Developing an integrated end-to-end TeraGrid climate modeling environment

Lan Zhao, Carol X. Song, Christopher S. Thompson, Han Zhang, and Madhavan Lakshminarayanan
Purdue University, West Lafayette, IN
lanzhao@purdue.edu

Cecelia DeLuca and Sylvia Murphy
NOAA/CIRES, Boulder, CO
cecelia.deluca@noaa.gov

Kathy Saint
Silicon Graphics International, Fremont, CA
ksaint@sgi.com

Don Middleton, Nathan Wilhelmi, Eric Nienhouse, and Michael Burek
NCAR, Boulder, CO
don@ucar.edu

ABSTRACT
The Community Earth System Model (CESM) is a widely used community model for studying the climate system on the Earth. The CESM model is both data and computationally intensive, making it is difficult for users to set up and run CESM simulations using local resources. In this paper, we describe an integrated climate modeling environment that supports CESM simulations on the TeraGrid, comprehensive model metadata description, and automatic archival of model data and metadata for easy community access. This system builds upon and integrates several existing efforts – the Purdue CCSM modeling portal, the Earth System Grid, the Earth System Modeling Framework, and the Earth System Curator. We present the design and implementation of our prototype system as well as an end-to-end usage scenario which is broken down into three workflows: model execution, data publishing, and metadata collection/publishing. The system will be used to support research and education on climate systems. We describe our plan and early efforts to engage users and obtain their feedback.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Online Information Services – *Data sharing, Web-based services.* H.5.2 [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces – *User-centered design.*

General Terms
Design, Reliability, Performance.

Keywords
CESM, CCSM, climate model, Earth System Grid, ESMF, metadata, Purdue CCSM portal, science gateway, TeraGrid.

TeraGrid'11, July 18–21, 2011, Salt Lake City, Utah, USA.

1. INTRODUCTION
The Community Earth System Model (CESM) is a fully-coupled advanced global model for simulating the climate system on the Earth [1]. It is one of the most widely used climate simulation tools worldwide. It is also one of the climate models used for the Intergovernmental Panel on Climate Change (IPCC) assessment reports. The open source CESM software is developed by a community that includes participants from the National Center for Atmospheric Research (NCAR), DOE laboratories, and several universities. CESM 1.0 was first released in June 2010. It replaces the Community Climate System Model (CCSM), whose last release was CCSM4 in April 2010. The CCSM4 code introduced a more flexible component infrastructure and approach to configuration. As a result, users can set up simulations based on a larger combination of component sets and at a much higher data resolution. It enables researchers to investigate problems that they could not address previously. In addition, CESM 1.0 introduced many updates to individual science components, and a new land-ice model.

CESM model consists of five separate components simulating the Earth's atmosphere, ocean, land, land-ice, and sea-ice systems. A central coupler component is responsible for inter-component data transformations and transfers. The CESM model is data and computationally intensive, requiring high performance computing (HPC) resources and significant data storage for large scale experiments. Typical throughput for running CESM 1.0 on 512 processors of the IBM Power6 computing cluster "bluefire" at NCAR is 14.40 simulation years per day [2]. Because of the complexity of the model software, it is challenging and time-consuming for researchers to port, install, configure, and validate the full system as well as to archive the results. It presents a significant challenge for educators to train students in using the model, and for the broader research community to utilize the model in their research.

As an initial effort to tackle the above challenges, the Purdue TeraGrid resource provider developed a science gateway that allows users to run CCSM model simulations on the TeraGrid, and post process and visualize the results via a web browser. The gateway sets up a CCSM model on the TeraGrid Steele cluster and uses a shared community account to submit jobs via Globus.

It shields scientists from the many tasks involved in setting up and running CCSM on the TeraGrid, e.g. compiling CCSM on a specific computing system, validating the model installation, and acquiring individual accounts and time allocation on the TeraGrid. The gateway has been successfully used to support classes and research activities [3].

