Distributed Resources for the Earth System Grid Advanced Management (DREAM)

2016 Annual Report

NASA Goddard Space Flight Center
NASA Jet Propulsion Laboratory
DOE Lawrence Livermore National Laboratory (lead institution)
Princeton University
University of Utah

Accomplishments

The over-arching goal of the DREAM project is to provide a standard-based system architecture, and a reference implementation, for managing, accessing and analyzing large data resources stored at distributed facilities. DREAM proposed strategy is to leverage the existing software stack of the operational Earth System Grid Federation, and evolve it to make it more modular, more deployable, more scalable, and encapsulate each service through a simple yet powerful REST API. DREAM measure of success will be the ability to employee this new architecture in the Climate science domain, as well as applying it to other scientific fields such as Biology and Hydrology.

DREAM funding was available at the participating institutions in the second half of 2016. Since then, the team has worked at the following tasks, and accomplished the following milestones.

1. First version of Docker-based ESGF architecture

The DREAM team has agreed on using Docker as the core technology to modularize and generalize the ESGF architecture, as necessary pre-requisite to port the ESGF infrastructure to other scientific domains. Docker is the industry leading "containerization" technology, which allows applications to be built as "black boxes" that contain all the necessary software to run them, and can be deployed on any Docker-enabled host. The team has undertaken the monumental task of converting the current ESGF installation software into separate Docker images, which can be run together as a set of interactive containers. At this time, most of the major ESGF components have been converted to Docker (ESGF search, Identity Provider, Openid Relying Party, Thredds Data Server, Solr, CoG, Apache httpd, Postgres) and can be run together to stand up a full ESGF Node (see Figure 1). In this architecture, all of the specific site configuration is restricted to a location on the local host, outside of

the Docker containers, so that it can survive image upgrades. Additionally, the data used by each application (the Postgres database, the Solr index, the TDS catalogs, the CoG site data) are stored on specific Docker volumes, which are managed by Docker independently of the containers, so that they can also persist through software updates.
One of the many advantages of Docker is the ability to mix-and-match images into more complex applications by writing "docker-compose" configuration files. This functionality was leveraged to create alternative versions of an ESGF Node, specifically an ESGF Data Node only, an ESGF Index Node only, and an ESGF Index Node with Solr Cloud support. This is extremely important as it showcases the ability of ESGF to create target architectures that are specifically suited to a data center or environment (for example, a Tier 2 ESGF node that only needs to serve data to a parent Index Node), and it enables scaling to multiple host clusters, and on commercial cloud environments.

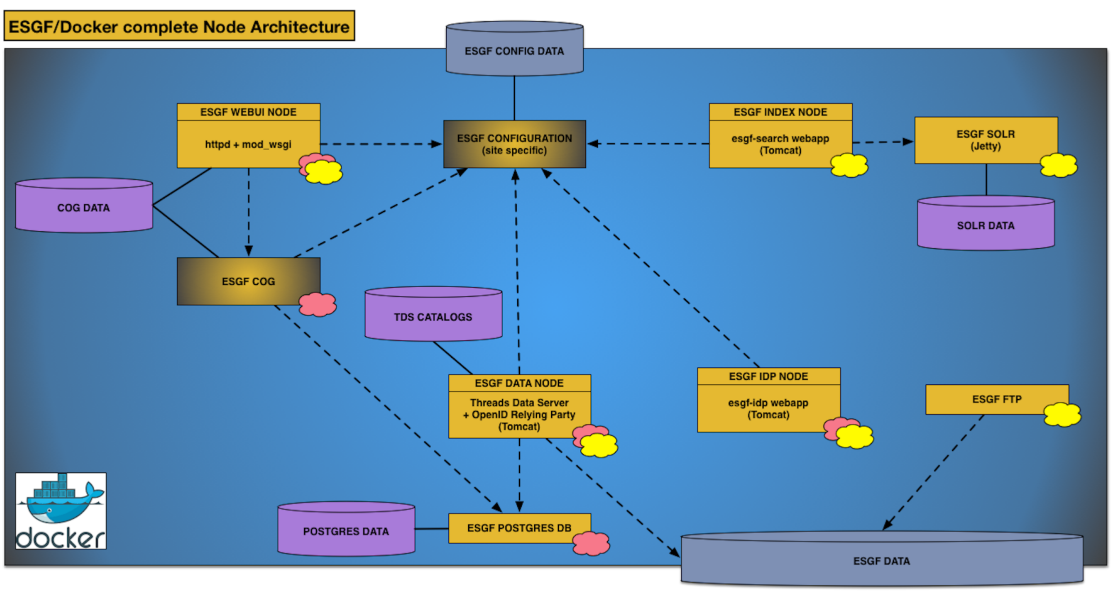

Figure 1: Docker-based architecture for a complete ESGF Node.

