The NASA Center for Climate Simulation Data Management System

Toward an iRODS-Based Approach to Scientific Data Services

John L. Schnase[1], William P. Webster[1], Lynn A. Parnell[2], and Daniel Q. Duffy[2]
[1] Office of Computational and Information Science and Technology (CISTO)
[2] NASA Center for Climate Simulation (NCCS)
NASA Goddard Space Flight Center
Greenbelt, MD 20771

***Abstract*—The NASA Center for Climate Simulation (NCCS) plays a lead role in meeting the computational and data management requirements of climate modeling and data assimilation. Scientific data services are becoming an important part of the NCCS mission. The NCCS Data Management System (DMS) is a key element of NCCS's technological response to this expanding role. In DMS, we are using the Integrated Rule-Oriented Data System (iRODS) to combine disparate data collections into a federated platform upon which various data services can be implemented. Work to date has demonstrated the effectiveness of iRODS in managing a large-scale collection of observational data, managing model output data in a cloud computing context, and managing NCCS-hosted data products that are published through community-defined services such as the Earth System Grid (ESG). Plans call for staged operational adoption of iRODS in the NCCS.**

Keywords-iRODS, data services, archive management

I. Introduction

The NASA Center for Climate Simulation (NCCS) supports NASA modeling groups, funded investigators, and the extended climate research community by providing state-of-the-art high performance computing, networking, software, visualization, and data services. Its activities bring NASA observational and model products to various assessment activities carried out on behalf of national and international organizations, including the US Global Change Research Program and the Intergovernmental Panel on Climate Change (IPCC) [1].

Recognizing that large-scale scientific endeavors are becoming ever more data centric, we have begun to move toward a data services model for doing business. As part of this transition, over the past year, we have examined the potential of iRODS, the Integrated Rule-Oriented Data System, as a means of integrating and delivering scientific data services to the communities we serve. We call this effort the Data Management System project, and it has resulted in the NCCS Data Management System (DMS) — a testbed collection of independent iRODS data systems comprising observational and simulation data.

In the following sections, we describe our experiences with the DMS project, including motivating factors behind the effort, rationale for focusing on iRODS, early experiences with DMS, lessons learned, and our future plans regarding scientific data services in the NCCS.

II. Background

Modeling and assimilation are essential elements of climate research. They are the tools used to synthesize the diverse array of information from many satellites and bring that information to bear on improving prediction: weather and air quality, future climate change and its impacts, changes in atmospheric composition, or in terrestrial and marine ecosystems, important phenomena that contribute to climate variability such as changes in the water cycle, ocean circulation, or El Niño and its impacts. As a result, improvements in the accuracy of Earth science models are the end products of NASA research that most directly impact human society [2].

The NCCS plays a lead role in meeting the computational and data management challenges of modeling and assimilation. To accomplish its mission, the NCCS provides large-scale compute engines, analytics, data sharing, long-term storage, networking, and other high-end computing services designed to meet the specialized needs of the Earth science communities [1]. NCCS's Discover system is a scalable, commodity-based, open source cluster providing 30,000 computing cores, a peak speed of of 320 teraFLOPS, combined memory of 68 TB, access to 3 PB of disk storage, and access to a 16 PB robotic tape archive that is growing by 6 PB per year. The NCCS serves a customer base of over 635 scientists working on 160 projects.

The NCCS's mission is expanding to include a broader range of data and information services. First, the NCCS's two major customers, NASA's Global Modeling and Assimilation Office (GMAO) and the Goddard Institute for Space Studies (GISS) will be contributing products to the Intergovernmental Panel on Climate Change (IPCC) Fifth Assessment Report (AR5) [3, 4]. IPCC is coordinating a team of 831 climate change experts working throughout the world to produce AR5, which will be published between 2013 and 2014 [5]. These activities require that the NCCS provide the data management services and analytical tools necessary for GMAO and GISS to conduct their work and support the data publication requirements of the IPCC.

This work was funded in part by the American Recovery and Reinvestment Act of 2009.

Another requirement results from the tie that exists between NASA modeling efforts and satellite missions: observational data provide the means for evaluating and improving climate models. There is growing interest in bringing the climate modeling and observational communities together to work toward the goal of integrating model outputs and observational data [6]. Goddard Space Flight Center, being home to GMAO and many of NASA's Earth observing missions, is uniquely positioned to contribute to this effort, and these observational/ simulation data integration activities are becoming an important part of the NCCS's data services mission.