There are also major challenges in building a collaborative environment for climate simulations which enables scientists to understand and compare model runs with different versions and configurations, and to compare CESM simulations with the output of other climate models. Generation of provenance metadata that captures the components, versions, and configuration settings used, along with experiment and system information, is important for data reproducibility, result interpretation, and problem diagnosis. To date, the climate modeling community lacks an environment that can generates model metadata in standard formats, archives it along with the model outputs, and makes it available for scientists to discover and access. It is the objective of this project to provide this type of integrated workflow by utilizing TeraGrid resources and multi-institution efforts.

There are several well-known projects that aim to facilitate climate data/metadata management and increase model interoperability: Earth System Grid (ESG), Earth System Curator, the Earth System Modeling Framework (ESMF), and the EU-based METAFOR effort [5,6,7,8]. These projects support distinct parts of the end-to-end modeling workflow.

With an international community of over 20,000 registered users and over 500 TB of data collections, and a total download volume of over a petabyte, ESG is a key piece of the infrastructure needed for climate model comparisons. It provides a data and knowledge management system and distribution portal which stores the model output for a variety of intercomparison projects, including those used for IPCC. The “gateway” portion of ESG, which includes the user interface and underlying infrastructure for accessing data, was developed at NCAR.

The Earth System Curator project collaborated with the ESG team at NCAR to extend the ontology that underlies the ESG gateway, and to build tools for collective analysis and comparison. The result, the ESG-Curator (ESGC) portal, includes “trackback” pages that retrieve a detailed description of the model associated with a particular dataset, dynamic comparison tables for selected components and properties, and faceted search and browse through models, components, simulations and datasets. This trackback capability has been absorbed into the ESG production gateway and is being used to display metadata for the next IPCC assessment.

The Earth System Modeling Framework (ESMF) is infrastructure for building and coupling model components in climate, weather, coastal, and other applications [7]. To address the difficulties in collecting and standardizing metadata, the ESMF team developed an Attribute class to capture and write out standard model metadata in XML and other formats. ESMF Attributes are being integrated into the CESM model. Members of the Curator project developed a tool that ingests that XML into the ESGC portal and populates the database and user interface.

To build upon and integrate the synergistic community efforts in the Purdue CCSM portal, ESG, ESMF, and ESGC, a new collaboration was formed between Purdue RP and the ESG/C teams at NCAR and NOAA. The primary goal is to create a seamless environment for end-to-end CESM data and model workflows. Our long term vision for this project is a semantically enabled environment that includes modeling, simulated and observed data holdings, and visualization and analysis for climate as well as related domains. We identified three steps towards the long-term goal:

- Extend the Purdue CCSM gateway to support online CESM 1.0 simulations on the TeraGrid
- Automatically generate standardized metadata for model runs initiated from the CESM portal
- Integrate data publishing and wide-area transport capabilities such that model run datasets and metadata may be published back into ESG from Purdue’s new CESM gateway

The end result of this work is an environment that provides access to integrated modeling, data management, search, browse, analysis, and visualization functionality for climate research, effectively harnessing TeraGrid resources and engaging new users. The rest of the paper is organized as follows. Section 2 describes the overall system design and details of the main components. In Section 3 we break the end-to-end CESM workflow into three sub-workflows and describe their implementation. Section 4 discusses our early efforts to engage users. Section 5 presents the conclusion of our paper and discusses our future plans.

2. System Design and Implementation

The primary goal of this project is to build a distributed environment which provides seamless services that range from setting up and running CESM climate simulations on the TeraGrid to automatically publishing model provenance information together with the output back to ESG. ESG is a distributed data repository and a central place for the climate science community to locate data. With this goal in mind, we designed and developed a prototype system based on the architecture shown in Figure 1. There are three layers. At the bottom are the computation and storage resources. The model currently runs on the Steele Linux cluster at the Purdue TeraGrid site. The output is stored in a scratch disk space after simulation and may be transferred into an iRODS data grid later [4]. There is a standard ESG data node running on a virtual machine (VM) at Purdue which has access to the model data in either the scratch disk or iRODS via a FUSE interface. It directly communicates with and publishes model results to the ESG gateway. The middle layer consists of a set of web service interfaces named *CESM-WS*. It provides standard interfaces for setting up and running CESM simulations, post processing output, transferring data, and publishing model data/metadata to ESG. Two science gateways are currently building on top of CESM-WS: the CESM modeling portal developed at Purdue and the ESG gateway developed at NCAR. These two gateways are connected by workflows enabled via CESM-WS. The design and implementation of CESM-WS is flexible and may be used to build third party applications as well. CESM-WS is integrated with a token manager which supports authentication against either the CESM gateway account database or the TeraGrid MyProxy server.

