High-Performance Remote Access to Climate Simulation Data: A Challenge Problem for Data Grid Technologies

Bill Allcock, Ian Foster, Veronika Nefedova

Mathematics and Computer Science Division, Argonne National Laboratory

Ann Chervenak, Ewa Deelman, Carl Kesselman

Information Sciences Institute, University of Southern California

Jason Lee, Alex Sim, Arie Shoshani

Lawrence Berkeley National Laboratory

Bob Drach, Dean Williams

Lawrence Livermore National Laboratory

Abstract

In numerous scientific disciplines, terabyte and soon petabyte-scale data collections are emerging as critical community resources. A new class of Data Grid infrastructure is required to support management, transport, distributed access to, and analysis of these datasets by potentially thousands of users. Researchers who face this challenge include the Climate Modeling community, which performs long-duration computations accompanied by frequent output of very large files that must be further analyzed. We describe the Earth System Grid prototype, which brings together advanced analysis, replica management, data transfer, request management, and other technologies to support high-performance, interactive analysis of replicated data. We present performance results that demonstrate our ability to manage the location and movement of large datasets from the user's desktop. We report on experiments conducted over SciNET at SC'2000, where we achieved peak performance of 1.55Gb/s and sustained performance of 512.9Mb/s for data transfers between Texas and California.

SC2001 November 2001, Denver (c) 2001 ACM 1-58113-293-X/01/0011 $5.00

1 Introduction

In numerous scientific disciplines, terabyte and soon petabyte-scale data collections are emerging as critical community resources. A new class of so-called Data Grid infrastructure is required to support management, transport, distributed access to, and analysis of these datasets by potentially thousands of users. One group of researchers that faces this challenge is the Climate Modeling community. Climate modeling is a challenging problem as extremely long-duration computations are accompanied by frequent output of very large files that are needed to analyze the simulated climate variability. The simulations then must be compared with what is known about the observed variability.

As the complexity and size of the simulations grow, the ability to generate model output threatens to outpace the ability to archive the output and to transport it from site to site. As an example, running a high-resolution ocean model on present-day computers with peak speeds in the 100-gigaflop range can generate a dozen multi-gigabyte files in a few hours at an average rate of about 2 MB/second. Computing a century of simulated time takes more than a month to complete and produces about 10 TB of archival output. Archival systems capable of storing hundreds of terabytes are required to support calculations of this scale on a regular basis. Moving to one-teraflop systems and beyond is expected to multiply these figures by a factor of ten or more, making Petabyte archives essential.

This tremendous volume of simulation data has the potential to revolutionize our understanding of complex climate processes. However, in order for this potential to be realized, geographically distributed teams of researchers must be able to effectively and rapidly develop new knowledge from these massive, distributed data holdings and share the results with a wider community. Fundamentally new methodologies for managing, accessing, recombining, analyzing and intercomparing distributed data are required if target research goals are to be realized. To address this critical problem, we established the *Earth System Grid* (ESG) project, with the goal of creating a next generation environment that harnesses the combined potential of massive distributed data resources, remote computation, and high-bandwidth wide-area networks as an integrated resource available to the research scientist.

In this article, we describe progress made to date in the ESG project. In particular, we describe the end-to-end functionality of our current ESG prototype, which integrates a number of advanced analysis and Data Grid technologies such as replica management, data transfer, request management, and others to support high-performance, interactive analysis of replicated data. As part of the prototype demonstration at the SC'2000 conference in November 2000, we conducted extensive usability and performance experiments, in which we demonstrated our ability to manage the location and movement of large datasets from the user's desktop. In addition to demonstrating the end-to-end functionality of the prototype, we evaluated the performance of one of its main components: our secure, efficient data transfer protocol. We focused on data transfer performance because transfers tend to dominate overall application performance for data-intensive applications such as climate modeling. We report on a national-scale experiment conducted over the SciNET network at the SC'2000, during which we achieved a peak performance of 1.55 Gb/s and sustained performance of 512.9 Mb/s for transfers of ESG climate model data between Dallas, Texas and Berkeley, California.

2 The Earth System Grid Prototype

R&D activities within the ESG project have been motivated by, and directed towards, the construction of an end-to-end ESG prototype that supports interactive access to and analysis of remote climate datasets. The essential elements of this prototype system we implemented are shown in Figure 1 and described briefly in the following; we provide details in subsequent sections. The main components of the ESG prototype are:

- *The Climate Data Analysis Tool (CDAT)* [1], developed within the Program for Climate Model Diagnosis and Intercomparison (PCMDI) group at LLNL, is a data analysis system that includes:

- *VCDAT*: an ESG-enabled climate model data browser and visualization tool
- A *metadata catalog* that is used to map specified attributes describing the data into logical file names that identify which simulation data set elements contain the data of interest

- A *request manager* developed at LBNL that is responsible for selecting from one or more replicas of the desired logical files and transferring the data back to the visualization system. The request manager uses the following components:
    - A *replica catalog* developed by the Globus project at ANL and ISI to map specified logical file names to one or more physical storage systems that contain replicas of the needed data
    - The *Network Weather Service (NWS)* developed at the University of Tennessee that provides network performance measurements and predictions. The request manager uses NWS information to select the replica of the desired data that is likely to provide the best transfer performance.
    - The *GridFTP* [2] data transfer protocol developed by the Globus project that provides efficient, reliable, secure data transfer in grid computing environments. The request manager initiates and monitors GridFTP data transfers.
    - A *hierarchical resource manager (HRM)* developed by LBNL is used on HPSS hierarchical storage platforms to manage the staging of data off tape to local disk before data transfer.

