsno	kg m-2 s-1	Snowfall Flux
prw	tqv	kg m-2	Water Vapor Path
ps	ps	Pa	Surface Air Pressure
psl	slp	Pa	Sea Level Pressure
rlds	lwgnt	W m-2	Surface Downwelling Longwave Radiation
rldscs	lwgabclr	W m-2	Surface Downwelling Clear-Sky Longwave Radiation
rsutcs	swtdn	W m-2	TOA Outgoing Clear-Sky Shortwave Radiation
ta	t	K	Air Temperature
tas	t2m	K	Near-Surface Air Temperature
tauu	taux	Pa	Surface Downward Eastward Wind Stress
tauv	tauy	Pa	Surface Downward Northward Wind Stress
tro3	o3	1.00E-09	Mole Fraction of O3
ts	ts	K	Surface Temperature
ua	u	m s-1	Eastward Wind
uas	u10m	m s-1	Eastward Near-Surface Wind
va	v	m s-1	Northward Wind
vas	v10m	m s-1	Northward Near-Surface Wind
wap	omega	Pa s-1	omega (=dp/dt)
zg	h	m	Geopotential Height

Simple ABoVE Related Example

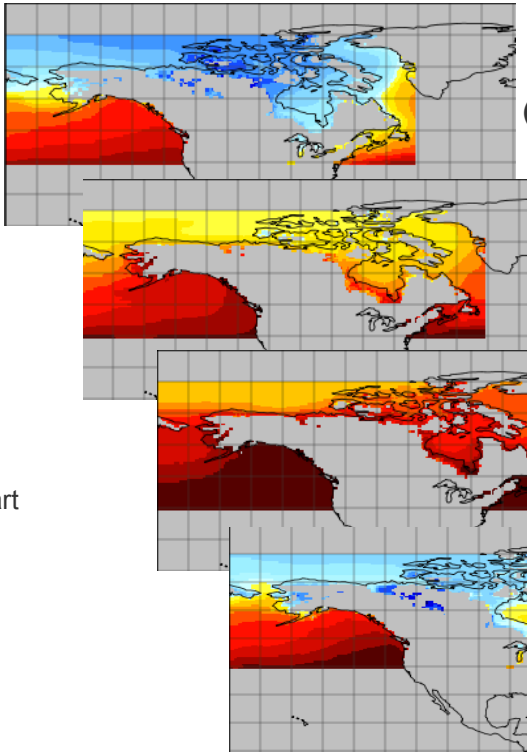
QUESTION: Extract the average temperature by season for the year 1980 for the ABoVE region at every level in the MERRA reanalysis data.

```
#!/usr/bin/env python
‘import sys
from CDSLibrary import CDSApi
cds_lib = CDSApi()
service = "MAS"

name       = "above_avg_seasonal_temp_1980_instM_3d_ana_Np"
job        = "&job_name=" + name
collection = "&collection=instM_3d_ana_Np"
request    = "&request=GetVariableBy_TimeRange_SpatialExtent_VerticalExtent"
variable   = "&variable_list=T"
operation  = "&operation=avg"
start      = "&start_date=198001"
end        = "&end_date=198012"
period     = "&avg_period=3"
space      = "&min_lon=-180&min_lat=40&max_lon=-50&max_lat=80"
levels     = "&start_level=1&end_level=42"
file_job_epoch1_aveT  = "./" + name + ".nc"
above_job_epoch1_aveT = job + collection + request + variable + operation + start
+ end + period + space + levels

class UserApp(object):
    if __name__ == '__main__':

        cds_lib.avg(service, above_job_epoch1_aveT, file_job_epoch1_aveT)
```



Winter (Jan, Feb, Mar)

Spring (Apr, May, Jun)

Summer (Jul, Aug, Sep)

Fall (Oct, Nov, Dec)

How are we doing this? Hadoop Infrastructure

Hardware Configuration

- 36 node Dell cluster (11.7 TF Peak)
- 576 total cores (Intel 2.6 GHz SandyBridge)
- 2,304 GB of RAM (64 GB per node)
- 1,296 TB of RAW storage (12 x 3 TB disk drives – 36 TB per node)
- FDR Infiniband

