Building a Large Scale Climate Data System in Support of HPC Environment

Feiyi Wang[1], John Harney[1], Galen Shipman[1], Dean Williams[2], and Luca Cinquini[3]

[1]Oak Ridge National Laboratory, Oak Ridge, Tennessee 37831
[2]Lawrence Livermore National Laboratory, Livermore, California 94550
[3]Jet Propulsion Laboratory, Pasadena, California 91109

***Abstract*—The Earth System Grid Federation (ESG) is a large scale, multi-institutional, interdisciplinary project that aims to provide climate scientists and impact policy makers worldwide a web-based and client-based platform to publish, disseminate, compare and analyze ever increasing climate related data. This paper describes our practical experiences on the design, development and operation of such a system. In particular, we focus on the support of the data lifecycle from a high performance computing (HPC) perspective that is critical to the end-to-end scientific discovery process. We discuss three subjects that interconnect the consumer and producer of scientific datasets: (1) the motivations, complexities and solutions of deep storage access and sharing in a tightly controlled environment; (2) the importance of scalable and flexible data publication/population; and (3) high performance indexing and search of data with geospatial properties. These perceived corner issues collectively contributed to the overall user experience and proved to be as important as any other architectural design considerations. Although the requirements and challenges are rooted and discussed from a climate science domain context, we believe the architectural problems, ideas and solutions discussed in this paper are generally useful and applicable in a larger scope.**

I. Introduction

The Earth System Grid project [1] aims to deliver a service-based platform for worldwide researchers to populate, collaborate, disseminate and analyze climate data as part of the scientific discovery process. The ESG project traces its origins back to 1999, when a prototype focusing on large dataset movement and replication was produced. Thanks to technological advances in that time and the growing need for one-stop site for climate science data, its architecture and design have undergone several iterations since. As a result, ESG has grown into a portal-based, fully federated distributed data management system.

The term *federation* reflects two salient features in ESG: (1) the user registration and access control is federated across organizational boundaries with single sign-on support, and (2) the metadata catalogs that constitute critical scientific data holdings (besides the raw data) are distributed and harvested such that each participating organization's gateway can present a unified view (browse and search) on the dataset under ESG management. As a multi-institution, cross-disciplinary project, ESG has made a significant impact to the climate community: 23,000 users registered from over 2,700 sites in 120 countries, over 1PB data downloaded, over 600 published papers, serving as one of the foundational vehicle for data exchange that supported the Intergovernmental Panel on Climate Change (IPCC) 4th Assessment Report (AR4) and will support the upcoming fifth phase Coupled Model Intercomparison Project phase 5 (CMIP5) [2].

As we stand to move ESG infrastructure forward and meet the challenges of upcoming exascale computing and speed up scientific discoveries, it is imperative for us to pause and reflect. This paper reports our experience on the design, implementation and operation of the ESG system.

The paper is organized as follows. Section II gives a brief overview of the current ESG architecture. Section III discusses the data characterization in the context of the data life cycle of a high performance computing (HPC) environment and implications. Section V presents the challenges and requirements of data population. Section VI reflects on the current process for data search and discovery. Finally, we conclude the paper in Section VII with a summary and avenues for future works.

II. ESG Architecture

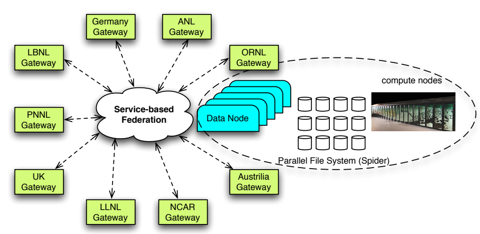

Fig. 1. ESG Architecture

At its core, ESG is a *federated* and *distributed* data system. Distributed means that the data can be hosted at geographically dispersed locations. Federated means the distributed data are conceptually joined together as one centralized archive, where users can search and discovery those data from any number of ESG gateways. In short, the basic setup and data flow for ESG works as follows: each major site can have a gateway, which essentially runs ESG portal Data centers that are collecting or generating and hosting data are responsible to scan and generate *metadata* for the dataset, and then publish to one of those gateways. All the ESG gateways share a trust relationships. The implication is that metadata can be harvested and shared among gateways. As a result, the user can search and download a dataset from any gateway once they have been authenticated by the system.