Finally, we recognize that computing requirements for Earth system modeling will increase significantly in the coming years [7]. We also recognize that high-end computing requires more than increased speed. Rapid access to large volumes of heterogeneous and geographically distributed data will be needed along with enhanced archiving capabilities, enhanced analysis capabilities, and the capacity to manage all aspects of high-performance scientific workflows. The NCCS must keep pace with innovations that can address these needs.

It is against this backdrop that the NCCS began looking at iRODS as a potential element in our technological and organizational response to changing demands.

III. The Integrated Rule-Oriented Data System

iRODS is an open source data grid software system being developed by the Data Intensive Cyber Environments (DICE) group at the University of North Carolina at Chapel Hill School of Information and Library Science [8]. It is described by its creators as peer-to-peer data grid middleware that provides a facility for collection-building, managing, querying, accessing, and preserving data in a distributed data grid framework. A key feature of iRODS is its capacity to apply policy-based control when performing these functions.

iRODS appealed to us for several reasons. It targets large repositories, large data objects, digital preservation, and integrated complex processing, making it one of the more promising technologies for grid-centric data services for scientific applications. We also liked the fact that its development culture has historic roots in digital libraries, persistent archives, and real-time data systems research, having received support from the National Science Foundation (NSF) and National Archives and Records Administration (NARA). Information about iRODS can be found in several places [9]; the description presented here is taken from a recent overview of the system [10]. iRODS provides the following capabilities:

- *Global persistent identifiers for naming digital objects*. A unique identifier is used for each object stored in iRODS. Replicas and versions of the same object share the same global identifier but differ in replication and version metadata.
- *Support for metadata to identify system-level physical properties of the stored data object*. The properties that are stored include physical resource location, path names (or canned SQL queries in the case of database resources), owner, creation time, modification time, expiration times, file types, access controls, file size, location of replicas, aggregation in a container, etc.
- *Support for descriptive metadata to enable discovery through simple query mechanisms*. iRODS supports metadata in terms of attribute-value-unit triplets. Any number of such associations can be added for each digital object.
- *Standard access mechanisms*. Interfaces include Web browsers, Unix shell commands, Python load libraries, Java, C library calls, FUSE-based file interface, WebDav, Kepler and Taverna workflow, etc.
- *Storage repository abstraction*. Files may be stored in multiple types of storage systems including tape systems, disk systems, databases, and, now, cloud storage.
- *Inter-realm authentication*. The authentication system provides secure access to remote storage systems including secure passwords and certificate-based authentication such as Grid Security Infrastructure (GSI).
- *Support for replication and synchronization of files between resource sites*.
- *Support for caching copies of files onto a local storage system and support for accessing files in an archive using compound resource methodology*. This includes the concept of multiple replicas of an object with distinct usage models. Archives are used as "safe" copies and caches are used for immediate access.
- *Support for physically aggregating files into tar-files to optimize management of large numbers of small files*.
- *Access controls and audit trails to control and track data usage*.
- *Support for execution of remote operations for data subsetting, metadata extraction, indexing, remote data movement, etc., using micro-services and rules*.
- *Support for rich I/O models for file ingestion and access including* in situ *registration of files into the system, inline transfer of small files, and parallel transfer for large files*.
- *Support for federation of data grids*. Two independently managed persistent archives can establish a defined level of trust for the exchange of materials and access in one or both directions. This concept is very useful for developing a full-fledged preservation environment with dark and light archives.

The iRODS data grid system consists of several components. It has a metadata catalog server, called the iCAT server, which provides the metadata and abstraction services for the whole data grid. There can be multiple resource servers that provide access to storage resources. A resource server (iRES) can provide access to more than one storage resource. The system can support any number of clients at a time. A client can connect to any server on the grid and request access to digital objects from the system. The request is parsed using the contextual and system information stored in the iCAT catalog, and a physical object is identified and transferred to the client. The request can be in terms of logical object names, or a conditional query based on descriptive and system metadata attributes. iRODS is a peer-to-peer server system; hence, requests can be made to any server, which in turn acts (brokers) on behalf of the client for transferring the file. The final file transfer takes the shortest path in terms of number of hops.