Virtual Hadoop Clusters

- Three Hadoop clusters on the same hardware (using containers)
  - Test, Pre-Production, Production
- Very agile way of testing new software and promoting the software changes into production

Reconfiguration of the Disk Layout

- Each disk now is its own separate volume within Hadoop; greatly reduces disk contention across the system

Increase of Memory

- The amount of memory has a direct effect on the overall performance; doubled the memory

Towards Operational Deployment

- Using standard NCCS practices for configuration management and authentication

Batch Processing NetCDF Data in HDFS

Sequence the Data

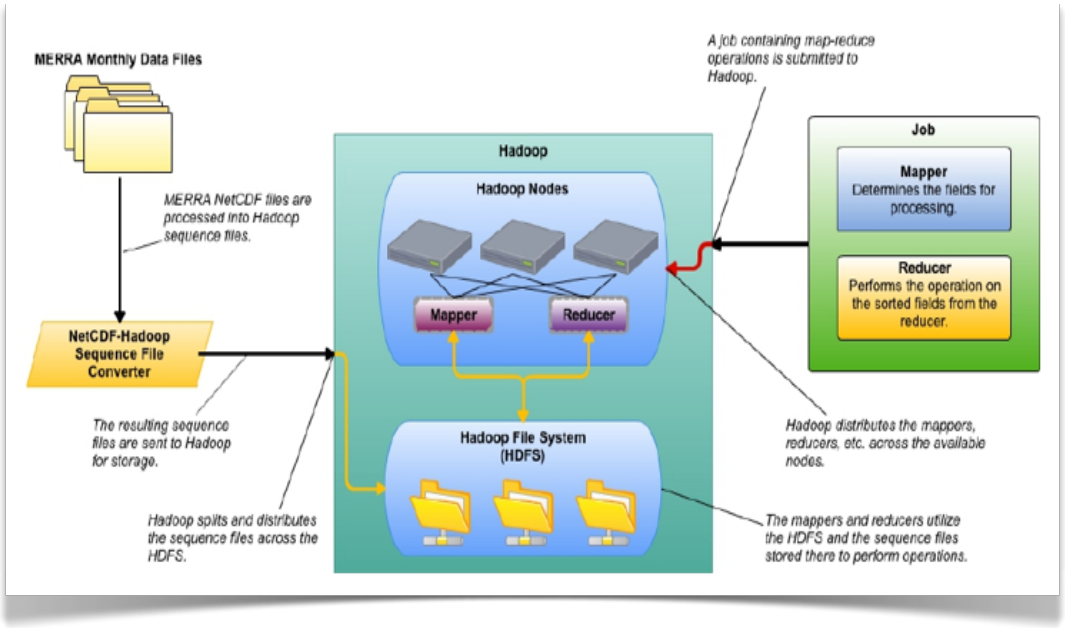
- Sequence the NetCDF data into HDFS
- Basically introduces <key, value> pairs
- No longer NetCDF (key point)

Run MapReduce

- Simple canonical operations of ave, max, min, sum, count, var

De-sequence the Results

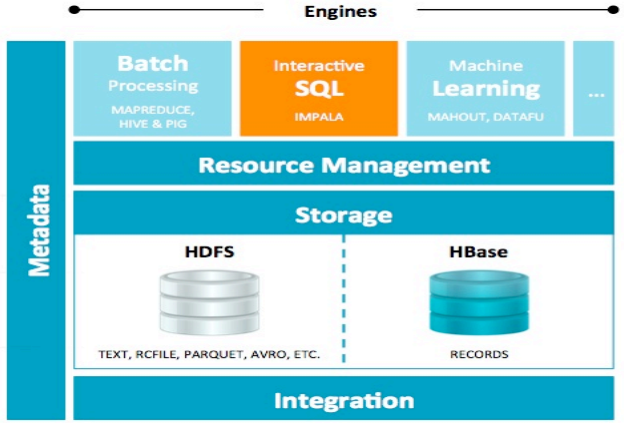
- Translate the result from the <key, value> object back into NetCDF