The following provides a subset on features that developed by various ESG teams:

- Federated Security Layer: Large scale distributed data systems often contain a security layer that adds an additional degree of complexity for both data providers and potential users. ESG abstracts this complexity by providing support for both single sign-on authentication and fine-grained data access control. Once users are signed into the gateway cloud, they may seamlessly search, discover and download data from anywhere on the federation, regardless of whether data is physically stored locally or remotely at another location.
- Bulk Data Transfer: Distributed frameworks like ESG often require support for bulk data transfer, where large-scale datasets are moved across networks around the world. Led by the ESG team at Lawrence Berkeley National Laboratory (LLNL), the Bulk Data Transfer Mover (BDM) [12] provides the capability of transferring files using parallel TCP streams and adjusting optimal levels of concurrency based on bandwidth capacity as well as achievable bandwidth on the wide area network.
- Transfer Management: The configuration steps needed to leverage high performance transport protocols and tools such as GridFTP [4] are often cumbersome and error-prone. Led by the ESG team at Argonne National Laboratory (ANL), Globus Online [9], using the software as a hosted service paradigm has greatly reduced the overhead required for transfer management setup. Integrated into the ESG software stack, Globus Online provides another option for data access.

III. The Lifecycle of Scientific Data in HPC

In HPC environments such as the National Center for Computational Science (NCCS) environment at the Oak Ridge National Laboratory (ORNL), the lifecycle of scientific data follows a very structured workflow, as illustrated in Figure 2. The first step involves the creation of the data itself by domain scientists. In this stage, experts design carefully conducted experiments. Tasks for theses experiments vary, and may include the gathering of initial conditions, validating results against data extracted from instrument observations, or optimization of models through iterative testing. The experiments usually run scientific applications on powerful supercomputers such as the Jaguar [5] supercomputer at ORNL. The results of these experiments, which may be conducted instantaneously or over a substantial period of time, normally generate data that exceed 100s of Terabytes (TB) or even Petabytes (PB). Once the experiments have completed the data providers may include additional information about the data (e.g., provenance information). The data is now prepared to enter the data dissemination and sharing stage. In this stage, two processes occur in parallel. In the first process, data is transferred to permanent storage (especially in HPC environments), which are typically long-term storage systems such as the HPSS system at ORNL. In the second process, metadata extracted from the generated data, is sent to the data management and portal system so that it may be indexed and subsequently discovered by other users. After these tasks are completed, the metadata are ready to be consumed by scientists around the world. These scientists can then search and download either part or all of the data for further analysis.

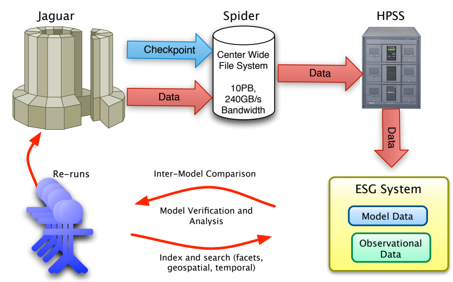

Fig. 2. Supporting End-to-end Scientific Discovery

We believe that the workflow of the data lifecycle in an HPC environment implies four distinctive characteristics about scientific data in general that underpin the overall design considerations of the ESG data management system. These characteristics are data scale, data dispersion, data diversity, and data access control. We briefly discuss these characteristics in the proceeding subsections.

A. Data Scale

At leading computing facilities such as the US Department of Energy (DOE), supercomputers (e.g., Jaguar at ORNL) are running simulations at unprecedented scales. In this context, we define *scale* as both the size of the dataset and the number of files generated from running these simulations. For example, a single run of FLASH astrophysics code produced 74 million data files in 2009. The upcoming CMIP5 collaborated climate data model runs are expected to generate between 5 to 10 PB of data, with over 3 PB of core data replicated at five sites for greater access and backup. The effects of this data explosion can also be corroborated by usage observations seen in Figure 3. It clearly shows the trend of acceleration of data avalanche: reaching 1 PB took us 101 months, 2PB only 20 months etc.