An important aspect of iRODS is its built-in rule framework. As part of each resource server, a distributed rule engine is implemented that provides extensibility and customizability by encoding server-side operations (including the main access APIs) into sequences of micro-services. The sequence of micro-services is controlled by user- or administrator-defined Event:Condition:Action-set:Recovery-set rules similar to those found in active databases. The rules can be viewed as defining pipelines or workflows. An ingestion or access process can be encoded as a rule to provide customized functionality. Rules also can be defined by users and executed interactively. Hence, changes to a particular process or policy can easily be constructed by the user, then tested and deployed without the aid of system administrators or application developers. The user also can define conditions when a rule gets triggered thus controlling application of different rules (or processing pipelines) based on current events and operating conditions.

The building blocks for the iRODS rules are micro-services — small, well-defined procedures or functions that perform a certain task. For example, one can use a rule that stipulates that when accessing a data object from a particular collection, additional authorization checks need to be made. These authorization checks can be encoded as a set of micro-services with different triggers that can fire based on current operating conditions. In this way, one can control access to sensitive data based on rules and can escalate or reduce authorization levels dynamically as the situation warrants. Apart from iRES servers and an iCAT server, iRODS also has two other servers: iSEC for scheduling and executing queued rules, and iXMS for providing a message-passing framework between micro-services.

IV. The NCCS Data Management System

The DMS project has been an effort to learn about iRODS and understand first hand if this technology might provide a comprehensive, end-to-end approach to managing data and data services in the NCCS.

Our strategy for gaining experience with iRODS has been to build four independent iRODS data systems comprising a range of data types and circumstances relevant to the NCCS. Two of the systems, merra_Zone and yotc_Zone, manage simulation data products; the other two, modis_Zone and isds_Zone, manage observational data products. Collectively, these form the basis of the NCCS Data Management System (DMS) [11]. Here we briefly describe two of the more interesting systems.

A. *modis_Zone: MODIS Earth Observational Data*

The goal in building the modis_Zone system was to gain experience using iRODS with a large, production observational data system. The modis_Zone provides experimental access to Moderate Resolution Imaging Spectroradiometer (MODIS) atmosphere data products. MODIS is a key instrument aboard the NASA Terra and Aqua satellites and is playing a vital role in the development of validated, global, interactive Earth system models that are able to predict global change accurately enough to assist policy makers in making sound decisions concerning the protection of our environment.

The creation of MODIS products is managed by the MODIS Adaptive Processing System (MODAPS). MODAPS generates Level 2 through Level 4 science products for distribution to the Goddard Earth Sciences Data and Information Services Center (GES DISC) for archival storage and to the MODIS science team for quality control. A web interface, MODAPS Web, is one method used by the MODIS science team to access these products.

The modis_Zone system, which was deployed on the operational MODAPS servers, consists primarily of the iRODS application, iRODS/iCAT database, and 22 storage nodes. The system was initially opened to MODIS users via the Filesystem in Userspace (FUSE) interface. With FUSE, the iRODS/iCAT database itself serves as the filesystem. As a result, database updates are reflected in the iRODS collection nearly instantaneously, thus eliminating the need for expensive consistency check jobs between filesystem and database.

The entire catalog of MODIS Atmosphere data products were registered in this exercise. The modis_Zone contains upwards of 54 million registered files, representing over 630TB of data with over 300 million defined metadata values across the collections. At the time of this writing, modis_Zone is the largest iRODS installation that has been built.

B. *merra_Zone: MERRA Climate Simulation Data*

The merra_Zone provides experimental access to Modern Era Retrospective-Analysis for Research and Applications (MERRA) model output data. Building the merra_Zone allowed us to gain experience using iRODS to manage simulation data products produced by the GMAO.

Retrospective-analyses (or reanalyses) have been a critical tool in studying weather and climate variability for the last 15 years. Reanalyses blend the continuity and breadth of output data of a numerical model with the constraint of vast quantities of observational data. The result is a long-term continuous data record. The MERRA project supports NASA's Earth science interests by using the NASA global data assimilation system to produce a long-term (1979-present) synthesis that places the current suite of research satellite observations in a climate data context. MERRA data can be accessed from the GES DISC through a variety of mechanisms, including OPenDAP, FTP, and GDS.

