The Earth System Grid Federation:

An Open Infrastructure for Access to Distributed Geospatial Data

Luca Cinquini, Daniel Crichton, Chris Mattmann
Jet Propulsion Laboratory (JPL)
Pasadena (CA), U.S.A.

Gavin M. Bell, Bob Drach, Dean Williams
Lawrence Livermore National Laboratory (LLNL)
Livermore (CA), U.S.A.

John Harney, Galen Shipman, Feiyi Wang
Oak Ridge National Laboratory (ORNL)
Oak Ridge (TN), U.S.A

Philip Kershaw, Stephen Pascoe
STFC Rutherford Appleton Laboratory, NCAS/BADC
Didcot, Oxfordshire, United Kingdom

Rachana Ananthakrishnan, Neill Miller
University of Chicago/Argonne National Laboratory (ANL)
Argonne (IL), U.S.A.

Estanislao Gonzalez
German Climate Computing Center (DKRZ)
Hamburg, Germany

Sebastian Denvil, Mark Morgan
Institut Pierre Simon Laplace (IPSL)
Paris, France

Sandro Fiore
Euro-Mediterranean Center on Climate Change (CMCC)
Lecce, Italy

Zed Pobre
Goddard Space Flight Center (GSFC)
Greenbelt (MD), U.S.A.

Roland Schweitzer
Pacific Marine Environmental Laboratory (PMEL)
Seattle (WA), U.S.A.

The Earth System Grid Federation (ESGF) is a multi-agency, international collaboration that aims at developing the software infrastructure needed to facilitate and empower the study of climate change on a global scale. The ESGF's architecture employs a system of geographically distributed peer nodes, which are independently administered yet united by the adoption of common federation protocols and application programming interfaces (APIs). The cornerstones of its interoperability are the peer-to-peer messaging that is continuously exchanged among all nodes in the federation; a shared architecture and API for search and discovery; and a security infrastructure based on industry standards (OpenID, SSL, GSI and SAML). The ESGF software is developed collaboratively across institutional boundaries and made available to the community as open source. It has now been adopted by multiple Earth science projects and allows access to petabytes of geophysical data, including the entire model output used for the next international assessment report on climate change (IPCC-AR5) and a suite of satellite observations (obs4MIPs) and reanalysis data sets (ANA4MIPs).

Keywords: Earth System Grid Federation (ESGF), peer-to-peer (P2P), Node, security, search, discovery, climate science, CMIP5.

I. Introduction

The study of climate change, including an evaluation of its impact on Earth's ecosystem and human society, is one of the most important scientific challenges of our time. Because the physical processes governing Earth's climate are extremely diverse and complex, this research involves sophisticated model simulations, which generate an unprecedented amount of output data, as well as the collection of observational data from multiple sources (such as remote sensors, in situ probes, and vertical profiles) on a global scale. These data sets are managed and stored at multiple geographic locations across the globe; yet, they need to be discovered, accessed, and analyzed as if they were stored in a single centralized archive.

For over a decade, the collaborative objective has been to establish the Earth System Grid Federation (ESGF) as a leader and visionary architect for all aspects of climate data discovery and knowledge integration. From its formative years (previously known as the Earth System Grid (ESG) [1]), ESGF has demonstrated a viable path forward into the future and attracted key national and international collaborators and potential sponsors recently reaching the critical mass required to undertake fundamental data-intensive and data-driven science problems. In the climate application arena, an increasing number of projects are consolidating on the ESGF peer-to-peer (P2P) infrastructure, a next generation version of the earlier architecture, based on loosely coupled administrative relationships and light-weight protocols for unification. The new architecture allows the federation to play a stewardship role and also enables the world to better organize and integrate all climate knowledge via a cooperative federation. With that said, the ESGF P2P software has distinguished itself from other collaborative knowledge systems in the climate community, as evidenced by its widespread adoption, federation capabilities,

and broad developer base. It is the leading source for current climate data holdings, including the most important and largest data sets in the global-climate community.

Through ESGF, users access, analyze, and visualize data using a globally federated collection of networks, computers, and software. In addition, an enterprise system was adopted to support (inter-)national intercomparison activities of climate models and observations as well as high-profile projects involving such organizations as the Department of Energy (DOE), National Oceanic and Atmospheric Administration (NOAA), National Aeronautics and Space Administration (NASA), and National Science Foundation (NSF).

II. Architecture

ESGF is based on a *distributed* and *federated* software architecture (see Figure 1). The system is composed of multiple sites (called "Nodes") that are geographically distributed around the world, but can interoperate because they have adopted a common set of services, protocols and APIs. Data and metadata are managed and stored independently at each Node; yet, clients can access them as if they were held on a single global archive. Nodes can join or leave the federation dynamically, and the services and holdings that are discoverable and accessible by clients will automatically change to reflect the latest state of the system.