Once data transfer operations are complete, the desired data are visualized and presented to the user.

In the sections that follow, we describe the components of the ESG prototype in more detail.

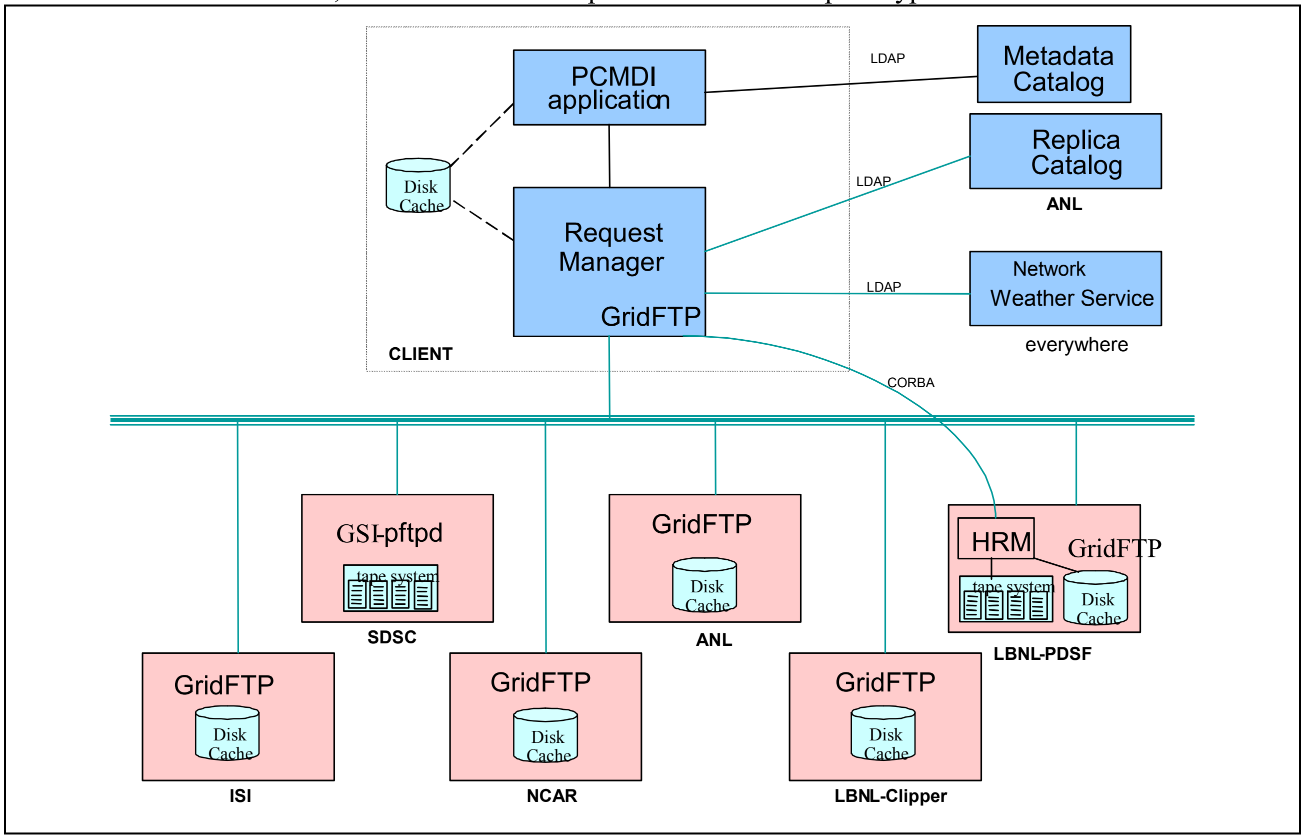


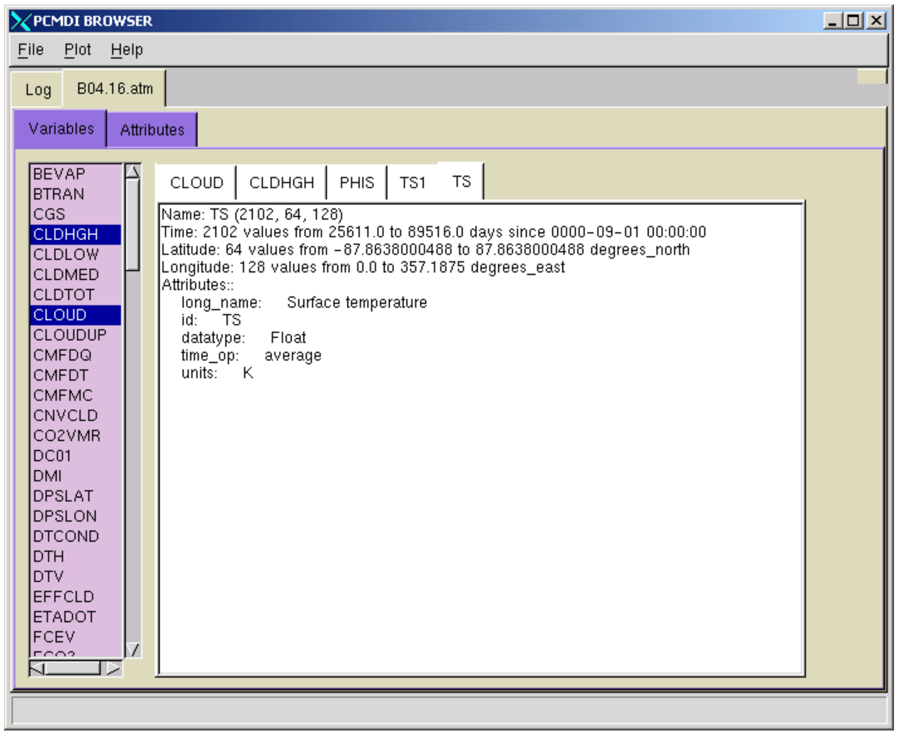
Figure 1: ESG-I Demonstration and Prototype Architecture

3 The Visualization and Computation System and Metadata Catalog

The *Climate Data Analysis Tool* (CDAT) [1], developed within the Program for Climate Model Diagnosis and Intercomparison (PCMDI) group at LLNL, is a data analysis system that includes an ESG-enabled climate model data browser (VCDAT). VCDAT allows users to specify interactively a set of attributes characterizing the dataset(s) of interest (e.g., model name, time, simulation variable) and the analysis that is to be performed.

CDAT includes the Climate Data Management System (CDMS), which supports a metadata catalog facility. Based on Lightweight Directory Access Protocol (LDAP), this catalog provides a view of data as a collection of datasets, comprised primarily of multidimensional data variables together with descriptive, textual data. A single dataset may consist of thousands of individual data files stored in a self-describing binary format such as netCDF. A dataset corresponds to a logical collection in the replica catalog described below. A CDAT client, such as the VCDAT data browser and visualization tool, contains the logic to query the metadata catalog and translate a dataset name, variable name, and spatiotemporal region into the logical file names stored in the replica catalog. Figure 2 shows a screen capture of the metadata catalog browser. It shows the initial selection of the desired