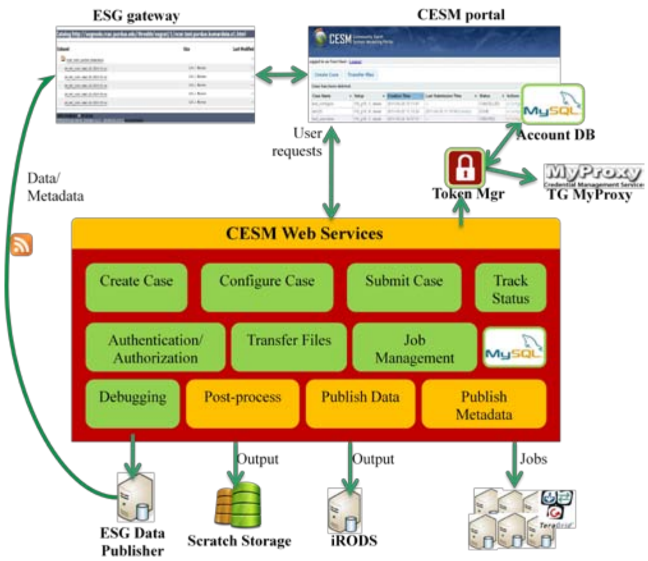

Figure 1. System architecture

2.1 Model Deployment

Feedback from our current user base indicates that different versions of CESM or CCSM will be needed during the transition from CCSM to CESM. Although the target modeling system for this environment is CESM 1.0 and beyond, some users may have CCSM4 or CCSM3 runs that they need to continue for their research. Metadata capabilities in CESM are evolving and future versions will have improved capabilities. The Purdue portal will retain the capability to run with multiple model versions.

The next release of CESM, scheduled for May 2011, will include metadata handling via the ESMF Attribute class. Attributes are sets of name-value pairs that are arranged into standard packages, such as component level information, field level information, responsible party information, etc. These packages may follow specific conventions – the ISO standard for responsible parties and the Climate and Forecast (CF) conventions for field level metadata, for example [9,10]. Attribute packages can be nested and aggregated into larger packages, such as the METAFOR Common Information Model (CIM) for describing full climate model simulations [8]. The CIM package is currently being integrated into the CESM. At this point, CESM can output many elements of the CIM metadata as an XML file, and validate the resulting XML file against the CIM schema definition. Once it is released, this new version of CESM will be used in the Purdue portal to produce comprehensive metadata.

Versions of CESM must be ported to the Steele cluster at Purdue to work in our end-to-end environment. This is a relatively straightforward process. The CESM installation is comprised of model source code and utility scripts for case creation and manipulation. The model source code is copied into each case created by users and compiled just before execution. The process of porting CESM to the Steele cluster is merely to create a new profile holding the collection of compilation flags and any other customizations required for the environment. Users create their model cases specifying this profile for the cluster, and the case is generated using these options for compilation and execution. Testing the installation involves creating cases and confirming the cases are operating as expected within the cluster environment.

2.2 CESM-WS

The CESM-WS interfaces connect the gateway front ends with the backend resources. They are implemented using a generic framework for domain specific application services called DomWS [11].

2.2.1 DomWS framework

The DomWS framework was designed to support rapid creation of domain specific services for running complex and computation intensive models and simulations on HPC resources. It separates the semantics and workflows that are domain and application specific from those used by the backend grid execution environment.

DomWS has a component-based architecture. It includes four layers: web service end point, application service, common components, and grid interface modules. Clients may send requests using either SOAP or REST protocols at the web service end points. Tasks submitted are persisted as entity Java Beans, making it easy to manage and track status. Application service first validates the input based on the domain-specific business logic of the application and then constructs a request for the common components layer. The common components layer consists of general-purpose components for user authentication, authorization via token management, job management, and job submission. As the final step, grid interface modules are responsible for submitting HPC jobs to a scheduling system such as Globus or Condor.