Internally, each ESGF Node is composed of a set of services and applications that collectively enable data and metadata access and user management (see Figure 2). The software stack includes components that were developed directly by ESGF (e.g., the Publishing program, many of the data and metadata web services, and the web portal user interface), as well as open-source, third-party applications that are commonly used in industry (Postgres, Tomcat, Solr), eScience (MyProxy, GridFTP) and the geospatial community (THREDDS, LAS). These components are logically grouped in four areas of functionality, which determine the Node "flavors" (color coded in Figures 1 and 2):

- The Data Node (blue color, Figure 2) includes services for secure data publication and access. Its main components are the data Publisher application that generates the metadata catalogs, the THREDDS and GridFTP servers (with security filters at the front end) to serve the data, and the OpenID Relying Party and Authorization Service to ensure proper authentication and authorization.
- The Index Node (tan color, Figure 2) contains services for indexing and searching metadata, currently implemented using Apache Solr as the back-end server. The Web Portal application provides a convenient browser-based interface for system users and administrators, while the Dashboard contains utilities that monitor and report on the state of the federation. The Common Information Model (CIM) Viewer is a plugin component that retrieves and displays model metadata from CIM repository sources.
- The Identity Provider (light green color, Figure 2) allows user authentication and secure delivery of user attributes. It includes an OpenID Provider for browser-based authentication, a MyProxy server for requesting limited-lifetime user certificates, and Registration and Attribute Services for distributed access control.
- The Compute Node (purple color, Figure 2) contains higher-level services for data analysis and visualization. Currently its only components are the Live Access Server, the Ferret-THREDDS Data Server and the Ferret engine. More analysis engines (e.g., Ultra-scale Visualization Climate Data Analysis Tools (UV-CDAT)) are to be added in the near future.

Additionally, each of these flavors always includes the Node Manager, a web application that allows a Node to continually exchange service and state information with all its peers throughout the federation. Each Node can be configured during installation to have one, more, or all of the flavors, depending on the site specific needs. Furthermore, a site can combine multiple Nodes configured for different flavors, to

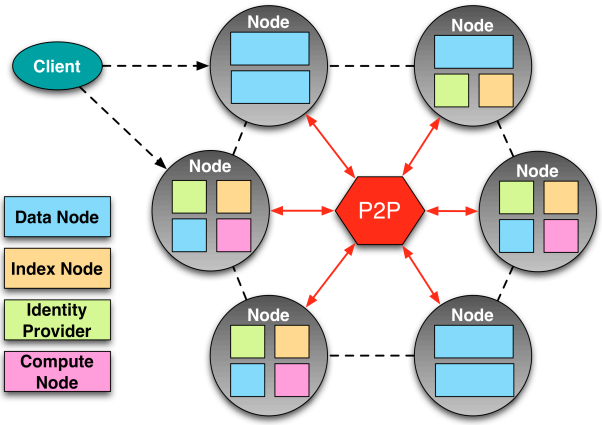

Figure 1. Schematic representation of the Earth System Grid Federation (ESGF) as a system of Nodes hosting services of four possible "flavors". Nodes continually exchange messages with each other through the peer-to-peer (P2P) protocol (here pictured as the central core of the system). A client can query any Node to access data and metadata throughout the federation.

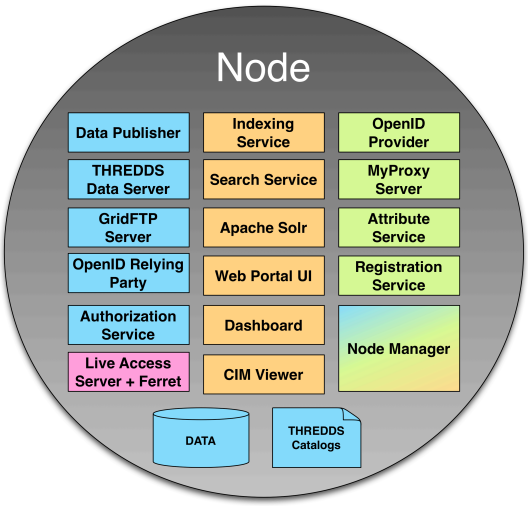

Figure 2. Software components that may be deployed onto an ESGF Node, grouped by "flavor" or area of functionality.

achieve higher performance and availability. For example, a common practice is to deploy the Index Node and Identity Provider services on one server, while keeping the Data and Compute Node services on one or more separate servers, e.g., for performance considerations.

ESGF Nodes interoperate through a set of common federation services and mutual trust relationships. A user who registers at any site can authenticate (via OpenID or a short-lived user certificate) across the entire federation because the other Nodes trust the original Identity Provider. A metadata search initiated at any Node will be distributed to all the other Nodes, and the search results will be merged before they are presented to the user. When data is accessed from a specific Node, the local Authorization Service will query the appropriate Attribute Services across the federation, to determine whether the user has been granted the necessary permissions to satisfy the local access control policies. The list of trusted Identity Providers, their Certificate Authorities, the target search Indexes, and the location of the Attribute Services are continually updated at each Node through the peer-to-peer messaging executed by the Node Manager. The net result of this setup is that a client (web browser or desktop application) can query any Node to investigate and access data and metadata holdings distributed throughout the federation.