variables (with each variable description). Once the user selects these variables, the prototype internally maps these variables onto desired logical file names. The use of logical rather than physical file names is essential to our use of data grid technologies, discussed below, because it allows us to localize data replication issues within a distinct replica catalog component.

The CDAT data analysis package uses the Python scripting language to provide a flexible system for analysis of climate model data. In our prototype, we have modified CDAT to access individual data files via the request manager. Analysis then proceeds in the client, as usual. In future work, we plan to extend this element of the ESG system to support server-side and/or client-side analysis. These enhancements will allow the climate community to manage the expected large amount of climate modeling data and process it in a distributed fashion, combining both local and remote testbed facilities into an easy-to-use computational facility.



```
Name: TS (2102, 64, 128)
Time: 2102 values from 25611.0 to 89516.0 days since 0000-09-01 00:00:00
Latitude: 64 values from -87.8638000488 to 87.8638000488 degrees_north
Longitude: 128 values from 0.0 to 357.1875 degrees_east
Attributes::
    long_name:    Surface temperature
    id:    TS
    datatype:    Float
    time_op:    average
    units:    K
```

Figure 2: Selection of Data by Specification of Application Attributes.

The CDAT system forwards the desired logical filenames to the request manager, which manages the data transfer as described in the next section. Once the data is available, VCDAT system performs the visualization. Example visualization output is shown in Figure 3.