The implementation of DomWS follows the Enterprise Java Bean (EJB) specification [12]. It uses the Plain Old Java Object (POJO) to avoid implementing services that are tightly coupled with a specific application server. Several components are implemented in BeanShell script [13] which is a light weight embeddable Java source interpreter with object scripting language features. It increases the flexibility and extensibility of the framework.

2.2.2 CESM-WS services

The CESM-WS service interfaces were developed using the DomWS framework. DomWS already provides several common components for grid operations and token management. It also provides the template for inserting business logic that is specific for the model services to be developed. As a result, it is relatively easy to extend DomWS to support CESM simulations.

The functions supported by the CESM-WS interfaces include case creation, configuration and submission, job status query, file transfer, authentication, job management, log file query, and data publication. We are in the process of implementing the methods for data post-processing and metadata publication.

The authentication method ensures that only valid clients can access the CESM-WS services. A client will first need to authenticate itself. Its request is forwarded to a token manager which checks against a portal account database or the TeraGrid MyProxy server. If successful, the client will receive an access token which is a 32-digit UUID string. The token manager also caches the proxy certificate returned by the MyProxy server in its database. A valid access token is required each time the client sends a service request. A client does not need to authenticate again until his token or cached proxy certificate expires. CESM simulations may be submitted using either a shared community account or a proxy certificate of an individual TeraGrid account.

Two major improvements have been made to CESM-WB since we presented the work in 2010: tracking resubmitted jobs and large data transfer. First, it is a common usage scenario to configure a long CESM simulation to run as a series of jobs, each is automatically resubmitted from where the previous one finished. This process is controlled by an internal CESM script. As a result, the CESM-WS submitter module is not aware of whether a CESM job has been resubmitted or not. In order to track the status of resubmitted jobs, we modified a wrapper script to collect the IDs of resubmitted jobs by inserting additional code into the template used to generate the internal resubmission script. The system periodically queries the status of these jobs via Globus, and enters them into a database. Second, currently the data upload and download interfaces are implemented using SOAP attachment. The size of the data that can be transferred is restricted due to its memory requirement. This approach also does not scale well when there are multiple users who need to transfer files at the same time. To solve this problem, we are setting up an FTP server which is integrated with the token manager for authentication. Users can directly transfer data to/from the FTP server with a valid security token.

2.3 CESM portal

The CESM portal is a web-based user interface for accessing CESM services. It allows users to configure and submit CESM cases to the TeraGrid, access the input and output data and publish results to the ESG. In the future, users will also be able to submit the data for post-processing and visualize the results directly in the browser.

The portal also manages a user account database, which is shared by the token manager for authentication. Users can register an account through the portal and use this account to access CESM services. The web service in turn talks to the token manager to authenticate the user and issues a token.

The CCSM3 portal was developed using the Gridsphere Portal framework. For the CESM portal we moved to the Spring MVC framework, which we found to be simpler to use and less restrictive than Gridsphere both in terms of code development and deployment. Both frameworks are based on Java servlet technology and can be run in any standard servlet container such as Apache Tomcat.

Another advantage of using the Spring framework is that it automatically manages the life-cycle (instantiation and disposal) of components based on an XML configuration. In our application, we use this to automatically manage connections to the web service and database. Previously with the CCSM3 portal, we were doing this manually and encountered resource leaks which were difficult to debug.

The design for the CESM portal user interface is also different from the CCSM3 portal based on user feedback. Instead of having separate portlet pages for each step of the model workflow (create case, configure case, submit case, and manage jobs), we chose a more streamlined interface that combines them into one web page which dynamically generates the content based on the user selections and the control flow of CESM simulations. After login into the portal, a user is presented with a page listing the existing CESM cases along with the current status of each case and options for managing the case (Figure 2). The user can then perform different actions by clicking on the corresponding links/buttons from the current page.