III. The ESGF Software Stack

ESGF software components are developed collaboratively by many institutions worldwide, which are funded by different agencies and programs. For the U.S., the DOE Office of Science is the lead funding agency. The software is made freely available on the ESGF *github* web site (http://esgf.org/gitweb, soon to be replicated to https://github.com/ESGF) under the Open Source BSD license, and it is documented for users, administrators and developers on the ESGF collaboration site (http://www.esgf.org/). The general development philosophy is to build modular components that interact with each other and with outside clients through well-defined interfaces, so that different implementations could be plugged into the system, possibly written in different computing languages (Java, Python, etc.). When possible, ESGF uses existing standards from industry (e.g., OpenID, SSL, SAML) or the geospatial community (e.g., NetCDF, CF, OPeNDAP) to maximize interoperability with other systems. Popular, freely available application servers such as Tomcat, Solr, THREDDS and LAS are integrated into the stack to avoid re-implementing a specific piece of functionality.

In this section, we describe in more detail the major components of the ESGF software stack.

A. The Node Manager

The Node Manager is the primary coordinating software component that manages the local ESGF Node and maintains the federation's cohesion. It is an event-driven system composed of 'components' that comprise a SEDA [2] inspired *finite-state machine* (FSM). Events are routed through an arrangement of components and are transformed as they travese the FSM from start state (ingress) to end state (egress). The main components of the Node Manager implement the discovery and information dissemination protocol that keeps the federation connected and they provide the necessary routing information used by higher-level Node services. The Node Manager is also responsible for collecting current system *health* information - i.e., system metrics - and monitoring information such as cpu, memory, and up-time information as well as quantitative information on a local Node's data holdings. The Node Manager implements *intra*-node message passing through its internal components and enables *inter*-node message passing by way of its peer-to-peer protocol implementation.

B. The Peer-To-Peer Protocol

A key feature of ESGF is the ability for Nodes to *discover* each other and therefore create a single federated *dataspace* [3] that spans all participating ESGF Nodes. The resultant overlay mesh network created by the ESGF protocol allows users to interact with this singular dataspace to seamlessly query and access data. At its core, the protocol is a *gossip* protocol [4] with two primary purposes: as the *discovery mechanism* allowing Nodes to learn about each other in an unsupervised way, and as the *anti-entropy* [4] protocol that maintains a distributed 'database' of currently federated, participating nodes. The implementation of the gossip protocol in ESGF follows the *push* model [4]. Each Node views the federation as a list of other Nodes it is authorized to access, which is essentially the 'database' maintained via the anti-entropy protocol. This list is encapsulated and sent as payload within a *remote event* that is forwarded to peer Nodes. Nodes receive remote events and inspect the payload information. The payload is then integrated with the receiving Node's internal list, and that new list is pushed randomly to two other peer Nodes. Nodes propagate state information directly after receiving an ingress remote event. Each Node initially creates a list with a single list entry containing its own time-stamped information. As ingress events are encountered, the introduced list is merged with the local list like a vector clock [5], maximally by time.

A gossip protocol was chosen because of its simplicity, scalability, robustness and reliability "*even in the most demanding settings*" [6]. ESGF leverages the gossip protocol's fault tolerance to create a fully connected overlay mesh network of Nodes with no single point of failure. The federation may be partitioned into *groups*, and a Node may participate in any number of groups. Each group must contain at least one Identity Provider Node to service the security needs of the group's users. Each Node is bootstrapped with a prior knowledge of at least one *default peer*. Nodes maintain an eventually consistent data store of all the Nodes in the union of the peer groups they are members of.

Several protocol optimizations are being studied, including tuning fanout [7] and frequency and taking advantage of a push-pull interaction between Nodes. Our ultimate goal is to optimally decrease database convergence time and network overhead..

C. Data Preparation and Publishing

The Publisher application is a Python package that is responsible for scanning data stored on a Data Node and making it available through the system. The program, which can be run from the command line or from a graphical user interface (GUI), can extract metadata from both the directory structure and filenames, and from the content of the files

themselves. Files are grouped in logical collections named *data sets* according to project-specific rules in the relevant configuration file. All information is stored in a local Postgres database and is used to generate THREDDS/XML catalogs. These catalogs contain the descriptive metadata and the URL pointers to access the data through all available protocols (e.g., HTTP, GridFTP). The THREDDS Data Server (TDS) and Live Access Server (LAS) use these catalogs to deliver data to clients for download, analysis, and visualization.

The Publisher contains prebuilt *handlers* for processing data that are stored in NetCDF format and follow the Climate and Forecast (CF) convention, a prescription for encoding richer metadata. Most of the projects that ESGF has focused on to-date mandate that data be stored as NetCDF/CF (e.g., CMIP5, obs4MIPs, ANA4MIPs). Nevertheless, the Publisher package is extensible, and custom handlers can be plugged-in to parse data that are stored in other formats or are organized on disk according to other specific directory structures.