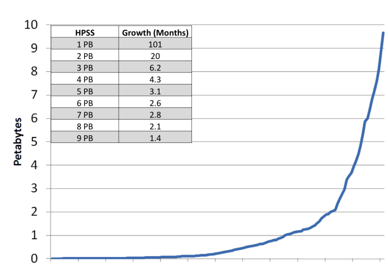

Fig. 3. Exponential Storage Growth at ORNL

This dramatic increase in the scale of data requires us to adopt a modern, restructured I/O model. At ORNL, the ESG team has constructed such a model consisting of two stages. The first stage is the use of a high-performance diskbased parallel file system, know as the Spider system [10]. In this system, the peak aggregate throughput can be as high as 240 MB/s. However, Spider is designed to be a *capability* system, and not a *capacity* system. Even with 10 PB capacity, it is mainly used as a scratch space for scientific applications. We rely on HPSS for long-term data archiving solution.

B. Data Dispersion

Modern climate research efforts often require international collaboration. For example, over 20 climate modeling centers around the world came together to agree upon a set of coordinated climate model experiments, known as the Intergovernmental Panel on Climate Change Fourth Assessment Report (IPCC AR4) and, subsequently, the Coupled Model Intercomparison Project (CMIP5), aim to address scientific questions and obtain a general understanding of the Earth's climate [14]. Each group tackles some portion of this experiment and shares the results. The internationally produced *data island* needs to be connected, metadata harvested, shared and searchable through a coherent interface - thus, the current two-level architecture design was conceived: a physical (i.e. data) node that hosts the data and a virtual (i.e. gateway) node that presents metadata through a THREDDS catalog service [7]. A *publisher* service harvests metadata from the data node and makes this information available to the gateway node, where metadata may be federated and prepared for discovery.

C. Data Diversity

The HPC data life cycle should be designed to support a variety of data formats and metadata structures. In other words, it is highly desirable for a system to be flexible, or *diverse*, so that data need not to be in a specific format for permitted access into the system. The climate sciences community, for example, has a broad spectrum of sub-branches. The modelling community utilizes the NetCDF Climate and Forecast (CF) Metadata structure [8] to define metadata that provide a definitive description in what data each variable represents, and of the spatial and temporal properties of the data. Other bodies of climate-related data, however, utilize other representations. Geographic data, for example, uses the (Federal Geographic Data Committee) FGDC standard for metadata description. This characteristic embodies the inherent open nature of data frameworks in general – both the ingress (i.e., the consumers of the data) and the egress (i.e., producers of the data) should be open-ended and designed for extension. We will revisit this issue in further detail in both Sections V and VI.

D. Data Access Control

Despite the data sharing and dissemination characteristics that a distributed and federated system like ESG exhibit, there are concrete reasons for data access control policies that promote security. Institutions such as ORNL often leverage *internal* resources, such as HPSS archiving systems, that require complex, two-factor authentication codes for access. Other institutions may have other elaborated internal security protocols and access control policies. These policies may include a number of different criteria, including authenticating which data to share, whom to share that data with, or the usage intentions of that data. The tension between the *desire to share* and the *desire to control* place additional burden on system architecture design and potentially has ramifications on both data population and deep storage access.

To summarize, four intervening issues emerged from this context: driven by the data scale comes to the need for deep storage. The requirement on data access control complicated the workflow of both the data population as well as deep storage access. These are the subjects of discussion in the following sections.

IV. Deep Storage Access: Challenge and Solutions

In the preceding sections, we hinted that there may be two reasons for utilizing deep storage (e.g. tape libraries) in the context of persisting scientific data as part of ESG. First, both near and long-term climate data needs far exceed the current resources available for ESG online. To address this shortcoming, we may utilize mature high performance systems such as HPSS, which have 10s of PB in storage capacity for long term storage archival purposes. Second, large scale computing environments, such as the Leadership Computing Facility at ORNL, often design online disk storage focusing on *capability* not *capacity*. For example, the high performance Spider filesystem [10] at ORNL is primarily used as scratch space – simulation results are typically temporary written to this space only to purged periodically. Spider users are often asked to transfer their data to HPSS prior to this purging.