Moving Toward Real-Time Analytics

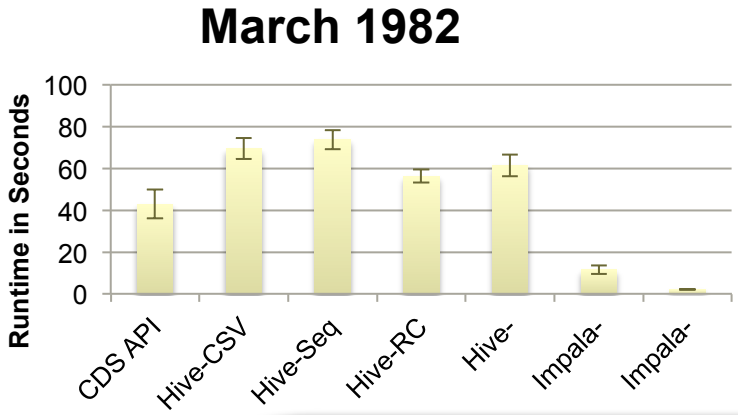
HDFS Software Stack

- Industry has continued to make major additions and improvements to the Hadoop software stack
- In addition to MapReduce, we have evaluated
  - » HIVE and Impala (Data Wharehouse)
  - » Different file formats – CSV, RC, Parquet
  - » SQL-Like Queries



Initial Results

- Dramatic improvements in performance
- Impala-Parquet shows the best performance thus far
- Figure shows the time required to query a single monthly mean for average Temperature



What Next

- Continue to evaluate other technologies, including Spark

Options for NetCDF and HDFS (Do we have to make another copy of the data?)

Using Native NetCDF Data and MapReduce

- Build an abstraction layer between NetCDF and HDFS to enable MapReduce operations
  - Exploring this possibility with George Mason University (poster at AGU)
- Possible to introduce/expose <key, value> pairs natively within NetCDF
- *Major benefit – do not have to make a separate copy of the data*
- Potential use of high performance file systems (GPFS, Luster, Gluster, etc.)
  - Planning a study on this at GSFC
- Both Posix and object interface to the same data
- Good for batch processing

Convert NetCDF Data into HDFS

- Sequence the data into <key, value> pairs or use specialized data formats
- Build an abstraction layer between HDFS to allow applications to see the data as native NetCDF (FUSE-like interface)
- *Major drawback – Have to make a separate copy of the data*
- Natively use the HDFS ecosystem of tools
- Both Posix-like and object interface to the same data
- Good for more interactive and near real-time processing

The answer may be to use a combination of both of these – using MapReduce processing on native NetCDF data for batch processing and create copies of commonly used variables (~10% of the data) into native HDFS file formats for interactive and near real-time processing.