D. Search Services

The ESGF search module, deployed as part of an ESGF Index Node, is responsible for making the data holdings of one or more ESGF Data Nodes discoverable and accessible by clients. Its two main features are the ability to execute distributed searches across the federation, and the support for a well-defined REpresentational State Transfer (REST) [8] API that insulates clients from the specific query syntax of the back-end metadata repository. Although the software allows for pluggable back-ends, it currently uses Apache Solr as the underlying search engine. Solr is a popular web application which is used in many commercial web sites, featuring high-performance text and faceted searching, geospatial and temporal querying, and partition of searchable metadata across multiple local indexes (*cores*) and distributed servers (*shards*).

The search module makes two high-level web services available to clients:

- The Indexing Service parses the metadata content available at some repository (located by its URL), and ingests it in the back-end metadata storage. To-date ESGF has focused primarily on parsing metadata from THREDDS catalogs, but support for other repositories such as GCMD archives is forthcoming. Indexing operations are subject to authorization according to the local policies established at the Index Node: the client must present a valid certificate belonging to a user that has been granted publishing rights for that specific data collection.
- The Search Service queries the index metadata content and retrieves matching results that include descriptive information as well as all the available data access points (e.g., HTTP, GridFTP, OPeNDAP, and LAS). The search service is invoked by clients through a REST API that includes the ability to execute free-text queries (e.g. "climate", or "surface temperature"), and to search for specific values of one or more search categories (e.g. "model = CESM and time frequency = monthly"). Among its many features, the API allows clients to search for either the latest or a specific version of the data as well as for its replicas, plus it can target the local index only or any number of remote indexes. Response documents can be returned in either XML or JSON format.

Internally, the search module partitions its metadata space into different object types, which are stored and queried separately – a strategy we employed to optimize performance. *Data sets* are the unit of discovery in the system, since they contain high-level information about a collection (such as the model that generated the data, the instrument that collected it, or the collection's temporal and geospatial coverage). As such, data sets are typically returned as results matching a query. *Files* and *aggregations* represent the content of data sets, and contain the URL endpoints of all services that clients can use to access the data. Typically, a single data set contains many files and/or aggregations, so searching for datasets is much faster.

Within Solr, each object type is stored in a dedicated "core" or index, which is separately updated and queried. Additionally, the Solr setup at each Node is such that metadata is indexed into a *master* Solr instance, which is only accessible locally. The master is replicated at one-minute intervals to a *slave* Solr instance, which is accessible in a read-only fashion.

E. Security Infrastructure

The ESGF security infrastructure [9] restricts resource access to users who have been granted the necessary permissions and it manages trust between participating parties across the federation. Permissions can be set at various levels of granularity, for example, authorizing a specific user to publish metadata to an Index Node, download specific data sets from a Data Node, or manage user permissions through a Node administrative web interface. A typical use case might limit access to a specific data collection to only those users who have subscribed to a license agreement (such as the CMIP5 Research or Commercial licenses). Other collections might be limited to a particular research group and not intended for public consumption. ESGF security allows Node administrators to maintain complete control over the local resources, while enabling federation-wide services such as single sign-on and distributed access control. This security framework is accomplished through a set of trusted Identity Providers and Certificate Authorities.

ESGF enables users and applications to authenticate through two different mechanisms. The *OpenID* protocol is used by humans accessing the system through a web browser. When users enter their OpenID at any Node in the federation, they are redirected to their Identity Provider (the Node where they registered), where they can sign-in using the mechanism enforced by that Provider (for example, username and password). Once authenticated, the user is redirected back to the original Node. *X.509 certificates* are used by programs (such as *wget* scripts or GridFTP) to act on behalf of a user. Before executing the program, the user requests a short-lifetime certificate from a MyProxy server that is linked to the user's Identity Provider, and thus to the user's OpenID identity. The certificate contains the user's identity and permissions, and it will be honored by all Nodes in the federation that trust the Identity Provider.

ESGF authorization is based on *security policies*, defined by each Node administrator, that match local to the groups and roles that are allowed to execute specific operations. Users from any site can request membership in one or more access groups (e.g., the "CMIP5 Research" or "NASA OBS" group), and are assigned one or more roles within each group (e.g., "user", "admin", or "super"). Groups may be managed by any Identity Provider in the federation, and the user attributes are made available to trusted clients through the Node's Attribute Service. When authorization is required, the local Authorization Service enforces the Node security policies by querying the particular Attribute Service in the federation that manages the configured access control group. For example, any Data Node participating in CMIP5 allows users to access the data if they have enrolled in the group "CMIP5 Research" that is administered by the Lawrence Livermore National Laboratory (LLNL) Identity Provider.

Because ESGF security is based on standards (OpenID, SSL, SAML), the security components can interoperate with trusted components from other software stacks that are based on the same standards. For example, a Java-based ESGF Authorization Service can request and consume user attributes from a remote Attribute Service part of a Python software stack, because the entire communication is encoded as SAML/XML documents that have been digitally signed. Similarly, users can authenticate throughout the federation via their OpenID, no matter what the underlying implementations of the OpenID client and server are.