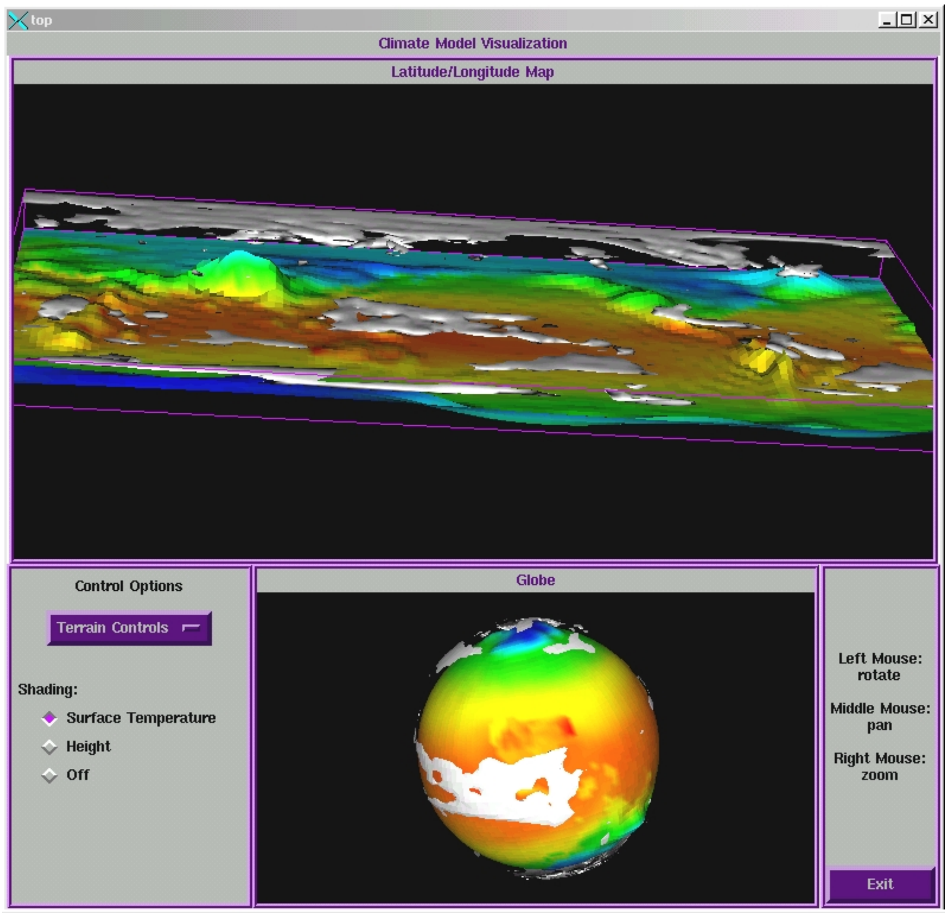


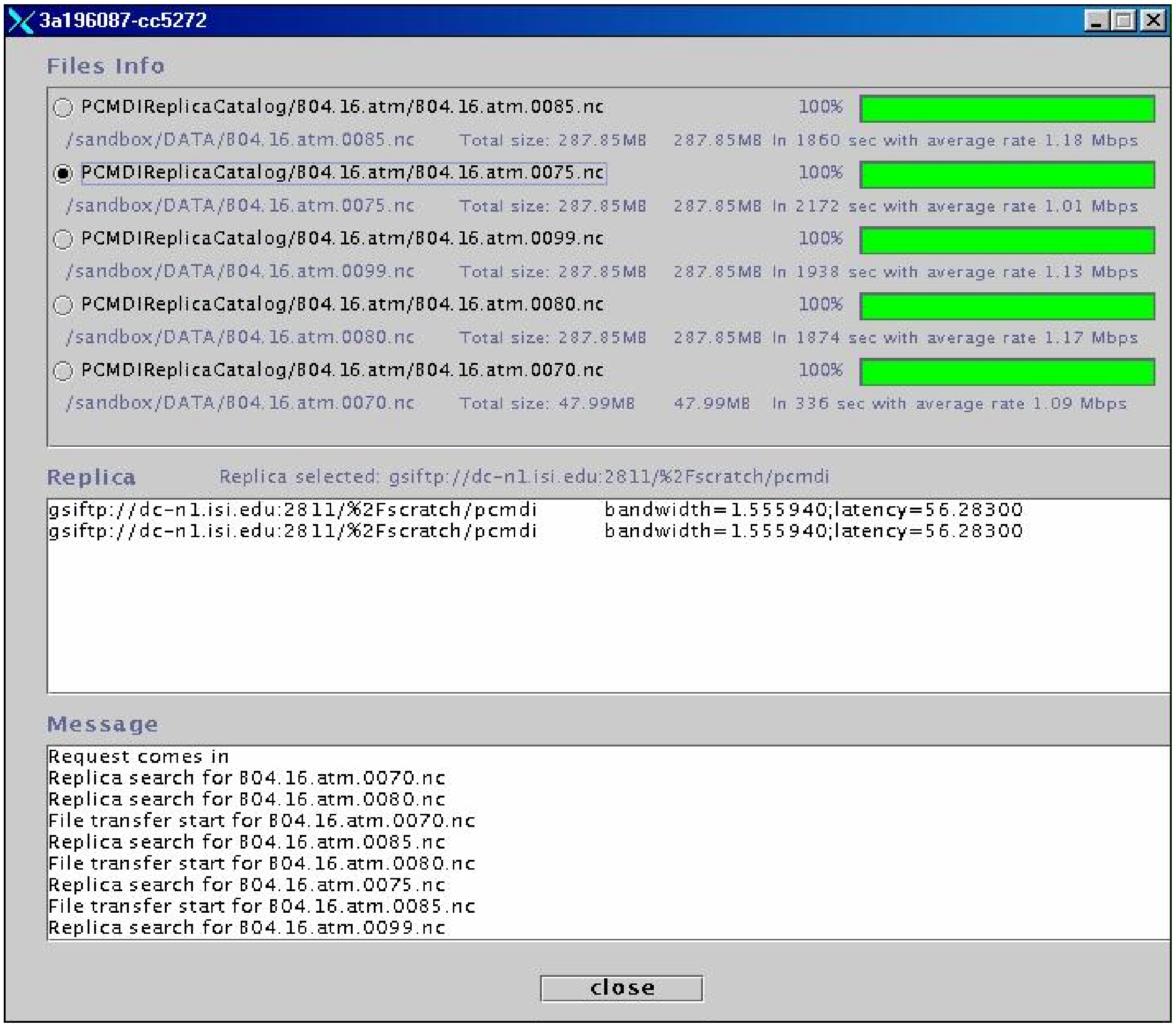
Figure 3: Visualization of Climate Data. Shown are Temperature (Color) and Clouds and Terrain (in 3D) .