ESGF-CWT Representative Use Case – Anomaly

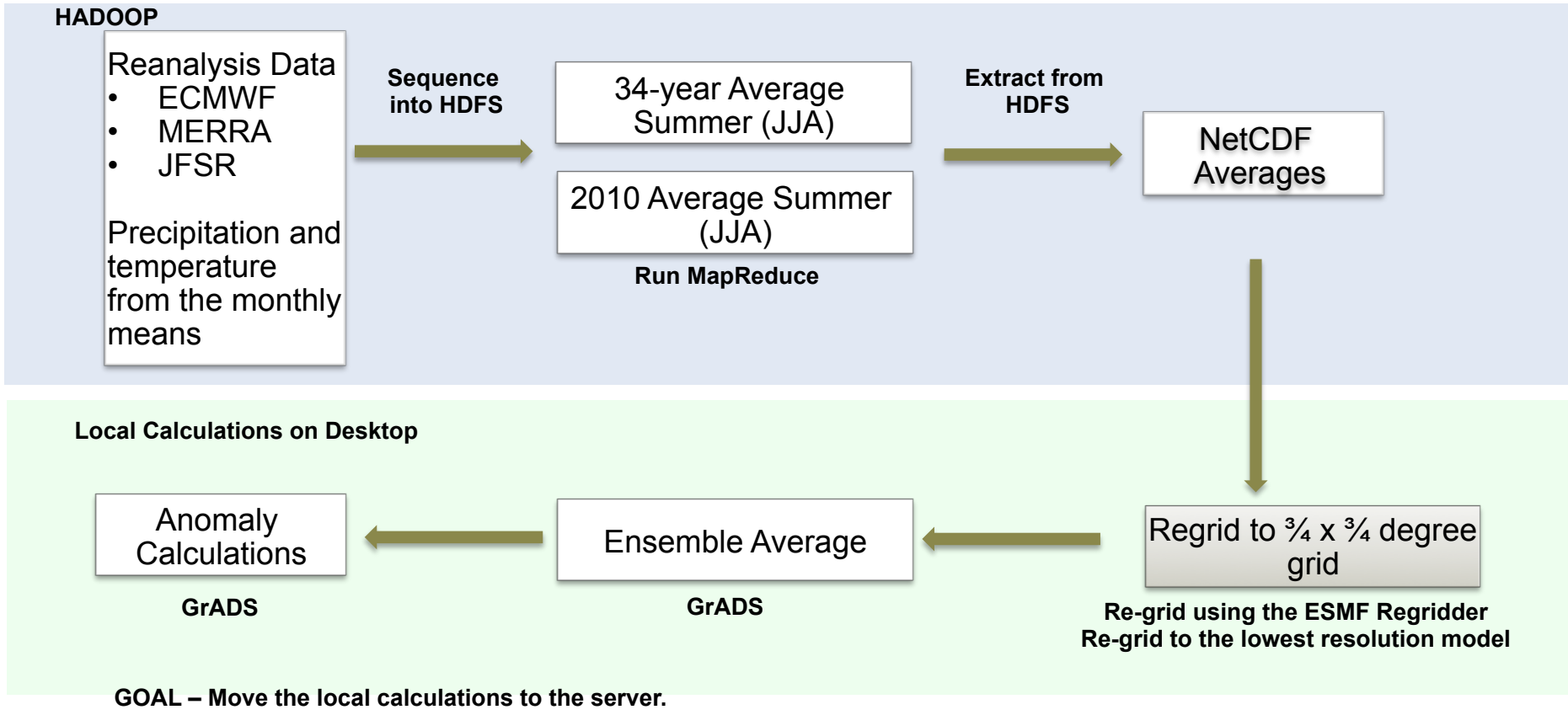
Use Case Statement: Multi-Model Averaging

- Generate an average of a variable over many models across federated data.
- Specify the temporal and spatial extent over which the averaging will be done
- ***Immediate Question:*** How do we handle the grids (space and time) and resolution for each of the models?
  - Use the CMIP5 regridder (ESMF) – resulting steps for the calculation
    - Create the average for each of the individual models.
    - Regrid the models to a common grid.
    - Compute the average across all the regridded data.

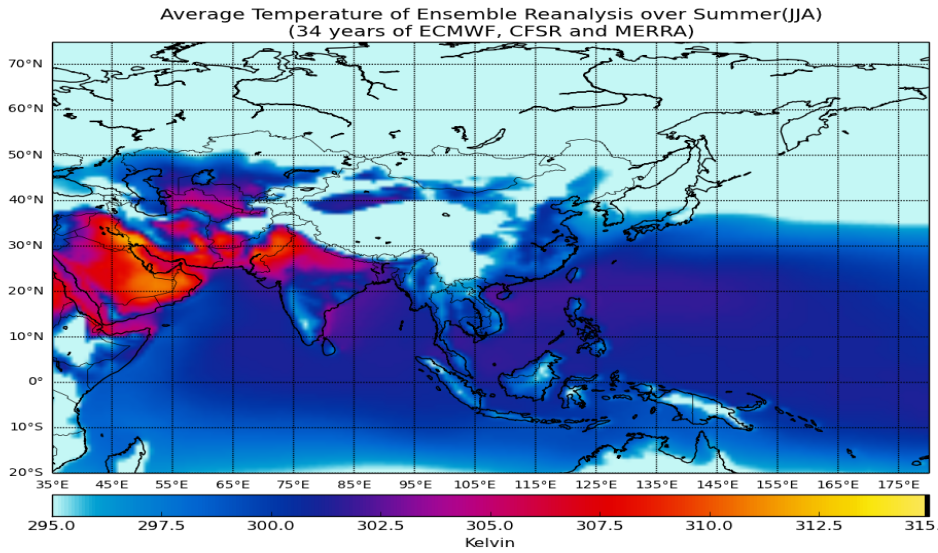
Use Reanalysis Data to Work Through This Use Case