F. Web Portal User Interface

A key goal of the current ESGF development effort is to provide an intuitive, easy-to-use web portal. To this end, ESGF has adopted a modern design inspired by emerging Web 2.0 technologies. The portal makes heavy use of the state-of-the-art JQuery JavaScript libraries and Model-View-Controller (MVC) design patterns. The user interface (UI) also conforms to the overall ESGF philosophy of decoupled software components that accommodate a more dynamic user experience. Services of other components (e.g. search and node manager components) may be utilized via Ajax-style HTTP calls. The web UI materializes a variety of ESGF-supported services through the browser: administrative and user account management interfaces, collection and file-level search and discovery, the dashboard service, the LAS visualization engine, and the CIM viewer. The first two features are briefly described in this section, the others later on in this paper.

The *user account and administrative component* delivers a clear, concise means for users and administrators to manage personal information and security aspects of the portal. Users may view and edit personal attributes and apply for group memberships that grant access to restricted data. Administrators are provided with tools that interface to the security infrastructure services allowing them to authorize group access to specific data holdings.

Search and discovery mechanisms are presented to the user via the *search component* of the web portal (see Figure 3). The search component is presented as a single composite page, which eliminates the clumsy "back-and-forth" navigation offered by static pages. Initially, a user may search the data holdings at the *collection* level. Here, the user has many search options. The sidebar widget on the left side of the page presents a configurable list of *facets*, which allow users to constrain their search results through categorization. Search constraints can also be added via the free text search bar. Furthermore, users may take advantage of the temporal and geospatial (based on Google Maps) search overlay widgets by searching for data measured or simulated at specific times or in specific geographical locations, respectively. The *state* of the search (i.e. the current constraints applied by the user) is shown prominently in the upper left corner of the page and may be changed at any time. Important information about the results is viewed through the portal search component.

Each data set contains a comprehensive metadata summary outlining descriptive information about the data set. The CIM viewer, which provides richer and more detailed metadata about climate models, may also be invoked. Additional information associated with a data set, such as a technical report, may also be accessed. Finally, the LAS browser visualization tool may be invoked to visualize details of the data sets. For file access, the search component uses a "shopping cart" approach commonly deployed by e-commerce sites. The user needs only to select a collection-level data set (via the "Add to Cart" link) and navigate to the "Data Cart" tab. The collections previously chosen are displayed in this tab and may be quickly expanded to their *file*-level data holdings. These files may then be downloaded using different methods, which are discussed in the next section.

G. Data Access

By default, every Data Node is configured to serve data through the following access types:

- Direct browser access is provided by the TDS and allows users to download files via simple HTTP hyperlinks.
- OPeNDAP access is also provided by the TDS. The Open Data Access Protocol provides an abstraction level on top of the files allowing users to tailor data selection by

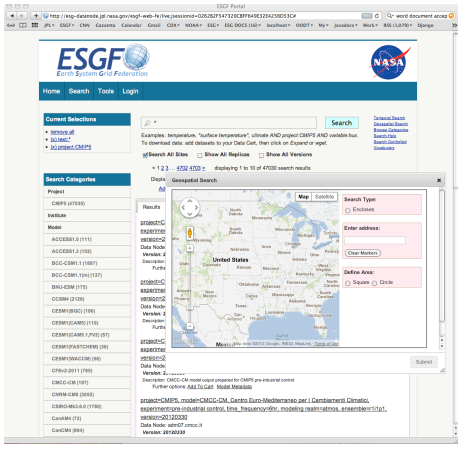

Figure 3. The ESGF web portal exposes a friendly user interface for searching, including the capability to specify free text, facet options, and geospatial and temporal constraintss.

variable, temporal or geospatial coverage. ESGF provides aggregations for variables that are normally split in files of no more than 2 GB to simplify access.

- Wget Scripts, contrary to other access mechanisms, are not tied to the Data Node hosting the data, but to an Index Node implementing the search API, which was extended to return a BASH script that simplifies data retrieval. The user can thus access any file in the federation without knowing where the file is stored or even what it is named. The script encapsulates calls to the *wget* utility—hence the name—and adds functionality such as: certificate verification, retrieval and renewal; download continuation, verification via MD5 checksums and verification results caching; download structure generation via the facets results; file overflow check (if the query resulted in a set of files larger than the maximum number allowed per script); and script auto-update and remote file changes check.
- GridFTP access uses a special protocol based on FTP to allow for a high-performance, securly authenticated, and reliable data transfer. Within ESGF, the most common use of GridFTP is for data replication, where bulk data is replicated across Nodes. This feature requires higher access permissions for the user, including write access to the ESGF Data Node, and typically is restricted to administrators.