4 The Request Manager

The Climate Data Analysis Tool (CDAT) forwards the list of desired logical files to the Request manager componend developed at LBNL. The Request Manager (RM) is a component designed to initiate, control and monitor multiple file transfers on behalf of multiple users concurrently. As explained above, the request manager in turn calls several underlying components: the Globus replica catalog, the Network Weather Service (via the MDS information service), the GridFTP data transfer protocol, and possibly the hierarchical resource manager (HRM).

The CDAT system calls the RM via a CORBA protocol that permits the specification of multiple logical files. For each file of each request, the multi-threaded RM opens a separate program thread. Each thread performs for the logical files assigned to it the following tasks: (1) it finds all replicas for the file from the Replica Catalog using an LDAP protocol; (2) for each replica it consults the NWS to determine the current transfer and latency from the site where the file resides to the local site; (3) it selects the "best" replica based on the NWS information; (4) it initiates a GridFTP "get" request to transfer the file; and (5) it monitors the progress of each file transfer by checking the file size of the file being transferred at the local site every few seconds.

The RM can communicate with various systems as needed to initiate a file transfer. As mentioned above, in this prototype the RM communicates with HRM for files stored on a mass storage system (MSS) that is not Grid enabled. HRM is a component that sits in front of the MSS (in this case an HPSS system at LBNL) and stages files from the MSS to its local disk cache. After this action is complete, the RM uses GridFTP to move the file securely over the wide-area network to its destination.



```
○ PCMDIReplicaCatalog/B04.16.atm/B04.16.atm.0085.nc                                  100%
  /sandbox/DATA/B04.16.atm.0085.nc    Total size: 287.85MB   287.85MB In 1860 sec with average rate 1.18 Mbps
● PCMDIReplicaCatalog/B04.16.atm/B04.16.atm.0075.nc                                  100%
  /sandbox/DATA/B04.16.atm.0075.nc    Total size: 287.85MB   287.85MB In 2172 sec with average rate 1.01 Mbps
○ PCMDIReplicaCatalog/B04.16.atm/B04.16.atm.0099.nc                                  100%
  /sandbox/DATA/B04.16.atm.0099.nc    Total size: 287.85MB   287.85MB In 1938 sec with average rate 1.13 Mbps
○ PCMDIReplicaCatalog/B04.16.atm/B04.16.atm.0080.nc                                  100%
  /sandbox/DATA/B04.16.atm.0080.nc    Total size: 287.85MB   287.85MB In 1874 sec with average rate 1.17 Mbps
○ PCMDIReplicaCatalog/B04.16.atm/B04.16.atm.0070.nc                                  100%
  /sandbox/DATA/B04.16.atm.0070.nc    Total size: 47.99MB    47.99MB  In 336 sec with average rate 1.09 Mbps

Replica selected: gsiftp://dc-n1.isi.edu:2811/%2Fscratch/pcmdi
gsiftp://dc-n1.isi.edu:2811/%2Fscratch/pcmdi      bandwidth=1.555940;latency=56.28300
gsiftp://dc-n1.isi.edu:2811/%2Fscratch/pcmdi      bandwidth=1.555940;latency=56.28300

Request comes in
Replica search for B04.16.atm.0070.nc
Replica search for B04.16.atm.0080.nc
File transfer start for B04.16.atm.0070.nc
Replica search for B04.16.atm.0085.nc
File transfer start for B04.16.atm.0080.nc
Replica search for B04.16.atm.0075.nc
File transfer start for B04.16.atm.0085.nc
Replica search for B04.16.atm.0099.nc
```