The basic simulation workflow proceeds as follows:

1. **Create a new case** by specifying a case name and parameters (component set and resolution)
2. **Configure the case** by editing various configuration parameters; the gateway provides a basic and an advanced interface to suit different users
3. **Submit the case** to a job queue on the TeraGrid

The gateway also provides file transfer capabilities so that users can upload input data for cases and download output files once the simulation is complete. The current implementation uses a Java applet on the client side for users to browse and upload/download files. Requests are first sent to the gateway, and then forwarded to the web service which accesses files in the user's directory on the back-end. This approach requires data files to be buffered by the intermediate servers, and we found that this causes problems for large files that are several hundred megabytes in size. As discussed in section 2.2, we are currently redesigning this feature to make use of an FTP server, so that clients can directly stream data to and from the server.

Our experiments with the CESM portal also helped us identify some performance bottlenecks in CESM-WS. We noticed that certain parts of the portal workflow took a long time (up to a minute) to process. By logging the sequence of web service calls invoked by the portal, we were able to identify those methods that took a long time to complete and optimize the web service by eliminating unnecessary Globus calls. For example, each request to retrieve the list of cases required a Globus call to update the status of running jobs. This caused the user to experience delays when loading the web page. We were able to eliminate this delay by caching the status information in the web service and running a *cron* job to periodically execute a Globus call to update the cache.

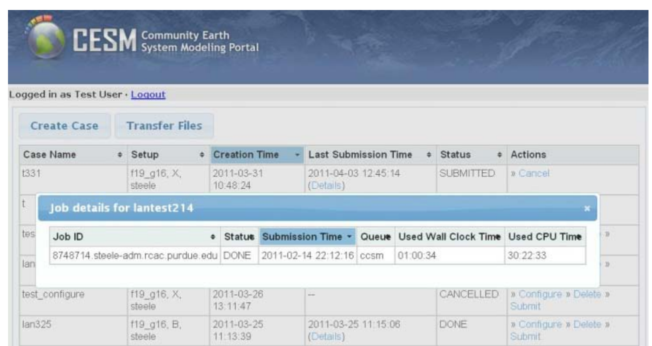

Figure 2. A screenshot of the CESM portal job table interface. The popup window displays the detailed information of one selected job.

2.4 ESG gateway and data publisher

The ESG Federation is built upon the model of a collection of gateways that are federated via metadata, security, and policy arrangements. There are currently ESG gateways in the USA, the UK, and Germany. ESG gateways are connected to one or more ESG data nodes, each of which has a software stack that provides a configurable range of services, including dataset publishing, download, security, metrics, data product services, and more. A data node manager can execute a publication process that shares extensive dataset metadata with its host gateway, which aggregates that information with all of the other data node information along with information shared by other federated gateways.

The ESG gateway provides authentication, authorization, search, browse, and data download capabilities that utilize the service provided by the data nodes. Through collaboration with the Curator project, the ESG gateway can also ingest CIM metadata records via an ATOM feed, and provides a model trackback interface where users can inspect and navigate the rich metadata contained therein. A snapshot of the ESG/Curator interface is shown below.

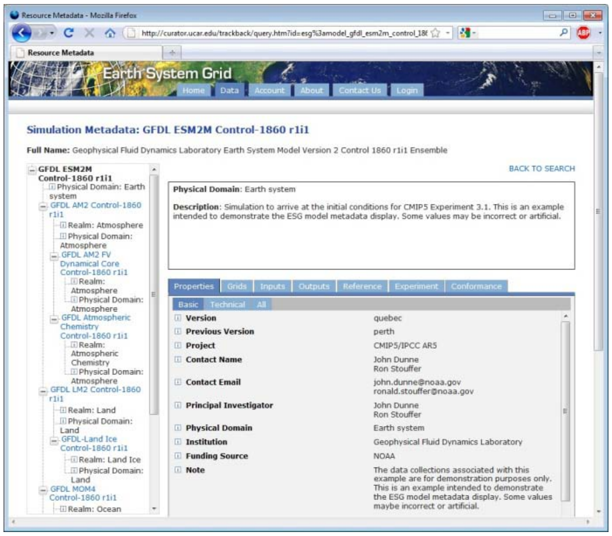

Figure 3. The ESG/Curator Model Trackback Interface

The CESM portal utilizes the ESG data node components within a virtual machine, the preferred installation environment for the ESG data publication software, which allows the entire publication process to be controlled asynchronously from the cluster. The cluster and the virtual machine share a storage file system through which data is transferred between them, and a queue system has been developed for the virtual machine to monitor for publication requests. In addition to the standard ESG data node publishing subsystem, we also developed a specific ATOM feed component that allows for model-generated CIM documents to be fed to the NCAR ESG gateway. The data node, the model, the CIM ATOM feed, and the ESG gateway can be located near one another, or distributed.