The entire catalog of monthly MERRA products was ingested for the purposes of the prototype merra_Zone system. This resulted in the ingestion of 360 files occupying 47 GB. During the ingest process, the MERRA data was registered with iRODS and stored on the filesystem. When each file was registered, standard iRODS-based metadata was stored in the iCAT along with MERRA-specific embedded metadata, which was was parsed and stored using routines we developed specifically for this purpose.

While working with the merra_Zone, we took the opportunity to build a prototype iRODS data system in the cloud using NASA Cloud Services [11].

NASA Cloud Services provide a progressive, efficient and highly-scalable containerized cloud computing infrastructure. Instead of procuring servers, software, data center space, and network equipment, users can stand up computing storage and virtualization instances in an accessible and affordable pay-as-you go environment. NASA Cloud Services enhance NASA’s ability to collaborate with external researchers by providing consistent tool sets and high-speed data connections. NAS-

Cloud Services are currently being used for education and public outreach, for collaboration and public input, and also for mission support. NASA Cloud Services are based on Amazon's EC2 cloud model. It provides Infrastructure as a Service (IaaS), which is an aspect of cloud computing that centers on the delivery of platform virtualization as an alternative to traditional data center installations.

In the DMS, we have essentially replicated the entire merra_Zone system in the cloud using a base Ubuntu image that we modified for our purposes. The merra_Zone image was paired with a 60 GB Elastic Block Store (EBS) volume, which is a model of decoupling physical storage volumes from instances in a modular way. This sets the stage for further examination of the roll of cloud computing in the NCCS.

V. DMS Explorations

Using the DMS as a testbed, we are looking at broader issues of data integration, federation, and generalized data services support. We are trying to understand how iRODS might affect our operations from an organizational perspective by considering the level and type of technical staffing required to support iRODS; the work involved in building, deploying, and maintaining iRODS systems; the work required to implement, test, and maintain domain- and NCCS-specific iRODS micro-services; how we might accommodate new and legacy data; customer impacts, including early adopter and end user support issues; and the financial, political, and cultural issues attached to a major new technology initiative such as this.

One of our first explorations dealt with federation. As described above, integrated access to heterogeneous data is a topic of increasing interest to us, particularly as it applies to coordinated access to observational data and climate model outputs. The iRODS federation mechanism provides one way of accomplishing this integration. It is a feature of iRODS in which separate iRODS zones, can be integrated. When two or more zones are federated, they share otherwise isolated data collections. By eliminating the need to explicitly switch an iRODS client between distinct instances, federation allows perusal or download of data from multiple iRODS systems through a single interface.

In the DMS, we have federated zones in order to understand the ramifications of managing disparate data in this context: an 'observational' zone consisting of MODIS data and a 'simulation' zone consisting of MERRA data. This federation essentially unites separate data collections while providing a single consolidated view to the collections.

A related challenge for us is finding a way to manage NCCS-hosted data products as independent collections in our archive while simultaneously supporting the delivery of those products through community-defined data services. A particular example of this is Earth System Grid (ESG). ESG is a data distribution system that integrates supercomputers with large-scale data and analysis servers located at various national labs and research centers throughout the world. It is the portal through which the Program for Climate Model Diagnosis and Intercomparison (PCMDI) at Lawrence Livermore National Laboratory (LLNL) is distributing data for the upcoming IPCC 5th Assessment Report [12].

The NCCS will participate as an ESG data node, which will be the primary mechanism for publishing GMAO and GISS contributions to AR5. In our setting, ESG becomes one of a collection of data services provided by the NCCS. However, while ESG may be well suited to the IPCC community's data publishing needs, ESG alone does not provide the full spectrum of archive management capabilities we need as a hosting data center. We need to manage the intermediate data products and processes relating to the pre-publishing tasks of IPCC research. We also need independent access to IPCC products for other uses and other customers, and we need device independence and the ability to annotate the data and otherwise manage these products as archived artifacts within the NCCS.

In the DMS, we are able to effectively register and publish iRODS-controlled data through the Earth System Grid using FUSE, thereby taking advantage of iRODS complete data lifecycle policy management capabilities (Fig. 1).

VI. Discussion

We learned several lessons in the course of the DMS project. Three topics in particular emerged as prominent factors in our assessment of iRODS: rule and micro-service development, namespace virtualization, and iCAT performance at production scale.