Furthermore, a Data Node can be configured to allow other optional data access methods:

- Globus Online [10] is a hosted service that provides reliable and managed file-transfer capabilities over GridFTP. The web portal component includes a preliminary integration with Globus Online. A user can choose to have Globus Online either transfer the items in their data cart to another server or download them to their local machines. Users must install Globus Connect on their machines to upload and download data locally.
- The Live Access Server (LAS) is a highly configurable, freely available open source web application that allows users to visualize and compare data on-the-fly, perform basic analysis, and download results in various file formats. New capabilities were implemented for reading THREDDS catalogs, managing the data selection UI for large collections, and generating distributed products to ensure a seamless and efficient integration of LAS into the ESGF software stack. In general, LAS can create the configuration it needs directly from THREDDS catalogs and the underlying NetCDF data sources. However, within ESGF, LAS was enhanced to build its configuration from the enhanced metadata in the THREDDS catalogs that are generated by the Publisher application. Search interface tools allow the user to quickly identify and select the data sets of interest. Once a set is selected and the user has transitioned to LAS, the LAS UI limits the available data sets to those preselected in the browsing session from the ESGF search interface. In this way, the user is not overwhelmed with tens of thousands of data sets while performing advanced visualization and analysis on the data of interest.

H. Dashboard

The Dashboard is the distributed monitoring system of ESGF. It is responsible for collecting historical information about the status of the federation in terms of:

- *Network topology* (peer-groups composition)
- *Node type* (host/services mapping)
- *Registered users* (including their Identity Providers)
- *Downloaded data* (both at the Node and federation level)
- *System metrics* (e.g., round-trip time, service availability, CPU, memory, disk, processes, etc.)

From an architectural viewpoint, the ESGF Dashboard is composed of the following three parts: the information provider, the dashboard catalog, and the user interface.

The *information provider* represents the system's back-end and is responsible for retrieving and storing all peer-to-peer metrics. The information provider strongly interacts with the node manager to synchronously retrieve updated snapshots of the federation's status and stores them in the dashboard catalog. This module can easily be extended to collect new metrics, by implementing additional sensors associated to new classes of information.

The *dashboard catalog* is a temporal database (running on top of a Postgresql RDBMS) used by the information provider to persistently store the five classes of information previously mentioned. It manages historical information with a *per-site* and *per-metric* configurable time-window.

Finally, the *user interface* (see Figure 4) is a web application that exploits the MVC design pattern. It relies on strong adoption and implementation of Web 2.0 concepts such as mash-up, Google maps and permalinks. It provides several views at four different (hierarchical) granularity levels: federation, peer-group, host and service. These views provide charts and summaries about the service availability, the round trip time between pairs of nodes, the host status, and other information. The views, which can be customized (at the presentation level) by the end user, are delivered in different ways (charts, tables) and formats (XML, CSV, JSON), and can be easily exported (both with and without authentication) in other web-applications, by exploiting the permalink feature implemented into the Dashboard module.

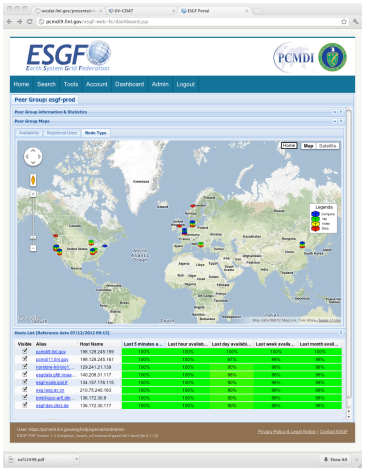

Figure 5. The dashboard screen indicating location, composition and status of all nodes in the "esgf-prod" peer group.

I. CIM Viewer

The Metafor Common Information Model (CIM) [11][12] is an ontology designed to become the standard for climate modeling related metadata. The CIM software eco-system is composed of several applications and services that collectively support a wide range of functionality: the ontology itself; validation; search; dissemination; and integration with other platforms such as the ESGF web front-end. For example, the CMIP5 questionnaire is a web based interactive tool for creating CIM documents describing climate models that contribute to CMIP5, and captures information about how those climate models were set up to run simulations for CMIP5 experiments. The CIM web services backend was implemented according to the Service Orientated Architecture (SOA) paradigm using the Python language, and following a REST architectural design. The web services support a diverse array of CIM resources, including documents, validation reports, queries, identifiers, and controlled vocabularies. The lightweight Javascript CIM Document Plugin Viewer is a platform independent web-based client used to display CIM documents from within a generic portal environment, such as for example the ESGF web portal. Currently it provides the basic capability to display one CIM document at a time. It will be extended with more advanced capabilities to take information from multiple CIM documents and merge them together into a cohesive and meaningful presentation.

IV. Application and Operation

From the beginning, the primary focus of ESGF has been to provide the IT infrastructure in suport of CMIP5, the Climate Model Intercomparison Project, phase 5. CMIP5 is a global effort coordinated by the World Climate Research Programme (WCRP) that involves tens of modeling groups spanning 19 countries. CMIP5 designed several decadal and century experiments that all models run to assess the future state of the Earth's climate and ecosystem and to understand the complex processes in play during simulated changing climate. These experiments include economic and human development scenarios, Green House Gases emissions and/or concentrations, idealized experiments, and quantifying the predictability of the climate system at the decadal scale. When completed, the CMIP5 data archive will total upwards of 3.5 petabytes, stored across 15 Nodes on four continents (North America, Europe, Asia, and Australia). In order to facilitate access and to secure the long-term preservation of this invaluable data store, ESGF plans to replicate 1.5 – 2 petabytes of data at five major data centers: LLNL (U.S.), BADC (U.K), DKRZ (Germany), ANU (Australia), and the University of Tokyo (Japan).