3. Workflow

In this section, we explain the end-to-end CESM model workflow enabled by our integrated environment. For the sake of convenience and clarity, we break it down to three sub workflows and discuss them sequentially.

3.1 Basic model simulation

As shown in Figure 4, the basic CESM model simulation workflow includes the following steps: model creation, model configuration, model submission, status monitoring, data postprocessing, visualization, and model data access. The CESM portal provides form-based user interfaces for each step. It then invokes the CESM-WS SOAP interface which submits Globus jobs to the Steele cluster. The Globus job typically runs a shell script which is a wrapper of the corresponding CESM internal script for case creation or submission.

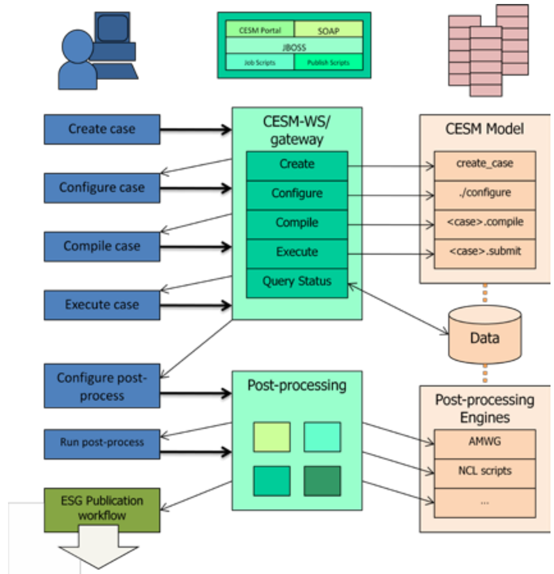

Figure 4. CESM model execution workflow

Currently all jobs submitted by the CESM portal use a shared community account to run on the TeraGrid. It is also possible for a third party client to submit jobs using its own TG allocation through CESM-WS. In this case, the client first invokes the authenticate method against the TeraGrid MyProxy server. If successful, the token manager stores the returned proxy certificate in its cache, and in the same time, it issues a token to the client which may be used later for job submission with the downloaded proxy certificate.

3.2 Data publishing

The ESG data node VM operates independently from the execution cluster upon which the model cases are submitted. This means there has to be a system through which the CESM portal and the cluster can signal to the VM that a publication is needed. The publication process can be lengthy with the amount of data that is scanned and processed before being published, so it must also happen asynchronously from the rest of the model workflow. The solution is to create a spool to queue publication operations for the VM to monitor.

As shown in Figure 5, scripts have been added to the CESM profile for the cluster such that when new cases are created, these scripts are generated within the case directory along with the standard CESM scripts for compilation and execution. These scripts serve two purposes. They queue the dataset generated by the case for publication and allow the system to check the status of the publication.

When a case has finished execution, the user can choose to publish the results. The CESM portal executes the script for queuing publication in the same manner it would run the script for compilation or execution of the job. The queuing script generates a description of the dataset and deposits it as a queue entry within the shared filesystem area designated to hold the spool of items waiting to be published.

The publication VM periodically checks the queue for the next item to process. The path to the dataset, the name under which to list the data, and other publication values are parsed from the queue entry, and the VM executes the publication process. The queue entry is updated to indicate the current status of the publication of the dataset. Throughout the whole process the

CESM portal can utilize the added scripts within the case directory to query for the status of publication by reading these values from the queue entry.