Figure 4: Dynamic monitoring of concurrent file transfers

We note that the ability to transfer multiple files from various sites concurrently can enhance the aggregate transfer rate to a client. Using this capability, one can choose to replicate popular collections in multiple sites. A RM can then plan concurrent file transfers to maximize the number of different sites from which files are obtained. After all the files of a request transfer successfully, the RM notifies CDAT. Since the transfer of large files can take many minutes, a transfer-monitoring tool was developed to show the status of the request transfer dynamically. Each file is monitored every few seconds as to its current size. This information as well as the total bytes transferred for all file requests are displayed on the client's screen. An example of such a screen is shown in the Figure 4. The top part of the screen shows for each file the amount transferred relative to the total file size. The middle part of the figure shows which replica locations have been selected based on the bandwidth and latency measurements provided by NWS. At the bottom of the screen, messages about the initiation of replica selection and file transfer for a selected replica are displayed.

5 The Network Weather Service

The request manager makes replica selection decisions based on network bandwidth and latency measurements and predictions that are supplied by the *Network Weather Service* (NWS), developed at the University of Tennessee [18]. NWS is a distributed system that periodically monitors and dynamically forecasts the performance that various network and computational resources can deliver over a given time interval; it forecasts process-to-process network performance (latency and bandwidth) and available CPU percentage for each machine that it monitors. The current implementation of the *request manager* selects the 'best' replica based on the highest bandwidth between the candidate replica and the destination of the data transfer. NWS information is accessed by the MDS information service [5].

6 The Globus Data Grid Toolkit

The request manager depends in particular on two components of the Globus data grid toolkit [4]: the replica catalog for replica location and GridFTP for secure, efficient transfer. (In addition, it uses the MDS information service for access to NWS information, and Grid Security Infrastructure for authentication.) The toolkit provides the enabling technology for wide area data management.

The data grid toolkit is based upon a Data Grid architecture [10], which provides a set of orthogonal, application-independent services that can then be combined and specialized in different ways to meet the needs of specific applications. We believe that this Data Grid infrastructure can usefully build on capabilities provided by the emerging Grid [8, 11], such as resource access, resource discovery, and authentication services. Our Globus Toolkit [6,9] provides a widely used instantiation of the lower layers of this Grid architecture.

A recently proposed architecture for Data Grids [4] includes four levels: fabric, connectivity, resource and collective. Figure 5 shows a partial list of components at each level of the proposed grid architecture. At the lowest fabric level of the architecture are the basic components that make up the Grid, including storage systems, networks and computational systems. In addition, the picture includes two catalogs: a metadata catalog such as the one provided by the CDAT system that contains descriptive information about files and a replica catalog where information is stored about registered replicas. At the connectivity layer are various standard protocols for communication and security. At the resource level are services associated with managing individual resources, for example, storage and catalog management protocols as well as protocols for network and computation resource management. GridFTP is placed here, as are the protocols used by the MDS information service and the GRAM protocol used for resource management. Finally, at the collective layer are higher-level services that manage multiple underlying resources, including services provided by the request manager such as reliable file transfer, replica selection and information services that provide resource discovery or performance estimation.

In the remainder of this section, we describe the two components of the Globus data grid toolkit used by the ESG prototype in more detail: (1) GridFTP: *a secure, reliable, efficient data transfer* service and (2) a replica catalog API, which provides the ability to *register, locate, and manage multiple copies* of datasets. These two services are used to construct a range of higher-level data management services, such as reliable creation of a copy of a large data collection at a new location and selection of the best replica for a data transfer based on performance estimates provide by information services [2,17].