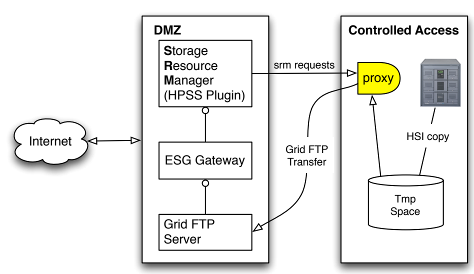

Fig. 4. Controlled Access to Deep Storage System

However, accessing and sharing such data in a system-agnostic way proves to be a challenge. Specifically, there are two aspects of the challenge that needs to be addressed. First, a mechanism is required to bridge file and space management strategies between the ESG gateway and backend storage systems. This is due to the fact that a disparity exists between the front end storage (e.g. HPSS), both in resource capacity and the manner in which files are handled. Second, there are data transfer issues due to the constraint of access control. For example, it is common in HPC environments that *data requests* *can only be initiated from within the internal network*. This constraint naturally stems from security policies that may exist in institutions that participate in ESG.

One potential solution to these issues is to leverage the hierarchical storage system (HSM) [6] technique, which automatically moves data between high-cost and low-cost storage media. This technique is commonly supported by many commercial products such as IBM Trivoli Storage Manager, HPSS, EMC DiskXtender, SGI Data Migration Facility etc. It is, however, not currently supported by the Lustre filesystem [15], which is one of the most widely adopted distributed parallel file systems in HPC centers around the world. While there are several activities within the Lustre community working to develop this feature, it is still not mature enough for production use.

The current solution ESG has employed is to leverage the Storage Resource Manager (SRM) [11], which is designed to abstract the complexity of these types of file transfers from potential users. In this configuration, ESG users will be notified that there will be a delay in their ordered data's delivery. During this time, the backend will be preparing the dataset by transferring from long term storage to ESG storage. When finished, the backend sends messages to the user that the data is available for download.

This workflow is illustrated in detail in Figure 4. The flow of operation corresponds to the following: The user browses the ESG gateway and selects the files to download. ESG gateway support the notion of access points, where SRM is just one of such types. Access point of SRM and Local Disk are transient state as the state changes pending on the data availability. The gateway communicates with Storage Resource Manager with a list of file URLs based on user's selection. Storage Resource Manager takes over the file transfer and notifies the gateway after the offline data is *fetched* over. Gateway can query the status of the request and notify the user the readiness of data when they are *in place* for serving. The file access point at this stage will be presented as "on disk", as it should be.

The storage resource manager such as BeStMan brought in two important mechanisms. First, it provides dynamic space reservation and utilization. You can pre-allocate a chunk of disk space as temporary space for serving fetched data; and *unclogging* the storage space based on usage and access pattern. This is in essence, a form of *cache* management. The second mechanism is a plugin framework, where we can write customized plugin module to access the backend storage system. This capability is paramount for us to address the second challenging aspect of working with institutional HPC system: conforming to security and access constraints. As shown in Figure 4, we have designed and developed a *proxy module* at the backend. It polls periodically from SRM plugin for SRM data request, and invokes the backend transfer (hsi, a data transfer tool for HPSS). We leveraged GridFTP for high performance data movement. We are also exploring Globus Online to simplify the process of setup and configuration.

This architecture and solution to deep storage access method have a reaching impact on how scientists publish and disseminate their dataset, and how metadata is harvested as well, which is the subject of next section.

V. Data Population

As implied in Figure 2, the next major step in the data life cycle is for users to validate, organize, and subsequently, publish the data to some accessible component (e.g. web portal) for peer review. Figure 5 shows major flow and components of the current ESG publication framework design.

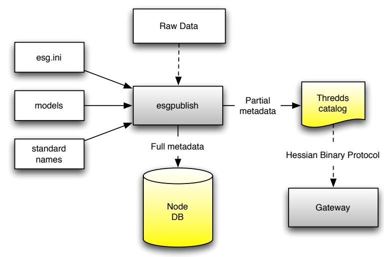

Fig. 5. Overview of Current Publisher Structure

At its core of this publication framework are *metadata harvesting* and *THREDDS catalog generation*. Metadata are first extracted and persisted into a local database from raw data (assumed to be placed in some standard format such as NetCDF). Some of this extracted metadata are output into a THREDDS catalog. These generated THREDDS catalogs are pushed to remote gateway for further indexing and ingestion, and presented through the web front end.

