

# A Community Climate System Modeling Portal for the TeraGrid

Ayon Basumallik   Lan Zhao  
Carol X. Song

Rosen Center for Advanced Computing,  
Purdue University, West Lafayette, IN 47907-1285  
{basumall,lanzhao,carolxsong}@purdue.edu

Ryan L. Sriver   Matthew Huber

Dept. of Earth and Atmospheric Sciences,  
Purdue University, West Lafayette, IN 47907-1285  
{rsriver,huberm}@purdue.edu

## Abstract

The Community Climate System Model (CCSM) is a coupled climate modeling framework for simulating the earth's climate system, that allows researchers to conduct fundamental research into the earth's past, present and future climate states. While the model is well documented, the learning curve for new users is steep and porting the model to a new platform can be difficult and time-consuming. In this paper, we describe an effort to make the CCSM framework more easily accessible and usable to a large class of users by means of a portal. We present a TeraGrid based web portal that allows TeraGrid users to run CCSM simulations on TeraGrid resources without having to individually port and install CCSM and without encountering lower level details such as specifics of batch queues and library locations. We describe the back-end CCSM installation, the front-end user interface for the CCSM portal and present an overview of a post-processing framework for the CCSM simulation results.

**Keywords** Community Climate System Model, Portal, User-Interfaces, Workflows, TeraGrid, Cyberinfrastructure

## 1. Introduction

The Community Climate System Model (CCSM) [3] is a coupled climate model for simulating the earth's climate system. Composed of four separate models simultaneously simulating the earth's atmosphere, ocean, land surface and sea-ice, and one central coupler component, the CCSM allows researchers to conduct fundamental research into the earth's past, present and future climate states. The CCSM is utilized by a large community of students and scientists worldwide. It was initially developed and housed at the National Center for Atmospheric Research (NCAR) located in Boulder, Colorado. Researchers use the model to simulate and understand the mechanisms affecting interdecadal, interannual and seasonal variability in the Earth's climate system on multiple spatial scales ranging from regional to global. Scientists model the distant past to gain insight into the changing nature of Earth's climate system with conditions much different from the present-day, and the knowledge gained from these simulations is an important tool for comparing against observational records. Furthermore, projections of future climate scenarios based on present-day conditions and trends of climate-sensitive atmospheric constituents serve as an important resource for policy makers to make informed decisions that may affect future generations.

The CCSM thus provides the modeling framework for confronting scientific questions about the Earth's past, present and future climate states. The CCSM simulations are resource intensive, in terms of computational cycles as well as storage space requirements. The CCSM documentation states that a typical model run on an IBM "bluesky" system (a set of IBM Power4 8-way nodes), at a dataset resolution of  $T42\_gx1v3$ , has the following requirements :

- History-File Volume: 6.5 Gbytes/model year
- Restart-File Volume: 0.9 Gbytes/model year
- Total CPUs: 104
- Simulation Years/Day: 7.5

In our own experiments with the validation runs of CCSM on the IBM "DataStar" platform at the San Diego Supercomputing Center, we have observed a throughput of 10 Simulation Years per 12 hour run for a dataset resolution of  $T31\_gx3v5$  with active models for all the CCSM components.

Thus, CCSM is well suited for the type of High Performance Computing resources available on Grid systems like the TeraGrid [9]. The TeraGrid comprehensively addresses the need of today's computational scientists by providing high-performance computing cycles that enable more powerful simulation and numerical approaches; network storage resources that accommodate larger datasets to allow modeling of scientific data at finer resolutions over larger meshes; and libraries and tools to facilitate the development of scientific applications. Keeping in mind these attributes of the TeraGrid, we started an effort to allow researchers in climatology to perform the CCSM simulations on TeraGrid resources.

To expose CCSM users to the plethora of possibilities provided by the TeraGrid, we perceived a need for tools and environments that allow them to easily interface with TeraGrid resources. While TeraGrid resources include software stacks for essential activities such as authentication, job submission and resource monitoring, directly learning and using these may be a daunting task for a large class of users. Additionally, the simple scenario of each CCSM user setting up a personal CCSM installation to perform simulations has certain disadvantages –

- The CCSM Installation procedure on a new platform, while well documented, is non-trivial and time-consuming.
- A per user CCSM installation would be wasteful in terms of space used by the standard input datasets and program code.
- The collaboration opportunities offered by the TeraGrid would not be exposed to the users without a framework that allows them to easily share their simulation results.