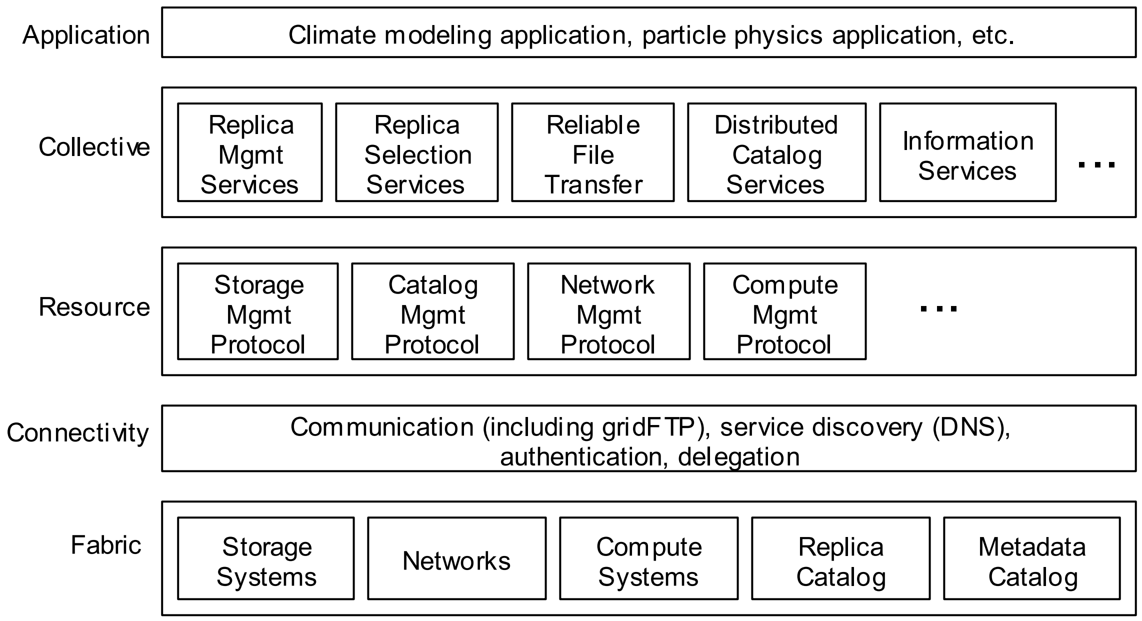


Figure 5: Partial list of elements of the Data Grid Reference Architecture [10].

6.1 GridFTP

The distributed nature of data sources and consumers means that data access and data transfer must be critical components of any data grid solution As such, the data transfer facilities need to be secure, fast, and reliable. The Globus Toolkit provides a protocol and API that meet these requirements. The protocol is called GridFTP.

GridFTP functionality includes features that are supported by the FTP standard, extensions that have already been standardized or are under consideration, and our own proposed extensions. We chose to extend the FTP protocol because we observed that FTP is the protocol most commonly used for data transfer on the Internet and the most likely candidate for meeting the Grid's needs. In addition, the FTP protocol provides a well-defined architecture for protocol extensions and supports dynamic discovery of the extensions supported by a particular implementation. Third, numerous groups have added extensions through the IETF, and some of these extensions are particularly useful in the Grid.

GridFTP provides the following features:

- Grid Security Infrastructure (GSI) [7] support for robust and flexible authentication, integrity, and confidentiality.
- Third-party control of data transfer that allows a user or application at one site to initiate, monitor and control a data transfer operation between two other sites.
- Parallel data transfer that uses multiple TCP streams between a source and destination, which can improve aggregate bandwidth in some situations [15].
- Striped data transfer that increases parallelism by allowing data to be striped across multiple hosts. Striping can be combined with parallelism to have multiple TCP streams between each pair of hosts.
- Server side processing that allows for the inclusion of user written code that can process the data prior to transmission or storage. Partial file retrieval is included by default.

- Support for automatic negotiation or manual setting of TCP buffer/window sizes that can dramatically improve data transfer performance.
- Support for reliable and restartable data transfer, to handle failures such as transient network and server outages. Also includes support for user-written fault recovery algorithms.

The motivation for GridFTP is to provide a uniform interface to various storage systems. Such systems include hierarchical storage systems, disk caches and storage brokers. Typically, these storage systems have incompatible data access protocols. These incompatible protocols effectively partition the datasets available on the grid: applications must either choose only a subset of storage systems or must use multiple methods to retrieve data.

Our first approach to dealing with these incompatible protocols was to design a layered client or gateway that would present the user with one interface to these heterogeneous storage systems. The gateway approach was attractive because it would present a single interface to applications and would not require changes to storage systems. However, our experience with the gateway approach exposed several disadvantages. First, performance suffered due to costly translations between the layered client and storage system-specific client libraries and protocols. In addition, efficient transfer of datasets between storage systems was challenging. Second, building a client or gateway that supports numerous storage systems was complex. This was exacerbated by the need to provide support for multiple client languages, such as C/C++, Java, Perl, Python, shells, etc. Finally, maintaining the client or gateway as each storage system evolved independently would have been a daunting task.

Based on our experience with the layered client approach, we decided to address the issue of incompatible data access protocols directly by proposing GridFTP as a common data transfer and access protocol. GridFTP is designed to provide a superset of the features offered by the various Grid storage systems currently in use. We believe that it will be mutually advantageous to both storage providers and users to have a common but extensible underlying data transfer protocol that provides interoperability between disparate storage systems. Storage providers would gain a broader user base, because their data would be available to any client. Storage users would gain secure, efficient access to a broader range of storage systems and data.

6.2 Replica Management

In a data grid environment that supports the management of, and distributed access to, huge data sets by thousands of researchers, management of replicated data is an important function. The Globus data grid toolkit provides a layered replica management architecture. At the lowest level is a *Replica Catalog* that allows users to register files as logical collections and provides mappings between logical names for files and collections and the storage system locations of file replicas. Building on this basic component, we provide a low-level API that performs catalog. This API can be used by higher-level tools such as the request manager that select among replicas based on network or storage system performance.