- **Representative Questions**
  - Compare and contrast the features from the various reanalysis data sets.
  - Does an ensemble reanalysis provide a more accurate representation of the historical climate?
  - Can we generate uncertainty quantification using multiple reanalysis data sets?

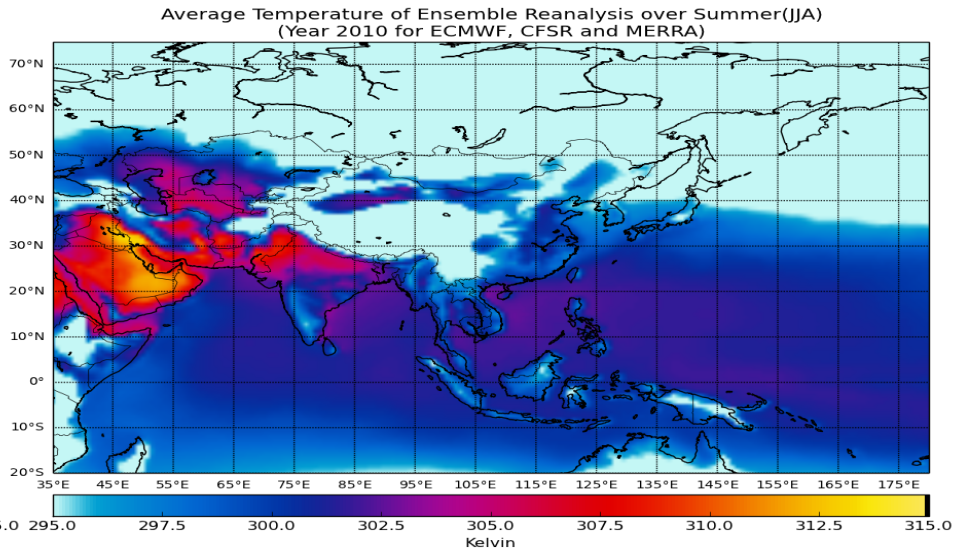
Representative Use Case Workflow Using Reanalysis Data
Work performed by Denis Nadeau (GSFC)