Revisiting HPC data lifecycle and viewing the problem from perspective of scientific users, we have the follow reflections and improvements to share on the framework design:

A. *Flexible data organization*: As an international collaborative project, CMIP5 showcased the design of controlled experiments with *preciseness*: it defines an extensive list of variables [3] described as *standard output*, results from extensive discussion from sub-discipline experts and interest groups within climate modeling community. Along with it, it defines Data Reference Syntax Document [13] for file name encoding, directory structure encoding, publication-level dataset id encoding, etc. This is necessary and a must for inter-model comparison. At the other end of the spectrum, we have scientific users who like to make use of ESG publisher and gateway except that they have a very "unstructured" layout with mutually agreed protocol among circle of colleagues for understanding the dataset. Publisher framework needs to accommodate the needs of both classes of users.

B. *Metadata extensibility*: We have discussed the scientific data diversity in Section III-C. In addition to well-defined data formats such as NetCDF and its CF extension, there are also a wide range of *ad-hoc* provenance data that scientists like to publish along with raw data. They are in the form of plain text, MS word, static generated plots and pages that describe the running environment, initial conditions, compiler version, extra tweaking, diagnostic results, which are critical to the understanding of the output and maybe necessary for reproducing the results. The publishing framework needs to be *open-ended* on both inputs and outputs.

C. *Ease of use*: we believe ease of use is paramount for publishing tool to support daily activities of scientific users. Being accustomed to utilizing the latest advances in technology, HPC users exhibit a greater level of tolerance for intuitive tooling support. However, as complicated as the supporting workflow might be, or as many as the constraints it must work within, we shall make every engineering effort to *make simple things simple, and make complicated things possible*. One of the proven techniques we can leverage is *convention over configuration* - i.e., make sensible default that cater to the particular working environment.

All of above issues are being tackled by the ORNL team. Next section will touch on our solutions on metadata extensibility as well as easy of use in the context of data indexing and search. Due to limited space, we will skip the rest of the design details.

VI. Data Indexing and Search

The final phase of the ESG data portal lifecycle illustrated in Section III is the data dissemination stage. This stage comprises several services that are vital for completion of the scientific data life cycle in HPC. In simplest terms, dissemination involves the explicit exchange of data being made available by data publishers and data being discoverable by clients. The manner in which this exchange occurs is relatively straightforward. Publishers make their data "discoverable" by publishing metadata descriptions about their datasets. This metadata is ingested by some common search index, where it may be made available to clients whom perform queries over the metadata to find specific datasets that may interest them. The clients are then free to interpret the data to build future experiments (perhaps re-entering ESG's HPC scientific data life cycle).

ESG has placed considerable time and effort in creating a robust data dissemination environment. In particular, we believe that our search and indexing architecture abides to the primary requirements that both data producers and data consumers demand – easy-to-use, intuitive tools to ensure that their proper participation in the workflow. We believe that data producers emphatically favor search indices that support a wide, diverse range of metadata ingestion. This eases the burden of making their data discoverable to potential interested consumers. Consumers, on the other hand, stress system usability. We outline both criteria in detail in the following subsections.

A. Addressing ESG Data Producer Requirements: Metadata diversity

One of the design artifacts of current publishing framework is its close coupling on modeling data.

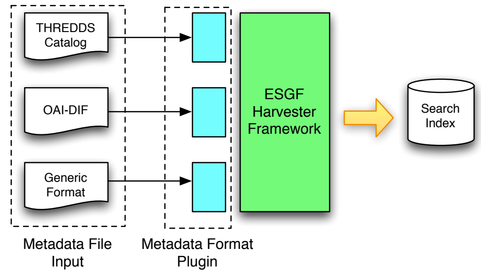

Fig. 6. Overview of Current Harvester Framework

We have designed a new metadata diversity framework that promotes a more positive metadata harvesting experience for data producers. One of the new additions is the adoption of the Apache Solr. Solr is an open-source search platform built on top of the Lucene search library and is used by many commercial enterprises worldwide.