To address these issues, we have created a "community" installation of the CCSM on the TeraGrid and a web-portal based interface for accessing this installation. The motivation for this CCSM portal is to lower the barriers of entry for people into climate modeling by making available a working CCSM installation on the TeraGrid readily accessible via a web interface. In recent times, web-based portals such as the NanoHUB [14] and the LEAD portal [10] have become invaluable tools for their user communities. We envision that the CCSM portal will similarly assist climatological research and encourage the use of CCSM and the TeraGrid for climate modeling purposes. The goal of this portal is to provide an intuitive interface, both for experienced and new CCSM users,

while shielding them from the complexities associated with command line grid-computing interfaces. While it is important to provide a user-friendly portal interface to run CCSM simulations on TeraGrid resources, there is also a pressing need to make it easier for researchers to further process and analyze the results from previous and current model runs. For example, very often researchers need to examine the intermediate model output by running some diagnosis tools to decide whether the current configuration is appropriate. To address this demand, we are developing a climate data post-processing component for the CCSM portal which provides an easy-to-use interface to diagnose and analyze final or intermediate model output. The CCSM portal for the TeraGrid project thus aims to comprehensively address the issues required in using CCSM on the TeraGrid.

The rest of this paper is organized as follows – Section 2 provides an overview of the issues involved in designing a portal for CCSM. Section 3 describes how our portal implementation addressed these design issues. Section 4 describes a post-processing component for CCSM data in the context of a broader data service framework. Section 5 describes future work. Section 6 concludes the paper. Related work is discussed throughout the paper.

## 2. Portal Design for the Community Climate System Model

In this section, we discuss the design issues considered in creating a portal for CCSM. We first present a brief overview of CCSM itself and describe the work flow associated with a typical CCSM run. Then, we describe the setup of the CCSM “community installation” on the TeraGrid and discuss how the portal connects to different units within this installation.

### 2.1 The Community Climate System Model

As discussed previously in Section 1, CCSM is made up of four components (interchangeable referred to as models) that simultaneously simulate the earth’s atmosphere, ocean, land surface and sea-ice, and one central coupler component. Each model contains *active*, *data* and *dead* component versions allowing for a variety of “plug and play” combinations. The active model versions (also called dynamical models) perform actual simulations. The data-cycling model versions (data models), on the other hand, are small, simple models which simply read existing datasets that were previously written by the dynamical models and pass the resulting data to the coupler. These data-cycling components are very inexpensive to run and produce no output data. For these reasons they are used for both test runs and certain types of model simulation runs. Currently, the data models run only in serial mode on a single processor. The dead model versions are simple codes that facilitate system testing. They generate unrealistic forcing data internally, require no input data and can be run on multiple processors to simulate the software behavior of the fully active system. A CCSM component set is comprised of five model components - one component from each model. All model components are written primarily in FORTRAN 90. During the course of a CCSM run, the four non-coupler components simultaneously integrate forward in time, periodically stopping to exchange information with the coupler. The coupler meanwhile receives fields from the component models, computes, maps and merges this information and sends the fields back to the component models. By brokering this sequence of communication interchanges, the coupler manages the overall time progression of the coupled system.

Figure 1 shows interaction of the active models and the parallelization scheme used for each of them. The models periodically communicate with the coupler component via MPI messages.

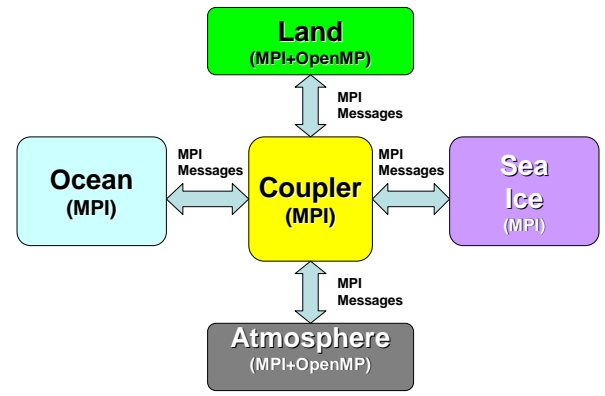


Figure 1. CCSM Components

A typical CCSM simulation run consists of the steps shown in Figure 2. The user first creates a new simulation case, specifying a case name, a dataset resolution and a *component set* consisting of one version of each of the five models. With these parameters, scripts within CCSM generate platform specific configuration scripts. The user then executes these configuration scripts to generate build and execution scripts for a CCSM run.