Compute Ensemble Surface Temperature Averages

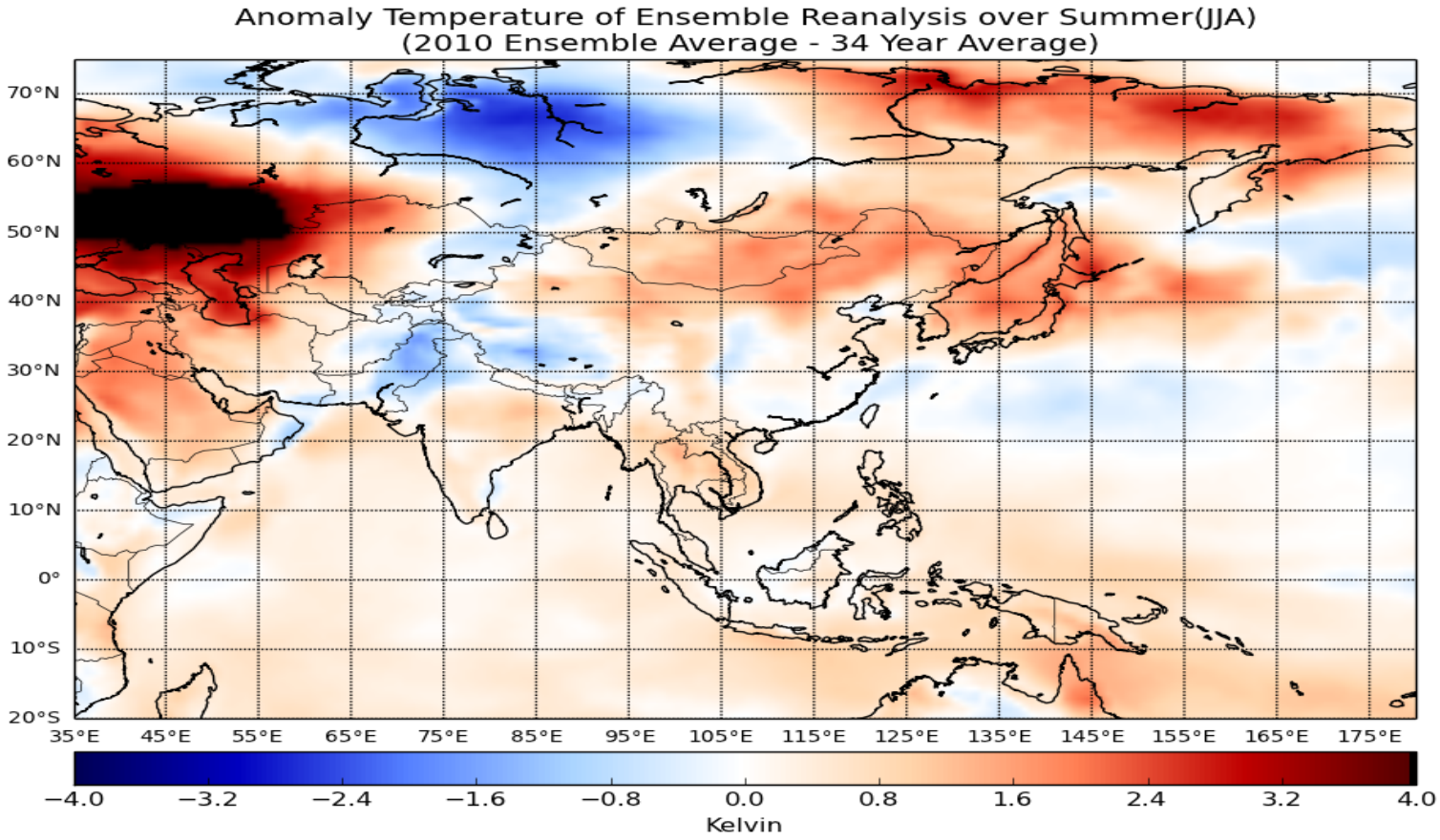


34-Year Ensemble Average over Summer (JJA)

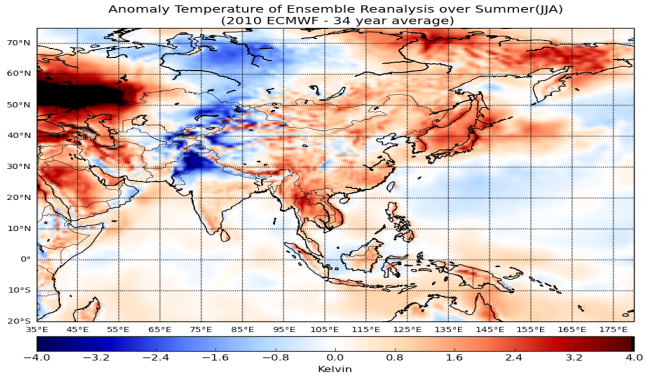


2010 Ensemble Average over Summer (JJA)

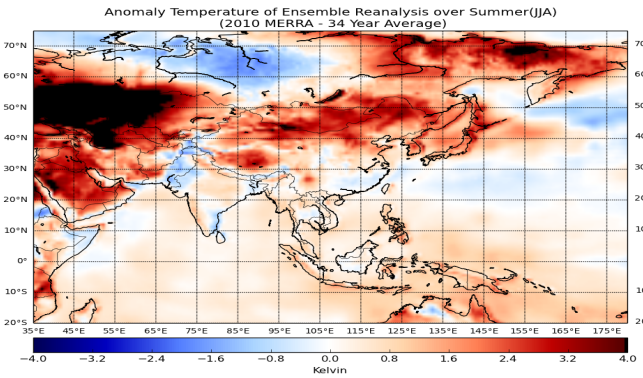
Surface Temperature Anomaly of the Ensemble Average