2. Distributed Computing (Server and Client side Computing APIs)

Over the last year the Earth System Grid Federation (ESGF) Compute Work Team (ESGF-CWT) made great strides toward a functional working system.
First of all, an Application Programming Interface (API) has been completed. This allows developers to implement services for a compute node. The API is based on the web processing services (WPS) standards and is language agnostic. This means developers can use the language or technology of their choice to create services. It was very important to ensure that this API imposes no technological restriction whatsoever for the implementation. The API description is available at: https://docs.google.com/document/d/1GSLwSJPUCfs-ZrYCG1n7BcWyldvb4pT9nYD3zkPCwao

This allowed various institutions to implement a few services in various way. Technologies used at the moment include the Ultrascale Visualization Climate Data Tools (UV-CDAT), Scala and Ophidia. Examples of implemented services include average, ensemble computation (including re-gridding across models), data aggregating and sub-setting.

A "test" server is being designed to allow institutions to easily test and/or deploy these basic services. Currently the server is based on django. To ensure easy deployment, Docker images are provided. This allows a potential user to be up and running with a fully functional and configured server in a matter of minutes.
Server code is at: https://github.com/ESGF/esgf-compute-wps.
Installation instructions can be found at: https://github.com/ESGF/esgf-compute-wps/blob/master/README.rst.

Finally, to facilitate the adoption and use of these services, an end-user (aka client) Python-based API has been implemented. With this API a user can easily access and use services on a CWT server. The API also allows the user to chain together various CWT services. The end-user API can be found at: https://github.com/ESGF/esgf-compute-api. A set of tutorials for this API can be found at:
https://github.com/doutriaux1/jupyter-nb/tree/master/cwt

The ESGF-CWT holds a general meeting on the first Monday of each month as well as a developers-oriented meeting on the third Monday of each month. Meeting notes can be found at: https://acme-climate.atlassian.net/wiki/pages/viewpage.action?pageId=3997828

The core CWT API has been implemented in Python. We're currently testing it with various server implementations and working on flushing out bugs and adding features. Workflow support is a feature we're currently working towards for the ACME and CMIP DOE efforts.

3. Streaming Data Visualization

As part of the DREAM infrastructure the current deployment strategy is complemented with continuous development of new capabilities. The focus of this work is on the challenges of retrieving, converting, resampling, and combining remote and often disparately located data ensembles. We have developed a first prototype of embedded domain-specific language (EDSL) specifically designed for the interactive exploration of large scale, remote data. The current EDSL prototype is based on a Javascript interpreter and allows users to express a number of data analysis operations in a simple and abstract manner. To demonstrate the effectiveness of this approach the framework adopted a runtime system that resolves issues such as remote data access and resampling while at the same time maintaining interactivity through progressive and interruptible computation.

For the practical demonstration of the results we have used a prototype infrastructure with an Apache server located at LLNL that can connect to a NASA server using the OPeNDAP protocol. This allows to query indirectly and explore interactively the massive dataset 7km NASA GEOS-5 Nature Run. In particular, each query directed to the LLNL server allows to build a cache, which is stored in IDX format and enable subsequent, fine granularity queries. The results of such queries are delivered to remote clients that can engage in interactive data analysis and exploration sessions in streaming mode.
The results of this work have been presented at the 2016 AGU fall meeting and published in the proceedings of the 6th IEEE Symposium on Large Data Analysis and Visualization.