In the second step, the user edits the generated scripts to set specific parameters such as the time span of the simulation, the frequency of data and trace generation and so on. The user, at this step, also prestages any non-standard input data that will be used in the simulation. The user then invokes the generated build scripts that build libraries and executables, as required, for the five components and also pre-stages the standard input data required by the simulation.

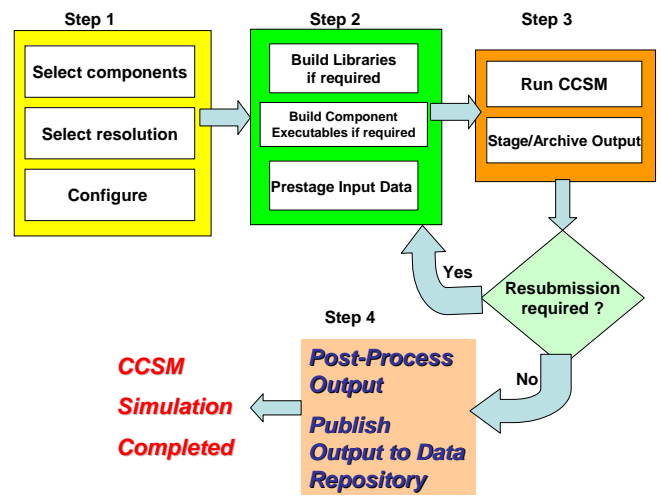


Figure 2. Steps in a Typical CCSM Simulation

In the third step, the CCSM simulation is run. Depending on the platform, archiving scripts may be used at this step to stage and archive the output data produced. In the fourth and final step, the output data produced by the CCSM run is post-processed and may be published to data repositories.

To feel intuitive, the portal interface needs to correspond to the three steps discussed above. In Section 3, we discuss how the portal front-end was implemented, keeping in mind this requirement.

## 2.2 Portal Back-End Design

To implement the CCSM portal, the TeraGrid resource that we selected for running the CCSM simulations is the IBM DataStar system at the San Diego Supercomputing System (<http://www.sdsc.edu/us/resources/datastar/>). We found this system to be well suited to the type of parallelism used by the CCSM components. As a first step, CCSM was installed and the CCSM scripts were ported for the DataStar platform. Then, a 50 year validation run was completed, at a dataset resolution of  $T31\_gx3v5$  and the results were validated. Having validated the DataStar platform, we created a community disk space for CCSM. The community space was designed to hold a single copy of the CCSM codebase and the standard input files. For each user, separate directories are created to hold the user's CCSM cases created and the corresponding output. Additionally, separate per-user directories are created to store CCSM input files and user specified CCSM component code versions as well. Figure 3 shows the layout of the CCSM community space.

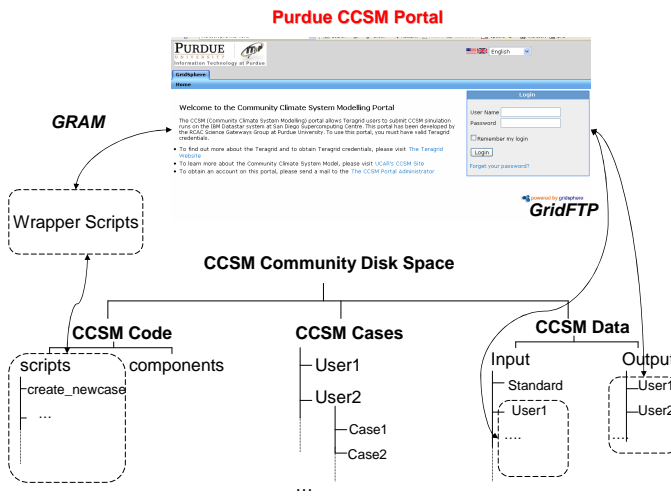


Figure 3. Back-End layout of the CCSM Portal

CCSM contains its own scripts for building the components, staging input and submitting jobs to DataStar's LoadLeveler scheduler. For our portal, we reuse these scripts by using constructing wrapper scripts for configuring, building and running CCSM cases.