Alongside the CMIP5 model output, ESGF provides access to the obs4MIPs data sets. These are satellite observations from several NASA missions that have been purposedly formatted to allow easy comparison with models: they have been converted from the original HDF/EOS format to NetCDF/CF, averaged as monthly means, and regridded over global grids from the original satellite swaths. In addition, the system provides access to scientific Technical Notes, authored by the data processing engineering teams in collaboration with the mission scientists. These Technical Notes contain invaluable information on how to use the observations for comparing and validating models.

obs4MIPs also includes the CFMIP-OBS data set, which contains observational data and diagnostics about clouds and cloud-related processes. These data can be of great value for climate prediction over a wide-range of space and time scales: for numerical weather predictions, for climate prediction at seasonal to decadal timescales, and for the prediction of the long-term climate response to anthropogenic perturbations at the global and regional scales. This stems from the strong interaction of clouds with the global Earth's radiation balance, the local energy balance, the atmospheric circulation and the hydrological cycle.

Another such effort in support of CMIP5 is ANA4MIPs, which allows users to find and download reanalysis data via ESGF. Through reanalysis climate scientists are continually updating gridded data sets that represent the state of the Earth's atmosphere and incorporating weather observations and numerical output of weather predictions. To enhance climate science resources, five participating reanalysis institutions have agreed to host the following primary reanalysis data sets on ESGF Nodes: (1) MERRA (NASA/GSFC), (2) CFSR (NOAA/NCEP), (3) Interim (ECMWF), (4) 25 (JMA), and (5) 20CR (NOAA/ESRL). These reanalysis data will be translatted to the NetCDF/CF format for greater CMIP5 compatability.

Additionally, CMIP5 has become the benchmark standard that other similiar Earth science projects have leveraged in order to make their data accessible to users under a distributed and integrated infrastructure. Some of the most prominent examples are:

- CORDEX: The COordinated Regional climate Downscaling EXperiment is a WCRP-sponsored program aimed at producing an improved projection of regional climate change worldwide for input into impact and adaptation studies. CORDEX will use the multiple-forcings general circulation models (GCMs) from the CMIP5 archive to generate an ensemble of dynamic and statistical downscaling models. The amount of data managed by the system is expected to be comparable to that of CMIP5.
- TAMIP: Transposed-AMIP is a WCRP endorsed activity to run climate models in weather-forecast mode. This activity will primarily focus on analyzing the accuracy of climate models by comparing hindcasts run by those models with observational data.
- GeoMIP: The Geoengineering Model Intercomparison Project is a WCRP-endorsed project aimed at analyzing the effects of manipulating an environmental processes by implementing one or more large-scale geoengineering technologies.
- PMIP: The Paleoclimate Model Intercomparison Project is a WCRP endorsed project designed to evaluate how accurately climate models reproduce climate states that are radically different from those of today. The third phase of this project (PMIP3) shared some experiments with CMIP5 but proposes an additional set of paleo-climate experiments.
- LUCID: The Land-Use and Climate Identification of robust impacts project explores how robust hanges in land cover—that is, those above the noise generated by model variability—affect model output.

All these projects rely, at least partially, on the CMIP5 archive. By providing their results over the same platform they guarantee that users can find, use, and share their data in a standardized manner, thus minimizing the amount of time employed in non-core activities.

Other non-CMIP5 projects that leverage ESGF include:

- CSSEF: Sponsored by DOE's Office of Science, the Climate Science for a Sustainable Energy Future project is developing a new climate-modeling methodology to reduce uncertainty through the provision of a test bed.
- CMDS: The Coastal Marine Discovery Service is a NASA-sponsored activity aimed at providing Global Information System visualization capabilities for prominent OPeNDAP-accessible oceanic data sets.

V. Future Directions

Many challenges remain for climate scientists, and new ones will emerge as more and more data pour in from climate simulations run on ever-increasing high-performance computing (HPC) platforms and from increasingly powerful, higher-resolution satellites and instruments. From these current and future challenges, ESGF envisions a scientific data universe in which data creation, collection, documentation, analysis, preservation, and dissemination are expanded not just to benefit the geoscience community, but also for other scientific domains, such as biology, chemistry, energy, fusion, and materials. By combining with other disciplines, ESGF seeks to strengthen the entire scientific community with advances that will be beneficial for all. In addition to our trademark advances in data preparation, search and discovery, data access, security, and federation—as mentioned in this paper—we will focus our efforts on expanding into the new areas of data mining, provenance and metadata, workflows, HPC, data movement, and data ontology. Furthermore, we intend to fully explore the possibility of providing a configurable and scalable ESGF environmemt that can be easily deployed on the Cloud, so that resources and services can be instantiated on demand for specific short-lifetime projects, or to meet other requirements such as high availability, extended storage, and elastic allocation of computing processes. As future computing platforms, satellites, and instruments expand and reach extraordinary speeds and levels, we anticipate that in the coming decade ESGF software will migrate from managing petabytes to zettabytes of federated data. To prepare for such an expansion—with continual delivery of a comprehensive end-to-end solution for the overall increase in scientific productivity—ESGF is also partnering with private industry in an effort to make substantial investments in data-driven software and technologies paramount for future computing platforms and ultrascale data archives.