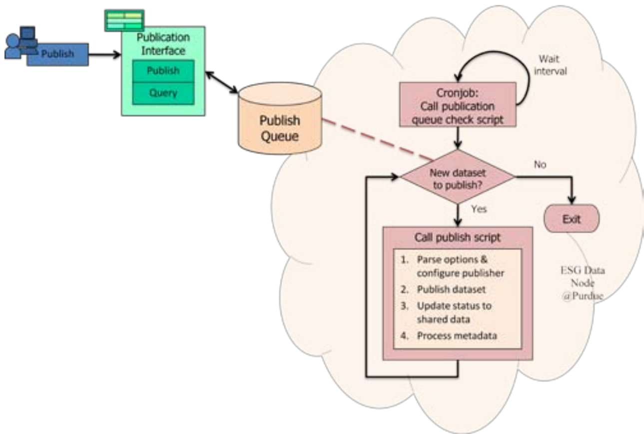

Figure 5. CESM model output publication workflow

The ESG Publisher component is used by the data manager to publish selected datasets into a gateway. The Publisher scans the datasets, checking for correctness and completeness, extracts metadata from the files, and ultimately generates a THREDDS catalog that is inserted into the data node's TDS (THREDDS Data Server) [14]. After that is done, it contacts its gateway via a protected web service, authenticates using a MyProxy certificate, and delivers the URL of the new catalog. At this point, the ESG gateway fetches the new catalog, parses it, inserts the new metadata into its database, and harvests a portion of discovery-level information into its Sesame triple-store. At this point, the new dataset(s) will be available in the system for users to access. As the process proceeds, the Publisher periodically checks status with the ESG gateway until it receives a successful "publication completed" status.

3.3 Metadata publishing

All of the main components in CESM plus the driver/coupler were modified in order to accommodate the inclusion of metadata using ESMF Attributes. This metadata follows the CIM schema and includes the names and descriptions of components, authors, versions, and references; computing system information, such as compiler name and version; names and units of fields transferred between components; grid description and resolution; and more. The CIM supports detailed scientific attributes, so that specific algorithms can also be identified, although this has not yet been implemented in CESM. The Attribute class uses the Xerces library to write out a XML document in a prescribed form, which can then be ingested into ESG or another gateway.

Metadata publication is an unobtrusive process in the overall data publication workflow. The ATOM feed server is deployed on the publication VM at Purdue alongside the existing ESG services. When the VM processes a dataset for publication, it copies the metadata file generated by the model into the ATOM feed server staging area. The model metadata (CIM) will automatically be included for the ESG gateway to find during its next scan of the feed.

The CIM XML is then fed to the ESG gateway through the ATOM feed, at which point it is converted into RDF, and then loaded into the ESG Sesame triple-store. At this point, the information is available to the ESG gateway, and the model trackback interface (shown above) is available to users for accessing and navigating the information. The overall process is depicted below (Figure 6).

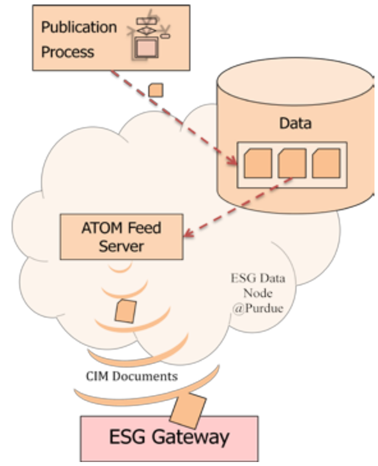

Figure 6. CESM model metadata publication workflow

4. Use Cases

We are currently working with three user groups as their use cases help improve the design and implementation of our end-to-end system. These use cases are outlined below.

High resolution paleoclimate runs. The first use case involves a research group of geologists from the University of California at Davis who wish to compare atmospheric model output data from paleoclimate runs with observational data that they have collected. Although the researchers are collaborating with a climate modeler to perform the study, they have little experience running large models and would like an interface that is as easy to use as possible. They need to use the CCSM3 model since they are continuing previous runs from restart files at particular points in the simulation. The researchers wish to run the climate model at a higher resolution than is typical for paleoclimate runs, so the restart files are being interpolated from T31 (about 4 degree spacing) to T85 (about 1 degree spacing). The Purdue portal is being used to support configuration and submission of the runs, and archival of the resulting data in ESG. Since the earlier version of CESM is being used, this use case will not exercise the metadata capability.

CESM users at NCAR. The second use case concerns scientists who perform research using the CESM model at NCAR, and staff there who run the model in specified experiment configurations for assessments such as IPCC. For both of these cases, collecting adequate metadata is a challenge. The environment constructed on the TeraGrid is contributing elements such as automated metadata collection to the local CESM workflow, and a means for importing this metadata either into ESG or a local database display is needed. This workflow will exercise the metadata part of the environment. It may not address ESG publishing since NCAR maintains a local run database for research.