The portal front-end (discussed in Section 3), has visual interfaces for configuring, building and running CCSM cases. The interfaces invoke the wrapper scripts by means of the GRAM protocol [12]. These wrapper scripts, in turn, invoke the corresponding CCSM scripts. The reason for having this intermediate layer of wrapper scripts is two-fold. Firstly, by using wrappers, the actual CCSM scripts are invoked within a unix shell that makes the requisite TeraGrid environment variables available to the CCSM scripts. Secondly, there is currently no additional globus support for directly communicating with the LoadLeveler batch system (there do exist direct globus support for other batch queuing systems such as PBS and Condor). The use of these specialized wrapper scripts allow the CCSM loadleveler scripts to be invoked as they would be from a normal unix shell while allowing job responses to be conveyed back to the portal.

For setting up this CCSM community space, we have selected the GPFS-WAN file system [5]. This makes the community space visible on other supported TeraGrid sites as well. This allows users to directly access the data produced by CCSM runs on other sites,

where they may wish to execute the post-processing phase, without having to explicitly move the data.

Finally, to directly access the data (both output data and input data/programs that the user may wish to pre-stage for a particular CCSM run), our portal provides data transfer interfaces that access the community space using the GridFTP protocol.

## 3. CCSM Portal Implementation

In the previous section, we discussed the CCSM installation done on the DataStar platform and the layout of the CCSM community space on the GPFS-WAN file system. In this section, we briefly discuss how we implemented the portal front end to provide a simpler and more intuitive access mechanism for this CCSM installation.

In implementing the CCSM portal, we have made extensive use of the GridSphere portal framework [8]. The GridSphere framework provides an open-source portlet based Web portal. It includes a portlet API that is completely JSR 168 [6] compliant and also includes additional API for almost full compatibility with IBM WebSphere. It provides a flexible XML based presentation description that allows customization of the visual interface, as well as a portlet service model for encapsulating, reusing and sharing portlet logic. In addition to the GridSphere API and services, our portal implementation also uses the Grid Portlets API. The Grid Portlets API provides a high level implementation of several Globus protocols and grid resources. Apart from the high level grid resource abstractions provided by these API, our portal also uses the *Action-Portlets* [4] and *ActionComponents* [1] portlet models provided by GridSphere and GridPortlets respectively.

In Section 2, we have discussed the sequence of steps followed in a typical CCSM simulation run. The portal user interface is created to correspond to these steps. Figures 4, 5, 6 and 7 show the interfaces provided by the portal.

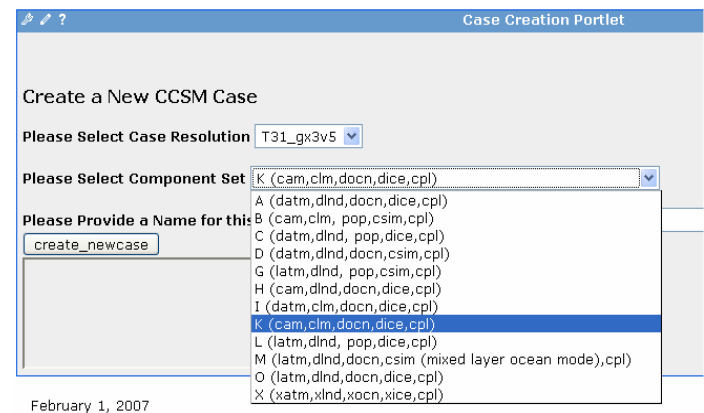


Figure 4. CCSM Portal Interface for Case Creation

Figure 4 shows a snapshot of the case creation interface. The user starts a typical new CCSM run by selecting a component set (a combination of active, dead or data component for each of the five models) and a dataset resolution. Figure 5 shows a snapshot of the case configuration interface. In this interface, the user edits case parameters such as the length of the run, the resubmission criteria, the frequency of trace data collection and so on. After editing existing environment parameters, the user may use this interface to create configuration files for the specific run and edit them further if required. This interface for configuring CCSM cases was created keeping in mind that CCSM users are accustomed to editing certain environment files directly to specify run parameters. So, rather

than a form based interface for selecting the multitude of possible run parameters, our portal allows the users to edit the simulation parameter scripts directly through the portal interface. Figures 4 and 5 correspond to steps one and two in Figure 2.