4. Climate Data Analysis

We build upon prior work undertaken during the ExArch Project aimed at "Climate Analytics on Distributed Exascale Archives" (see e.g http://slideplayer.com/slide/3417446/). The aim of this work under DREAM funding is to be able to deploy analysis packages developed by individual scientists and deploy them for the use of other analysts around the world, portable to any ESGF node that has the capabilities outlined above in (2). The ESGF Docker architecture outlined in (1) provides the capability to maintain and share unique custom environments for each such package, avoiding dependency conflicts with other packages. Sharing within the Docker architecture allows individually developed analysis packages to "scale" to community-wide impact.

At the recent ESGF F2F meeting we presented current status of analytic frameworks ("Earth System Model Development and Analysis using FRE-Curator and Live Access Servers: On-demand analysis of climate model output with data provenance", by Aparna Radhakrishnan, V.Balaji, Roland Schweitzer, Serguei Nikonov, Kevin O'Brien, Hans Vahlenkamp, Eugene Francis Burger).

We also provided results on our estimations of projected data volumes across all of CMIP6, which provides the upper bound on necessary scaling for such an analysis architecture. See CMIP6 "Impact" on Scientific Community (Sergey Nikonov, V. Balaji, Aparna Radhakrishnan, Nalanda Sharadjaya , Hans Vahlenkamp).

5. Analysis of RESTful API Description Languages

One of DREAM goals is to expose each service module through a well defined RESTful API that hides internal implementation details, and can be used by ubiquitous HTTP clients to interact with the service. To this goal, the team has conducted a survey of the currently available standards and languages for describing REST APIs. The result was that many such competing languages are currently in use, without a clear winner that could be identified. Ultimately, the team has decided to adopt RAML (REST API Markup Language) to describe all DREAM service APIs, because of its maturity, its expressive power, and its modular constructs. RAML seems to be the best choice to describe APIs that are independently designed and implemented. Another possibility was to adopt Swagger, which is instead more like a framework of tools to define APIs first, then build client and server side software to implement them and use them. Fortunately, there are tools that can be used to automatically convert RAML specifications to Swagger, and vice-versa.

6. Big Data Requirements for Hydrology

During 2016, the team has started involvement in the Hydrology domain by executing a careful analysis of the data challenges involved with understanding and predicting water availability across the U.S. The resolution of this problem necessitates the combined analysis on massive hydrological datasets from disparate sources, including model output data, in-situ observations, remote sensing, historical data, and others. Other key computing challenges described in the document include orchestrating different model execution environments, optimizing data movement, working with distributed data archives, running analytics in real time, and development of machine learning algorithms.

7. Applications to Biology

In order to enhance the ESGF publication services for DREAM to support biology, we have performed several proof-of-concept publications. In preparation, we had several datasets in epidemiology as identified by an expert in that domain, namely the project Tycho sets of US historical and a PAHO set for the dengue virus. We provided a esg-publisher data handler that accepts the ASCII or binary (not-netCDF) formats. Additionally, we identified the controlled-vocabulary and properties to use for the datasets, provided this information to the publisher tool, and configured the CoG front-end on our test server to properly display the dataset properties.

Future Milestones

In the next 6-12 months, the DREAM team is planning to progress on the following tasks:

- <u>Bring the ESGF Docker architecture to fully operational status</u>. Several tasks need to be completed before a Docker ESGF node can be deployed into production. This includes converting some of the remaining ESGF components to Docker (Globus, OAuth2 server, MyProxy, the ESGF Publisher); setting up an automated procedure for building images; possibly developing a Docker-based test-suite; and accomplishing some other tasks related to enhancing security, generating long-term certificates, and so on.
- <u>Cloud Deployment and Scaling</u>. As the ESGF Docker architecture matures, we want to experiment with deploying it on the Cloud, to provide options for other

groups and organizations to re-use it without having to purchase and maintain dedicated hardware and network resources. Cloud studies will include the ability to automatically provision new nodes and scale specific services on demand, in order to provide high performance, load balancing, and fault tolerance.