Climate researchers and students at Purdue. The third use case includes students and researchers at Purdue who will be able to leverage the enhanced Purdue portal to make the CESM simulations they perform more widely accessible through federation with ESG. They will also be able to document their runs more comprehensively and automatically through the enhanced metadata capability. These workflows will engage all aspects of the end-to-end environment, including post-processing and analysis.

5. Conclusions

In this paper we describe the design and implementation of a collaborative project for building an integrated, seamless environment for CESM simulation, analysis, description and publication. A prototype system has been developed that supports end-to-end climate modeling workflows including CESM simulations on the TeraGrid via the CESM portal, generation of standardized metadata for model description, and automatic archival of data and metadata to the ESG gateway.

For future work, we are developing additional data analysis, visualization, and transfer services in the CESM portal. For example, we are implementing data post-processing capabilities in the CESM portal so that users can directly analyze their data online without the need to download a large amount of output to their local systems. We also added the server side support for uploading large model input data using secure FTP and are in the process of integrating it into the CESM portal. This function will solve the issue of size restrictions on uploading input data through the current http based interface. It will also significantly shorten the amount of time it takes to upload data to a model site. We also plan to develop another prototype Earth System Grid-Curator (ESGC) Science Gateway which can configure and initiate CESM runs on the TeraGrid as part of a documented workflow.

The long term objective for the project is to make the prototype into a production system that supports daily research and teaching activities related to climate simulation. The success will depend on the improvement of performance, scalability and reliability of the system, in addition to easy access to model input and output. To achieve this goal, we plan to expand the CESM portal to be able to run CESM simulations on more TeraGrid computation resources in addition to Purdue's Steele cluster, and investigate the possibility of using TeraGrid DC-WAN for storing model data generated from distributed TG sites. More importantly, we will continuously work to help and support users, as well as to engage the broader user community and obtain feedback in order to make a system that truly meets users' needs.

6. ACKNOWLEDGMENTS

This research is sponsored by National Science Foundation under the TeraGrid Resource Partner award (#0503992 and #0503697), the TeraGrid Extension award (0932251).

7. REFERENCES

[1] Community Earth System Model (CESM): http://www.cesm.ucar.edu/.

[2] CESM1 timing table: http://www.cesm.ucar.edu/models/cesm1.0/timing/.

[3] L. Zhao, W. Lee, C. X. Song, M. Huber and A. Goldner. "Bringing High Performance Climate Modeling into the Classroom", TeraGrid 2010 conference, Pittsburgh, PA, August 2010.

[4] iRODS: Data Grids, Digital Libraries, Persistent Archives, and Real-time Data Systems. http://www.irods.org/

[5] Earth System Grid: http://www.earthsystemsgrid.org/

[6] Dunlap, R., L. Mark, S. Rugaber, V. Balaji, J. Chastang, L. Cinquini, C. DeLuca, D. Middleton, and S. Murphy (2008). Earth System Curator: Metadata Infrastructure for Climate Modeling, *Earth Science Informatics*, Volume 1, Numbers 3-4, pp. 131-149.

[7] Hill, C., C. DeLuca, V. Balaji, M. Suarez, and A. da Silva (2004). Architecture of the Earth System Modeling Framework. *Computing in Science and Engineering*, Volume 6, Number 1, pp. 18-28.

[8] http://metaforclimate.eu

[9] http://www.iso.org/iso/home.html

[10] http://cf-pcmdi.llnl.gov/

[11] Han Zhang, Jungha Woo, Lan Zhao, David Braun, Carol X. Song, Madhavan Lakshminarayanan. "Domain-specific web services for scientific application developers," submitted to Gateway Computing Environments workshop at SC10, New Orleans, LA, November 2010.

[12] Floyd Marinescu, "EJB Design patterns", Wiley publishing, 2006, pp 112-116

[13] BeanShell –Lightweight Scripting for Java, http://www.beanshell.org/home.html

[14] THREDDS: http://www.unidata.ucar.edu/projects/THREDDS/