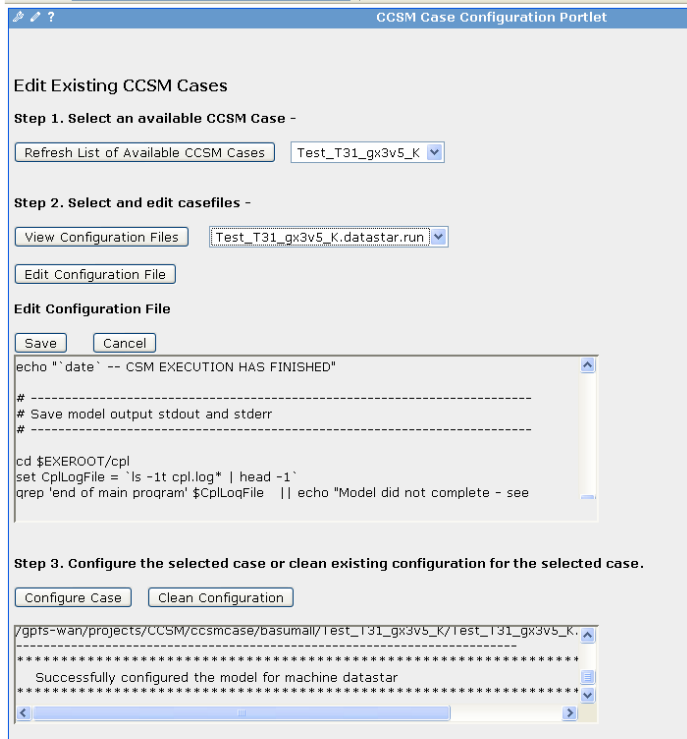


Figure 5. CCSM Portal Interface for Case Configuration

Figure 6 shows a snapshot of the interface used to run a CCSM case. The user simply selects from a list of his own configured cases and builds or runs the particular case. This interface also allows the user to query the status of a submitted simulation on the LoadLeveler queue on the IBM DataStar system.

For staging files throughout this process, or for moving output to other systems (corresponding to Step four in Figure 2), we have created a data transfer interface. A snapshot of this interface is shown in Figure 7. The case creation, configuration and build/run submission interfaces have been created using the *ActionPortlets* API provided by the GridSphere framework. The Data Transfer interface has been created using the *ActionComponent* API provided by Grid Portlets, which allows some Grid Portlet user interface components to be reused in our portal.

#### 4. Climate Data Post-Processing

In conjunction with the portal interface described in Section 3, we have developed a post-processing component for the data produced by the CCSM runs. The climate data post-processing component is built on top of a distributed service-oriented workflow platform currently being developed as part of the Purdue TeraGrid cyberinfrastructure [13]. It provides a web front end that allows users to select the modeling dataset and the types of processing. It then launches a workflow that prepares the data, composes an analysis job, submits it to the TeraGrid Condor pool, and publishes the result to a web server once the job completes. The workflow is a service pipeline that consists of several web services-based modules implementing data-related tasks and modules that connect to the computation and data resources through Grid middleware. It