Solr is a key component in our new harvesting infrastructure. Populating a Solr search index is relatively straightforward. Ingestion requires posting a document containing a series of key-value pairs representing the metadata of a dataset into a RESTful style interface. Although not immediately evident, this provides a valid solution to the transformation problem stated above. We are now able to construct a robust harvesting framework for transforming multiple data formats into the format Solr expects for ingestion. Let us refer to Figure 6, which gives an abstract view of the new harvesting environment. Metadata is fed to the *metadata handlers*, which manage how metadata can be parsed from its corresponding standard's structure. When the information is extracted, it is transformed into that key-value pair configuration that Solr expects. Only a trivial POST to Solr's HTTP interface is then required to fully ingest a metadata document.

The new ESG harvester currently supports a wide variety of formats that are commonly utilized by climate modellers and observational scientists alike. These formats include popular metadata structures, including THREDDS, OAI, CAS RDF, and FGDC. One of the greatest features of the new framework, however, is that it is fully extensible. Metadata handlers are essentially written as plugins to the harvesting framework (see Figure 6). Developers can design external modules that parse new metadata formats and plug it in to the existing architecture with relative ease.

B. Addressing ESG Data Consumer Requirements: Usability

In Section III-C, we presented a design that accommodated the data producer's requirements for metadata diversity support. The harvester framework provided data producers a simple way to be indexed and prepared for discovery in the ESG framework. Potential consumers of that data, however, are often concerned with the manner in which this harvested metadata is presented. Specifically, they are concerned with the *usability* of the harvested data's interface (i.e. the Gateway in the context of the ESG framework). An oft-studied concept, usability requires careful consideration in designing interfaces that are learnable, intuitive, and easy-to-use. To promote usability, we have altered the current Gateway scheme and believe that we have been able to improve ESG web usability by addressing two very relevant issues for current and future ESG data consumers.

To satisfy these requirements, we have added new components on the web front end that promote a comfortable search environment for a greater umbrella of users. Features like search box auto-completion and current search constraint listings aid in user comprehension. Our faceted-based category navigation tools have expanded to include terms that are non-biased. Moreover, we have built new geospatial and temporal range search tools, illustrated in Figures 7.

We also wanted to design a search environment that leverages the recent advances in web technology to promote efficiency. In this scheme, we create a single, intuitive search page, illustrated in Figure 7, that heavily utilizes the emerging ajax-style functionality. Data consumers need only visit *one composite page* where they pick and choose the component they would like to use to aid in their dataset search. Search components are encased in overlays, while search state is presented to the user in a clear and concise manner. This configuration has several inherent advantages. The overhead of the clumsy "back and forth" static page navigation is eliminated. Specific search constraints can be easily identified during the search process. Finally, consumers can add search requirements in parallel. For example, the user can perform a geospatial range search while simultaneously view facet values that are associated with that region, as Figure 7 illustrates.

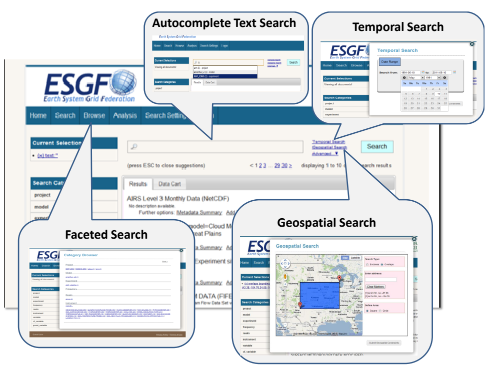

Fig. 7. Snapshot of Single Page View of the ESG Search Portal and components.

To summarize, our new search and index infrastructure has been designed to simplify both data producers and consumers interactions with the ESG system. Data producers now have a viable tool to publish metadata in their own customized formats to ESG, enhancing their discoverability. Data consumers may now navigate the ESG system in a more intuitive manner because of the various improvements in usability.