Policies and the mechanisms that implement them are at the heart of any data management framework. iRODS achieves its power and adaptability by using arbitrarily-defined server-side rules and micro-services to specify policies and mechanisms. Rules are expressed as Event:Condition:Action-set:Recovery-set statements. The actions taken by a rule are performed by micro-services — small, well-defined procedures (generally written in C, in our case written in Python) that perform various tasks relating to the data referenced by the rule. Server-side workflows are created by chaining rules together, and rules themselves can specify rules in a nested hierarchy. Changes to a policy or process can be made at any time with the addition of new rules.

We took a staged approach to ease our entry into the world of rule and micro-service development. As described above, we developed several custom extensions to handle data in the merra_Zone system. In both cases, we first scripted operations at a high level using iCommands. Only after prototyping and evaluating operations at this higher level did we attempt coding at the level of rules and micro-services.

The advantage of this approach is that high-level scripting allowed us to combine policies and mechanisms in a readable, easily understood context. It also allowed us to test our implementation using iRODS's robust suite of iCommands. The disadvantage, of course, is that these operations exist outside of the iRODS system and essentially function as custom interfaces, inaccessible to other iRODS clients, such as the Rich Web Browser. They also were inefficient, requiring numerous calls to iput, imeta, etc. to implement each operation.

The mapping of scripted functions to rules and micro-services was at times challenging. The design of the iRODS rule engine has been heavily influenced by logic programming, drawing on concepts such as recursive rule expression and forward rule chaining. It also builds on concepts from fields such as active databases, transactional systems, business rule systems, constraint management systems, service oriented architectures, and program verification. Its current state reflects both the power and complexity of this diverse inheritance, and it is