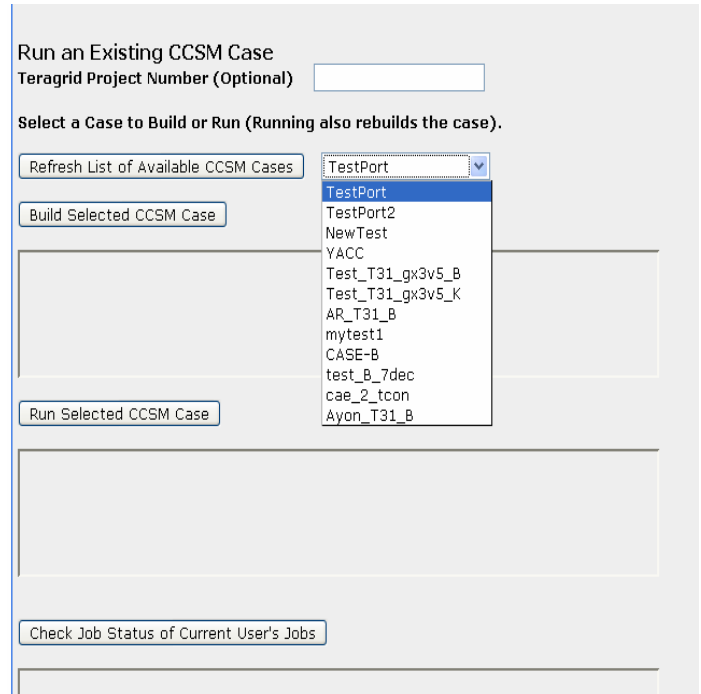


Figure 6. CCSM Portal Interface for Building/Running Cases

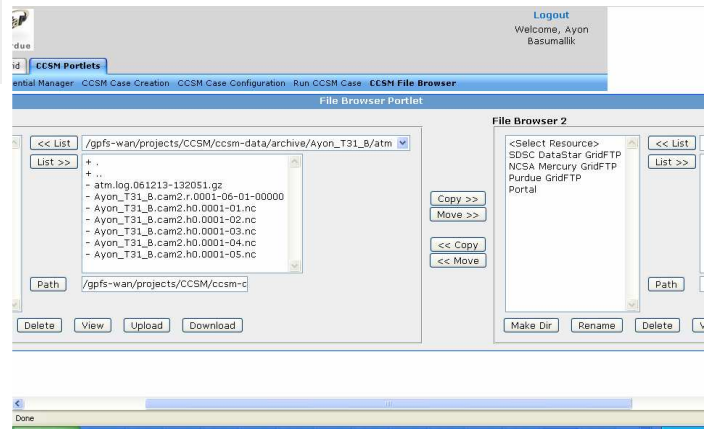


Figure 7. CCSM Portal Interface for File Transfers

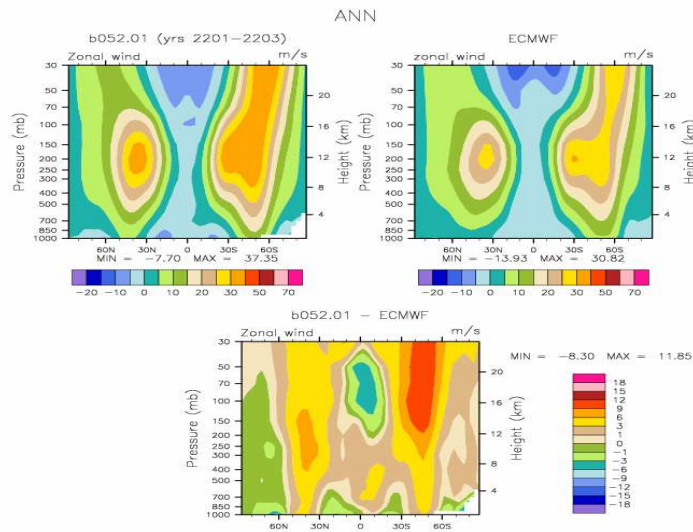
hides the system level details of data and computation resources so that researchers can concentrate on their research problems and tasks. We have implemented a workflow that operates on the result data transferred from the remote CCSM computation resource to the local data repository. The repository is managed by the OPeNDAP (Open-source Project for a Network Data Access Protocol) and SRB (Storage Resource Broker) servers. The workflow consists of the following processing components:

- (1) The *DataQuery* component queries the TeraGrid data management system to locate the data to be processed;
- (2) The *QueryURL* component returns the OPeNDAP URLs which can be used to remotely access the data.;
- (3) The *NCLJob* component constructs a RSL job specification and submits the job to the TeraGrid Condor pool using Globus GRAM API [11]. Internally, the job invokes an NCL (NCAR Command Language) script that performs data analysis operations on the

selected dataset using the AMWG (Atmosphere Model Working Group) diagnostics package [2];

(4) The *PublishData* component uncompresses the results and publishes them to a web server;

(5) The *EmailUser* component sends an email to the user with a link to the web URL where the results are published.