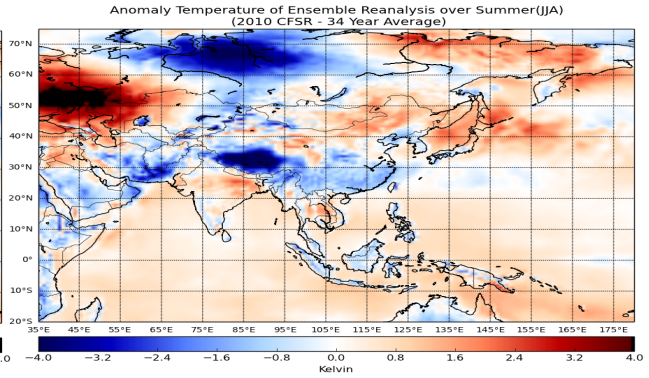
Surface Temperature Anomaly of the Independent Reanalysis Compared to the 34-Year Ensemble Average



2010 ECMWF – 34 Year Ensemble Average over Summer (JJA)



2010 MERRA – 34 Year Ensemble Average over Summer (JJA)



2010 CFSR – 34 Year Ensemble Average over Summer (JJA)

- Temperature variability in the pictures is due to such things as topographical features, different cloud parameterizations within the models, etc.
- Departure of average reanalyses over summer 2010(JJA) shows that ECMWF surface temperature is generally colder than MERRA.
- Himalayas have quite different values!

GSFC Climate Data Services (CDS) API

Actions or Methods

Order – Request data from a pre-determined service request (asynchronous).

Status – Track the progress of an order.

Download – Retrieve a Dissemination Information Package (DIP).

Ingest – Ingest a Submission Information Package (SIP).

Execute – Initiate a service-definable extension. Allows for parameterized growth without API change.

Query – Retrieve data using a pre-determined service request (synchronous).

Client Package (Beta)

Python tarfile that contains source code, build files, simple end to end use cases, and documentation to access the MERRA data exposed through a RESTful interface.

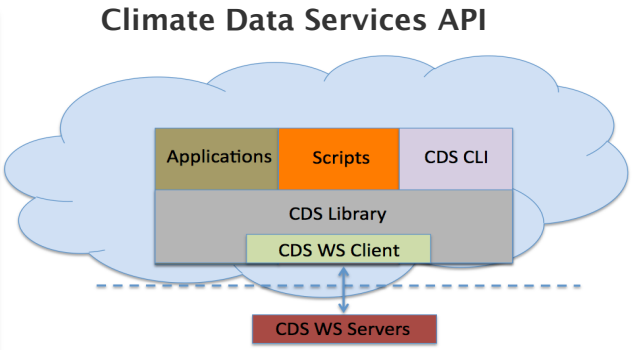
Simple Canonical Operations

The API exposes relatively simple operations to the end user:

Ave, Max, Min, Sum, Count, Var

Spatial-Temporal Extent

Users can specify a spatial bounding box and time over which to perform the canonical operation.



RESTFul Interface

Users can write a python script on their machine that contacts a RESTful web service to execute the operation.

At this point, the answer is downloaded as a NetCDF output file to their machine.

How Can We Expose This Capability Through an API

Using the Goddard CDS API that was used for the representative use case, the ESGF-CWT compared and contrasted several APIs and services

- CDS API
- ESGF
- WPS

The focus was on both general concepts of APIs but also the compatibility with the existing community of software and tools.