Acknowledgments

The development and operation of ESGF is supported by the efforts of principal investigators, software engineers, data managers and system administrators from many agencies and institutions worldwide. Primary contributors include ANL, ANU, BADC, CMCC, DKRZ, ESRL, GFDL, GSFC, JPL, IPSL, NCAR, ORNL, LLNL (leading insitution), PMEL, PNNL and SNL. Major funding provided by the U.S. Department of Energy, the National Atmospheric and Space Administration (NASA), and the European Infrastructure for the European Network for Earth System Modeling (IS-ENES).

References

[1] Williams, D.N., Ananthakrishnan, R., Bernholdt, D.E., Bharathi, S., Brown, D., Chen, M., Chervenak, A.L., Cinquini, L., Drach, R., Foster, I.T., Fox, P., Fraser, D., Garcia, J., Hankin, S., Jones, P., Middleton, D.E., Schwidder, J., Schweitzer, R., Schuler, R., Shoshani, A., Siebenlist, F., Sim, A., Strand, W.G., Su, M., and Wilhelmi, N., "The Earth System Grid: Enabling Access to Multi-Model Climate Simulation Data", in *Bulletin of the American Meteorological Society*, 90(2): pp.195-205, 2009.

[2] M. Welsh, D. Culler, and E. Brewer, "SEDA: an architecture for well-conditioned, scalable internet services", in *Proceedings of the eighteenth ACM symposium on Operating systems principles (SOSP '01)*, ACM, New York, NY, USA, pp.230-243.

[3] Michael Franklin, Alon Halevy, and David Maier, "From databases to dataspaces: a new abstraction for information management", SIGMOD Rec. 34, 4, pp.27-33, December 2005.

[4] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance", in *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing (PODC '87)*, ACM, New York, NY, USA, pp.1-12.

[5] L. Lamport, "Time clocks and the ordering of events in a distributed system", *ACM Communications*, 21(7), pp.558-565, 1978.

[6] W. Vogels, R. van Renesse, and K. Birman, "The power of epidemics: robust communication for large-scale distributed systems", in *Proceedings of HotNets-I*, Princeton, NJ, 2002.

[7] S. Verma and W. T. Ooi, "Controlling Gossip Protocol Infection Pattern Using Adaptive Fanout", in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005)*, pp.665-674, June 2005.

[8] Fielding, Roy Thomas, "Architectural Styles and the Design of Network-based Software Architectures*",* Doctoral dissertation, University of California, Irvine, 2000.

[9] Philip Kershaw, Rachana Ananthakrishnan, Luca Cinquini, Dennis Heimbigner, and Bryan Lawrence, "A Modular Access Control Architecture for the Earth System Grid Federation", in procedings of *The 2011 International Conference on Grid Computing and Applications*, July 2011.

[10] Allen, B., Bresnahan, J., Childers, L., Foster, I., Kandaswamy, G., Kettimuthu, R., Kordas, J., Link, M., Martin, S., Pickett, K. and Tuecke, S., "Software as a Service for Data Scientists", in *Communications of the ACM*, 55(2): pp.81-88, 2012.

[11] Eric Guilyardi, V. Balaji, Sarah Callaghan, Cecelia DeLuca, Gerry Devine, Sebastien Denvil, Rupert Ford, Charlotte Pascoe, Michael Lautenschlager, Bryan Lawrence, Lois Steenman-Clark, Sophie Valcke, "The CMIP5 model and simulation documentation: a new standard for climate modeling metadata", in *CLIVAR Exchanges*, 16(2), pp. 42-46, 2011.

[12] Lawrence, B. N., Balaji, V., Bentley, P., Callaghan, S., DeLuca, C., Denvil, S., Devine, G., Elkington, M., Ford, R. W., Guilyardi, E., Lautenschlager, M., Morgan, M., Moine, M.-P., Murphy, S., Pascoe, C., Ramthun, H., Slavin, P., Steenman-Clark, L., Toussaint, F., Treshansky, A., and Valcke, S., "Describing Earth System Simulations with the Metafor CIM", Geosci. Model Dev. Discuss., 5, 1669-1689, doi:10.5194/gmdd-5-1669-2012, 2012.

[13] Crichton, D.J., C.A. Mattmann, L. Cinquini, A. Braverman, D.E. Waliser, M. Gunson, A. Hart, C. Goodale, P.W. Lean, and J. Kim, "Software and Architecture for Sharing Satellite Observations with the Climate Modeling Community", in *IEEE Software,* in press, 2012.