**Figure 8.** An example of the published climate data processing results

The current system is able to generate thirteen data sets with more than five hundred different climate modeling data analysis products using the AMWG diagnostics package. An example data product of latitude vs. pressure/height annual zonal means vertical contour plot is shown in Figure 8. Our next step is to enable post-processing of intermediate model outputs as well as connecting the workflow with other portal components including the job submission and file browsing portlets.

## 5. Future Work

An area of future work for us is credential management. Currently, the CCSM portal requires users to upload their TeraGrid credentials to the TeraGrid MyProxy server. When a user logs in, the portal retrieves their proxy credentials from the TeraGrid MyProxy server and uses it for all communication with the back end such as GRAM based job submission and GridFTP based file transfers. As part of ongoing work, we are implementing a mechanism whereby users, when they first register for a portal account, will directly upload their TeraGrid credentials to the CCSM portal. For this, we are considering existing solutions such as PURSE [7]. Additionally, we are also considering solutions for more easily adding the users DN to the gridmap files on the TeraGrid systems concerned (the IBM DataStar at SDSC and other TeraGrid sites where the user wishes to do any post-processing). In this way, we hope to completely shield the user from the complexities of certificate and proxy management.

## 6. Conclusion

In this paper, we have described an effort to make the Community Climate System Model accessible to a large class of users. Our CCSM portal provides a fully functional installation of the

CCSM on the TeraGrid and an intuitive web-based portal interface to access this installation. We have discussed the design issues considered in creating this portal and described both the structure of the CCSM community installation on the TeraGrid and the front-end portal interface for accessing this installation. Finally, we have presented a post-processing component for data produced by the CCSM simulations. We envision that the CCSM Portal will expose the full plethora of possibilities provided by the Community Climate System Model and the TeraGrid to a wide audience of climate scientists.

## References

- [1] Action Component Developer's Guide. <http://www.gridisphere.org/gridisphere/docs/gridportlets/docbook/ActionComponentGuide/ActionComponentGuide.html>.
- [2] CCSM Atmosphere Model Working Group Diagnostics Package. <http://www.cgd.ucar.edu/cms/diagnostics/index.html>.
- [3] Community Climate System Model. <http://www.cesm.ucar.edu>.
- [4] GridSphere Portlet Reference Guide. <http://www.gridisphere.org/gridisphere/docs/ReferenceGuide/ReferenceGuide.html>.
- [5] IBM General Parallel File System. <http://www-03.ibm.com/systems/clusters/software/gpfs.html>.
- [6] JSR 168 : Portlet Specification. <http://jcp.org/en/jsr/detail?id=168>.
- [7] PURSe: Portal-Based User Registration Service. <http://www.grid-center.org/solutions/purse/>.
- [8] The GridSphere Portal Framework. <http://www.gridisphere.org>.
- [9] C. Catlett. The philosophy of TeraGrid: Building an open, extensible, distributed terascale facility. In *Proc. CCGRID*, 2002.
- [10] K. K. Droegemeier, V. Chandrasekar, R. Clark, D. Gannon, S. Graves, E. Joseph, M. Ramamurthy, R. Wilhelmson, K. Brewster, B. Domenico, T. Leyton, V. Morris, D. Murray, B. Plale, R. Ramachandran, D. Reed, J. Rushing, D. Weber, A. Wilson, M. Xue, and S. Yalda. Linked Environments for Atmospheric Discovery (LEAD) : A Cyberinfrastructure for Mesoscale Meteorology Research and Education. In *20th Conf. on Interactive Info. Processing Systems for Meteorology, Oceanography, and Hydrology*, 2004.
- [11] I. Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. In *IFIP International Conference on Network and Parallel Computing*, pages 2–13. Springer–Verlag, 2005.
- [12] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11(2), 1997.
- [13] R. Kalyanam, L. Zhao, T. Park, and S. Goasguen. A Service-Enabled Distributed Workflow System for Scientific Data Processing. In *Proceedings of IEEE Intl Workshop on Future Trends of Distributed Computing Systems (FTDCS07)*, March 2007.
- [14] nanoHUB. Online Simulations for Nanotechnology, 2004. <http://www.nanohub.org/>.