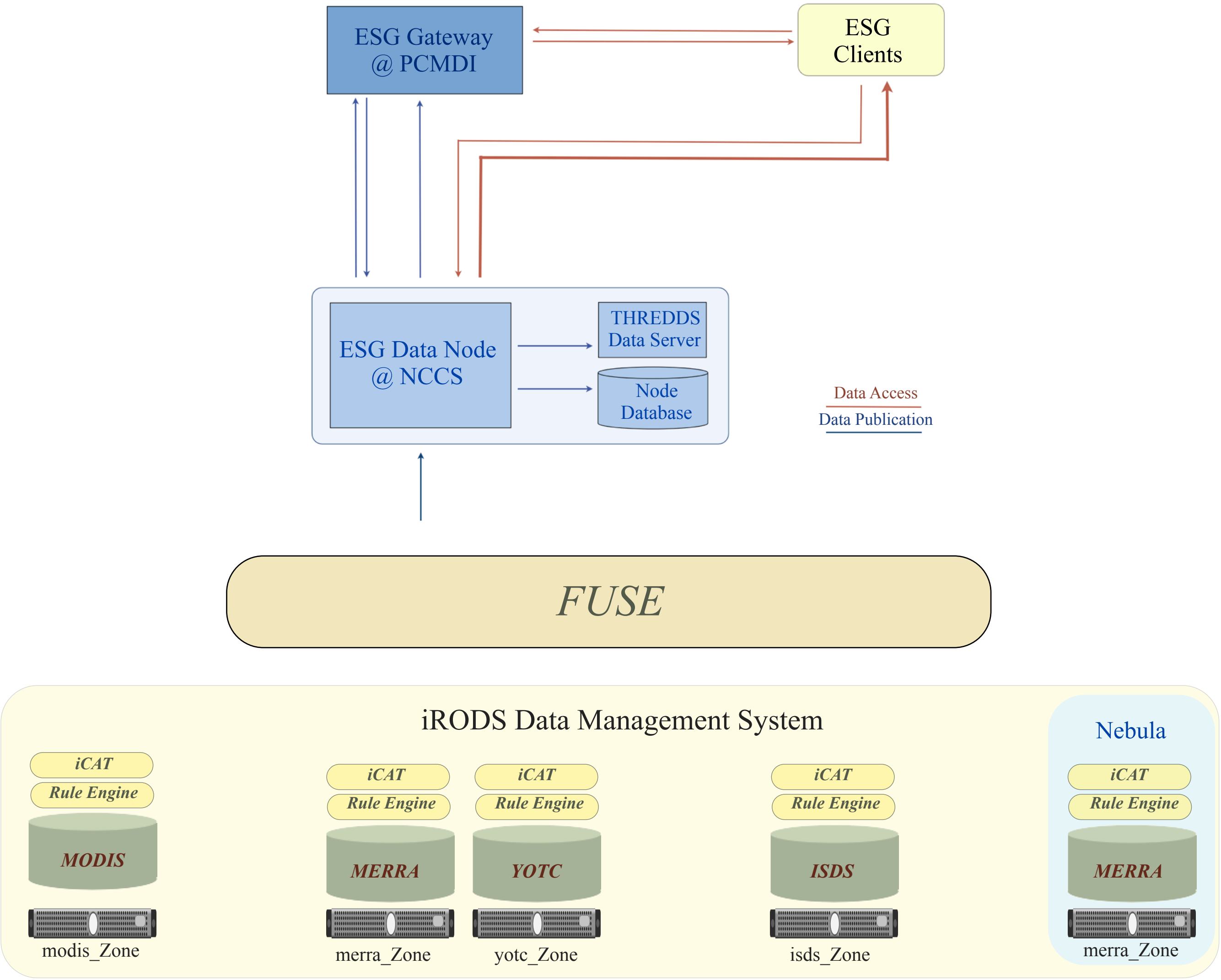

Figure 1. iRODS separates the management of storage resources from the way those resources are presented to users and applications. In the NCCS Data Management System, it allows Earth System Grid (ESG) to register and publish data from multiple resources through the iRODS FUSE mechanism.

probably fair to say that the syntax and semantics of the iRODS rule engine is continuing to evolve. Taken together, these factors make for a steep learning curve.

If there was a lesson to be learned here, it would be that overall risk — perceived or real — and general anxiety about a wholesale organizational commitment to iRODS is elevated by its rule engine complexity. One is unlikely to find experienced people to staff a new development effort, which implies that there may be a fairly long ramp-up as a new team hones is skills with iRODS. Once the investment is made in building a technical team, care must be taken to enable continuity of support should that team leave. Our efforts to build stable infrastructure around iRODS will depend on how well rule and micro-service development can be simplified and stabilized.

Virtualization is a key aspect of iRODS. The iRODS architecture decouples clients from dependencies on physical systems through multiple levels of abstraction, having users and applications interact with data through well-defined sets of logical namespaces. In dealing with this aspect of iRODS, we were reminded about the primacy of the filesystem in this domain: the vast majority of the storage and file manipulations performed by climate researchers relies on classic filesystem methods and constructs. Likewise, existing models, analysis tools, applications, and data services generally set atop POSIX-compliant filesystems.

This makes the iRODS FUSE mechanism particularly important to us, and FUSE filesystem virtualization appears to be the quickest path to integrating iRODS into existing NCCS processes. It provides a way to separate archive management from the idiosyncrasies of data services that need access to iRODS-managed collections, and it provides an interface and way of working that is familiar to our users.

Our experiences with modis_Zone, merra_Zone, and tests involving the Earth System Grid suggest that using FUSE for read-only delivery of data to existing filesystem-based services is fast enough to accommodate publication and distribution needs. Writing data to iRODS-managed collections through FUSE, however, is another matter. The write performance of user space filesystems is inherently and notoriously slow. Regardless of how effective FUSE is in the short term, the full effectiveness of iRODS will clearly require other mechanisms. We will need to help our customers become familiar with more direct access approaches to iRODS collections, such as the iCommands and Rich Web Browser, and perhaps even create interfaces tailored to their specific needs.

Another crucial element of an iRODS system is its metadata catalog, called the iCAT. iCATs store descriptive state information about the data objects in iRODS collections in an underlying DBMS, in our case, PostgreSQL. Since virtually every key interaction with an iRODS system involves the iCAT, understand its behavior and performance is of interest.

Our intent with modis_Zone was to gain iCAT experience with a large, production data collection. There are 54 million files in modis_Zone associated with over 300 million metadata values. Files were registered across 20 storage nodes at an initial rate of about 25 files per second (a little over two million files per day). After a million files, the process slowed to about one file per second. We were able to restore performance by creating multicolumn b-tree indexes for the iCAT.

After the registration process was complete, we experienced significant performance problems on searches, which could take as long as fifteen to twenty minutes over the entire collection. Here again, performance was improved by adding indexes. The one column indexes described above enabled imeta searches taking two seconds or less. Our impression is that the iRODS iCAT can deliver reasonable performance on the types of collections that we will manage, but that obtaining that performance will require straight-forward optimizations to tailor the iCAT to the local context.