- Distributed Computing: One of the major goals of next year's DREAM activities is to deploy operational climate computing services at several ESGF nodes, based on the new Docker architecture. We intend to demonstrate execution of several climate algorithms both on datasets located on a single server, as well as coordinate operations across multiple servers. Additionally, we intend to initiate a process of cross-validation of some basic climate algorithms that will be implemented in different languages and frameworks.
- Streaming Data Visualization: The development of the data streaming infrastructure will be enhanced in two main ways. The current prototype will be made more robust and made available via a simple binary distribution with basic analysis templates already created for user testing and modification. At the same time, the query model used for the current LLNL server will be standardized to allow use by more generic clients, through a detailed API based on HTML data queries.
- Climate Data Analysis: We will demonstrate the implementation of two independent user-developed analysis packages from NOAA/GFDL within the ESGF Docker architecture.
- RAML description of search API. We plan to start gaining experience with RAML by using it to describe the current REST API used by ESGF search. This will also provide examples on how to start describing other critical APIs such as those developed by the Computing working team.
- Hydrology: During 2017 we plan to start applying the DREAM framework to the Hydrology domain, by deploying a few DREAM services (in Docker format) to hydrological data and metadata. This will be accomplished by interacting with Hydrology scientists and data managers at JPL with whom our team has a standing collaboration.
- Biology: Similarly, during 2017 we plan to instantiate a test-bed system for biology using the ESGF Docker architecture, which will allow publishing, download and some preliminary analysis of biological datasets.

Product Details

- DREAM project web site: http://dream.llnl.gov/
- DREAM poster for NSG PI meeting: http://esgf.llnl.gov/media/2015-F2F/Posters/DREAM-Distributed-Resources-for-the-ESGF-Advanced-Management.pdf
- ESGF Docker repository: https://hub.docker.com/u/esgfhub/dashboard/
- ESGF GitHub repo for Docker images: https://github.com/ESGF/esgf-docker

- “Embedded Domain-Specific Language and Runtime System for Progressive Spatiotemporal Data Analysis and Visualization”, Cameron Christensen, Shusen Liu, Giorgio Scorzelli, Ji-Woo Lee, Peer-Timo Bremer, Valerio Pascucci, Proceedings of the proceedings of the 6th IEEE Symposium on Large Data Analysis and Visualization, 2016.
- Presentation at 2016 fall meeting of the American Geophysical Union (AGU), Program: Earth and Space Science Informatics, Session: IN13D Advanced Information Systems to Support Climate Projection Data Analysis II (https://agu.confex.com/agu/fm16/meetingapp.cgi/Paper/186194)
- Document on Hydrology Data Challenges (http://dream.llnl.gov/Data/pdf/Hydrology-Data-Challenges-draft.pdf).

Participants and other Collaborating Organizations

- Participant: Ames, Sasha (LLNL)
    - Project role: LLNL Senior Personnel
    - Person months worked: 0.1
    - Funding support (if other than this award): BER ESGF 50%, ACME 30%, LDRD 20%
    - Contribution to the Project: Biology data sets prototype publication and site configuration
    - International Collaboration: N/A
    - International Travel: none in 2016
- Participant: Balaji, V. (Princeton)
    - Project role: Princeton Principal Investigator
    - Person months worked: 1
    - Funding support (if other than this award): NOAA
    - Contribution to the Project: PI, project design and technical oversight
    - International Collaboration: none
    - International Travel: none in 2016
- Participant: Christensen, Cameron (University of Utah)
    - Project role: University of Utah Senior Software Developer
    - Person months worked: 0.5
    - Funding support (if other than this award): DOE, NSF
    - Contribution to the Project: Development of the data streaming infrastructure for data analysis and visualization
    - International Collaboration: ESGF international partners
    - International Travel: none in 2016
- Participant: Cinquini, Luca (JPL)
    - Project role: JPL Principal Investigator
    - Person months worked: 3
    - Funding support (if other than this award): NASA, DOE, NOAA
    - Contribution to the Project: overall system architecture, application to Climate science, REST APIs
    - International Collaboration: ESGF international partners