VII. Summary and Future Work

In this paper, we demonstrated that there are many aspects and intricacies that we must consider when designing data management strategies and solutions for ESG, especially in connection with supporting HPC environment. Specifically, we identified three key functional areas of the ESG data lifecycle – deep storage access, data publication, and data search and indexing – that particularly require dedicated attention and immediate optimal solutions. These areas all pose unique and interesting challenges that required us to stretch the limits and capabilities of modern HPC technology as well as truly understand how users interact with the ESG system. Moving forward, we would like to consider better integration on leveraging HPC as the data analysis engine, investigate scalable service infrastructure to accommodate both data diversity and data scale, and explore social aspects of such scientific data portal, with goals to make ESG both a platform as well as a truly indispensable tool that aid and support the end-to-end scientific discovery process.

References

[1] ESG: Earth System Grid Project. http://www.earthsystemgrid.org/, Accessed May 20, 2011.

[2] CMIP5: Coupled Model Intercomparison Project Phase 5. http://cmip-pcmdi.llnl.gov/cmip5/, Accessed May 25, 2011.

[3] CMIP5 requested parameters. http://cmip-pcmdi.llnl.gov/cmip5/docs/standard_output.pdf, Accessed On May, 2011.

[4] William Allcock, John Bresnahan, Rajkumar Kettimuthu, Michael Link, Catalin Dumitrescu, Ioan Raicu, and Ian Foster. The globus striped gridftp framework and server. *SC Conference*, 0:54, 2005.

[5] A.S. Bland, R.A. Kendall, D.B. Kothe, J.H. Rogers, and G.M. Shipman. Jaguar: The world's most powerful computer. *Memory (TB)*, 300(62):362.

[6] Ann Louise Chervenak. *Tertiary storage: an evaluation of new applications.* PhD thesis, University of California at Berkeley Berkeley, CA, USA, 1994.

[7] Ben Domenico, John Caron, Ethan Davis, Robb Kambic, and Stefano Nativi. Thematic real-time environmental distributed data services (thredds): Incorporating interactive analysis tools into nsdl. *Journal of Digital Information*, 2, 2002.

[8] Brian Eaton, Jonathan Gregory, Bob Drach, Karl Taylor, Steve Hankin, John Caron, Rich Signell, Phil Bentley, and Greg Rappa. Netcdf climate and forecast (cf) metadata conventions (version 1.4). http://cf-pcmdi.llnl.gov/documents/cf-conventions/1.4/cf-conventions.html, Feburary 2009.

[9] Ian Foster. Globus Online: Accelerating and Democratizing Science through Cloud-Based Services. *Internet Computing, IEEE*, 15(3):70 –73, may-june 2011.

[10] G. Shipman, D. Dillow, S. Oral, and F. Wang. The spider center wide file system: From concept to reality. In *Proceedings, Cray User Group (CUG) Conference, Atlanta, GA*, 2009.

[11] A. Shoshani, A. Sim, and J. Gu. Storage resource managers. *International Series in Operations Research and Management Science*, pages 321–340, 2003.

[12] Alex Sim, Mehmet Balman, Dean Williams, Arie Shoshani, and Vijaya Natarajan. Adaptive transfer adjustment in efficient bulk data transfer management for climate dataset. In *Parallel and Distributed Computing and Systems*, 2010.

[13] Karl E. Taylor, V. Balaji, Steve Hankin, Martin Juckes, Bryan Lawrence, and Stephen Pascoe. CMIP5 Data Reference Syntax (DRS) and Controlled Vocabulary, Version 1.2. http://cmip-pcmdi.llnl.gov/cmip5/docs/cmip5_data_reference_syntax.pdf, March 2011.

[14] Karl E. Taylor, Ronald J. Stouffer, and Gerald A. Meehl. A Summary of the CMIP5 Experiment Design. http://cmip-pcmdi.llnl.gov/cmip5/docs/Taylor_CMIP5_design.pdf, January 2011.

[15] Feiyi Wang, Sarp Oral, Galen Shipman, Oleg Drokin, Tom Wang, and Isaac Huang. Understanding Lustre Filesystem Internals. Technical Report ORNL/TM-2009/117, ORNL, 2009.