VII. Conclusions

The intellectual content forming the basis of current and future climate research is being assembled in massive digital collections scattered through the world. As the size and complexity of these holdings increase, so do the complexities arising from interactions over the data, including those involving use, reuse, and repackaging for unanticipated uses, as well as managing over time the historical metadata that keep the data relevant.

The result is an increasing demand for software infrastructures that support data publication, data sharing, and data preservation. For scientific data centers especially, there also is a need for infrastructures that support a comprehensive, end-to-end approach to managing data: full information lifecycle management, encompassing all of stages of a scientific work process, from initial data acquisition to final data analysis [13]. Modeling and data assimilation are among the essential elements of climate research that are forcing this change.

Ultimately, we would like to be able to manage diverse collections within a uniform, coordinated environment. This environment — a production NCCS Data Management System — would be accessible through direct interfaces, would provide storage for analysis and visualization applications, and would be an integrative middle layer on which various data services, tailored to the needs of a diverse and growing customer base, could be built.

We have been impressed with the potential that iRODS offers for building an integrated capability such as this. This year, we will use the arrival of IPCC AR5 data as an opportunity to use iRODS in an operational setting. Specifically, we plan to develop the policies, mechanisms, rules, and micro-services to manage IPCC data in the NCCS archive and use the iRODS FUSE mechanism to accommodate Earth System Grid publication of GMAO and GISS AR5 data products.

Controlling risk as we adopt this technology is a concern, and the incremental approach described in this paper is one element of our mitigation strategy. Simplifying the approach to rule and micro-service development and support is also a high priority; we will be looking for opportunities to work with the iRODS development team and the iRODS community to foster greater use of Python, for example, for scientific applications such as our. The development of an archive administrators toolkit, or interface, tailored to the needs of the NCCS (and perhaps more generally to other climate data centers) also may become an element of our work program. Time permitting, further consideration of iRODS in the context of cloud computing will undoubtedly be part of our future work.

References

[1] NASA Center for Climate Simulation. htttp://www.nccs.nasa.gov.

[2] NASA Science Mission Directorate. 2010. *Responding to the Challenge of Climate and Environmental Change: NASA's Plan for a Climate-Centric Architecture for Earth Observations and Applications from Space*. http://science.nasa.gov/media/medialibrary/2010/07/01/Climate_Architecture_Final.pdf.

[3] NASA Global Modeling and Assimilation Office.http://gmao.gsfc.nasa.gov.

[4] NASA Goddard Institute for Space Studies. http://www.giss.nasa.gov.

[5] Intergovernmental Panel on Climate Change. http://www.ipcc.ch.

[6] Teixeira, J., Waliser, D., Crichton, D., Ferraro, R., Hyon, J., Gleckler, P., Taylor, K., Williams, D., Lee, T., Kaye, J., Maiden, M., Berrick, S. 2010. NASA Observations for the IPCC.http://www.clivar.org/organization/wgcm/wgcm-14/talks/061010/NASA_obs.pdf.

[7] NASA Science Mission Directorate. 2008. *Computational Modeling Capabilities Workshop Final Report*. http://www.hec.nasa.gov/worksho p08/final_report.pdf.

[8] Integrated Rule-Oriented Data System. http://www.irods.org.

[9] Moore, R.W., Rajasekar, A., and Marciano, R. (Eds.). 2010. *Proceedings of the iRODS User Group Meeting 2010: Policy-Based Data Management, Sharing, and Preservation*, (March 24-26, University of North Carolina, Chapel Hill, NC), 77 pp.

[10] Wan, M., Moore, R., and Rajasekar, A. 2009. Integration of cloud storage with data grids. In: *Proceedings of the Third International Conference on the Virtual Computing Initiative*, (October), 10 pp.

[11] Schnase, J.L., Tamkin, G., Fladung, D., Sinno, S., and Gill, R. 2011. Federated observational and simulation data in the NASA Center for Climate Simulation Data Management System Project. In: *Proceedings of the iRODS User Group Meeting 2011: Sustainable Policy-Based Data Management, Sharing, and Preservation*, (February 17-18, 2011, University of North Carolina, Chapel Hill, NC), 14 pp. In press.

[12] Earth System Grid. http://www.earthsystemgrid.org.

[13] Allen, R.B. 2010. *Management and Analysis of Large Scientific Data Sets*. http://ww.grids.ac.uk/NWGrid/LargeData.