- Quite a bit of discussion about this
- WPS was the consensus of the team for several reasons, including the familiarity of the community to WPS and existing tools

Next Steps Over the Next Year

Data Proximal Analytics Technology at Goddard

- Continue the exploration of HDFS and the ecosystem (specifically Spark)
- Spark (in memory computing)
- Exploration of high performance file systems
- Continue development of the CDS API; compatibility with the ESGF WPS

Application Programming Interface by the ESGF-CWT

- Specification of a WPS API for ESGF
- Reference implementation
- Always looking for more volunteers for this group!

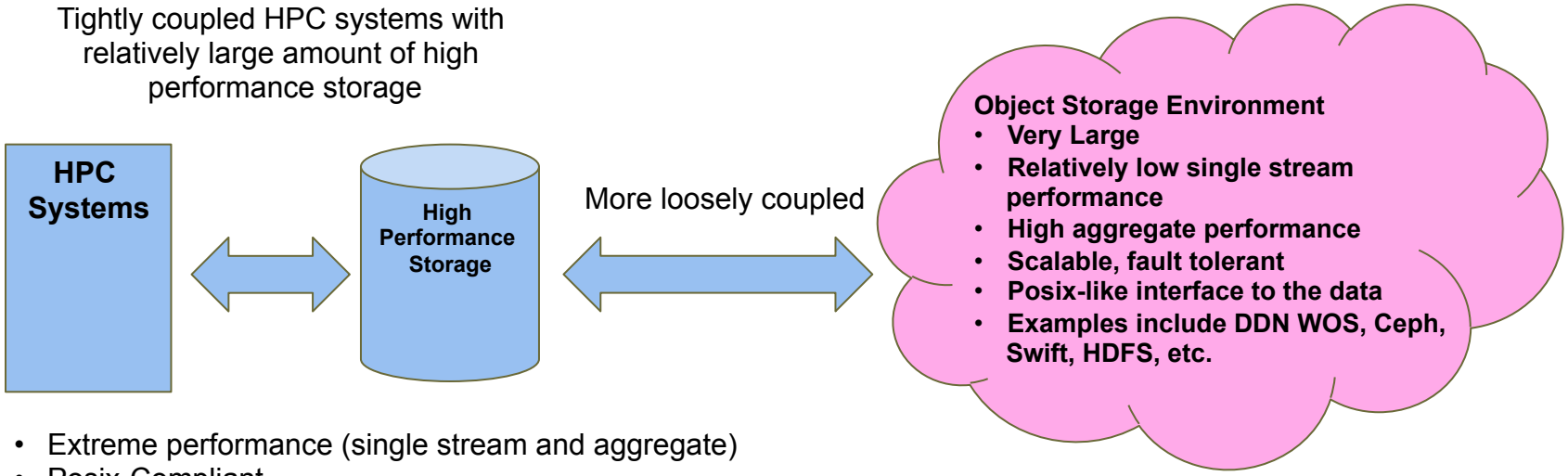
Science

- Compare observations to the ensemble reanalysis
- Work on uncertainty quantifications of reanalysis data
- Engage Goddard scientists to help drive requirements

Future of High Performance Data Storage (Dan's Opinion)

The concept is to both surround and permeate the object storage with compute resources that can be used for analytics.

Tightly coupled HPC systems with relatively large amount of high performance storage



- Extreme performance (single stream and aggregate)
- Posix-Compliant
- Typical examples include GPFS, Lustre, etc.
- Could contain some type of burst buffering capability.

Call it a cloud if you want (or not)! The fact is that we are going to get compute for "free" with the storage in the future, all the way down to the hard drive (check out the Seagate Kinetic Open Storage initiative). We should be working to exploit these capabilities.

Thank You

Very special thanks to

- Dean “I Never Sleep” Williams
- Charles “The New Father” Doutriaux
- And all the members of the working group, many of which I am meeting in person for the first time – I did not want to list all the names and either make mistakes or forget someone
  - Spell check does not work on your names!
- My collaborators at GSFC, including a special thanks to
  - Glenn “API Guru Currently in Iceland” Tamkin
  - Denis “Frantically Writing his AGU Poster” Nadeau