    - International Travel: none in 2016
- Participant: Chrichton, Dan (JPL)
    - Project role: JPL Co-I
    - Person months worked: 1
    - Funding support (if other than this award): NASA Advanced Information Systems Technology Program, NSF EarthCube, NSF Data Infrastructure Building Blocks
    - Contribution to the Project: Development of the hydrology/big data use case
    - International Collaboration: none
    - International Travel: none
- Participant: Doutriaux, Charles (LLNL)
    - Project role: LLNL Co-I
    - Person months worked: 1
    - Funding support (if other than this award): DOE/ESGF, DOE/ACME, DOE/PCMDI
    - Contribution to the Project: co-leading the ESGF-CWT working group
    - International Collaboration: ESGF, UV-CDAT, ACME national and international partnerships
    - International Travel: none in 2016
- Participant: Duffy, Dan (GSFC)
    - Project role: GSFC Principal Investigator
    - Person months worked: 1
    - Funding support (if other than this award): NASA
    - Contribution to the Project: co-leading the ESGF-CWT working group
    - International Collaboration: none
    - International Travel: none in 2016
- Participant: Ferraro, Robert (JPL)
    - Project role: JPL Co-Investigator
    - Person months worked: 0.0 in 2016
    - Funding support (if other than this award): NASA
    - Contribution to the Project: user community requirements, ESGF Executive Committee, liaison to NASA data community
    - International Collaboration: ESGF international partners
    - International Travel: none in 2016
- Participant: Hill, William (LLNL)
    - Project role: LLNL Senior Software Developer
    - Person months worked: 3
    - Funding support (if other than this award): DOE
    - Contribution to the Project: contributing to the ESGF installation efforts by leading the rewrite of the installation; also working to publish data from disparate science domains
    - International Collaboration: ESGF installation and publication working teams
    - International Travel: 2016 6th Annual ESGF F2F Conference

- Participant: Nikonov, Sergey (Princeton)
    - Project role: Princeton Technical Specialist
    - Person months worked: 7.25
    - Funding support (if other than this award): NOAA
    - Contribution to the Project: data volume estimates for DREAM; participation in ESGF WTs; design and deployment of containerized climate analytics
    - International Collaboration: none
    - International Travel: none in 2016
- Participant: Pascucci, Valerio (University of Utah)
    - Project role: University of Utah Principal Investigator
    - Person months worked: 0.25
    - Funding support (if other than this award): DOE, NSF
    - Contribution to the Project: system architecture of the data streaming infrastructure
    - International Collaboration: ESGF international partners
    - International Travel: none in 2016
- Participant: Williams, Dean (LLNL)
    - Project role: DREAM and LLNL PI
    - Person months worked: 1
    - Funding support (if other than this award): DOE/ESGF, DOE/ACME
    - Contribution to the Project: project coordination, planning and integration with other DOE projects
    - International Collaboration: ESGF, UV-CDAT, ACME national and international partnerships
    - International Travel: none in 2016

Impact

As described in the original proposal, we anticipate that DREAM will have a strong impact in accelerating and enabling new research across multiple scientific domains.

In Climate science, one of the most daunting challenges in recent years has been the exponential increase in the volumes of both model and observational data, compounded by the globally distributed nature of the archives. It is a known fact that any further advances in the field necessitate a paradigm shift with respect to the traditional data analysis workflow: searching for data, downloading data to the scientist's desktop, and executing local data analysis. DREAM has already started working in this direction by defining a server-side computing API that will enable to "bring the computation to the data". In the next years, we will be providing implementations of this API (several, as a matter of fact), as well as start to orchestrate data analysis, data movement and data reduction across multiple data centers. The need for server-side computational capabilities is particularly critical at this time since ESGF is preparing to serve CMIP6 data, which will start to be produced in early 2017, and which will provide the basis for the next assessment report on Climate Change by the IPCC panel, with great political and societal impact across the world.

Another area where DREAM is poised to provide significant impact is related to the ability to quickly install and configure a fully fledged set of services for management of scientific data (in Climate or other disciplines). By lowering the barrier to the installation and maintenance of an ESGF node, and further provide multiple options for system architecture and scalability, DREAM will foster the growth of the current federation to other data centers across the globe, service new data collections and data types, as well as enabling the adoption of this infrastructure in other fields.

Furthermore, the definition of well thought and documented REST APIs for different class of services should carry influence across other software and data projects, showcasing the concept of "SaaS" (Software as a Service) as applied to scientific data analysis. Users will find it much easier to develop clients in the language of their choice to gain programmatic access to the data, which will result in a much faster turnaround in the formulation and verification of a scientific hypothesis.

Problems

Because of several bureaucratic problems related to inter-agency coordination, most of the DREAM institutions did not receive their funding until late in 2016, and therefore could only work for a few months leading to this report. We expect that DREAM development will undergo a very significant increase throughout 2017.