The Replica Catalog provides simple mappings between logical names for files or collections and one or more copies of those objects on physical storage systems. The catalog registers three types of *entries*: logical collections, locations, and logical files. A *logical collection* is a user-defined group of files. We expect that users will often find it convenient and intuitive to register and manipulate groups of files as a collection, rather than requiring that every file be registered and manipulated individually. *Location* entries in the replica catalog contain the information required for mapping a logical collection to a particular physical instance of that collection. Each location entry represents a complete or partial copy of a logical collection on a storage system. The replica catalog also includes optional entries that describe individual *logical files*. Logical files are entities with globally unique names that may have one or more physical instances. Based on our early experiences with the replica catalog, we chose to make logical file entries optional to improve catalog scalability for large collections.

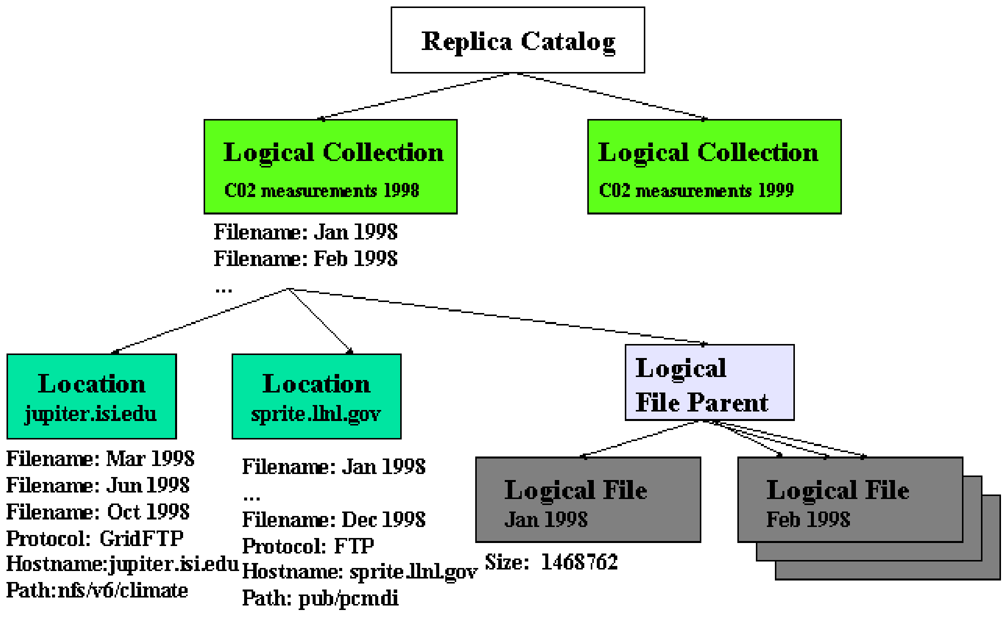


Figure 6: A Replica Catalog for a climate modeling application.

Figure 6 shows an example replica catalog for a climate modeling application, such as the one described in the introduction. This catalog contains two logical collections with CO_2 measurements for 1998 and 1999. The 1998 collection has two physical locations, a partial collection on the host jupiter.isi.edu and a complete collection on sprite.llnl.gov. The location entries contain attributes that list all files stored at a particular physical location. They also contain attributes that provide all information (protocol, hostname, port, path) required to map from logical names for files to URLs corresponding to file locations on the storage system. The example catalog also contains logical file entries for each file in the collection. These entries provide size information for individual files.

Current design effort for the replica catalog is focused on support for distribution and replication of the catalog and for more flexible mappings between logical and physical file names.

7 Experimental Studies over a Wide-Area Network

In November 2000, we took advantage of the excellent connectivity at SC 2000 to obtain WAN performance data for the ESG prototype and our data grid infrastructure.

In these experiments, we demonstrated the end-to-end functionality of the ESG prototype by performing visualizations of climate attributes such as precipitation and cloud cover using data sets that were distributed over several locations around the United States, including LBNL, LLNL, ISI, ANL and NCAR. First, we selected parameters to be visualized using the user interface shown in Figure 2, as well as additional interfaces for selecting the data set to be visualized and the range of time over which to visualize data. Based on the selection of these parameters, the CDAT system consulted its metadata database and identified the logical files of interest. The CDAT system passed these logical file names to the request manager, which performed replica selection and initiated gridFTP data transfers of the desired files. Figure 4 shows the request manager's monitor of the progress of data transfers. Once data transfer was complete, the CDAT system analyzed and visualized the desired data, producing output as shown in Figure 3.

We also ran a series of experiments to test the performance of wide-area gridFTP data transfers. In the remainder of this section we present peak and sustained wide-area bandwidth measurements and demonstrate the effectiveness of some of the fault-tolerant features of gridFTP. We focus on data transfer measurements since transfer costs dominate the performance of data-intensive applications such as climate modeling.

Figure 7 shows the network connectivity that was available during the convention. Our hardware configuration for this testing consisted of eight Linux workstations, in Dallas, Texas, sending data across the wide area network to eight workstations (four Linux, four Solaris), at Lawrence Berkeley National Laboratory in California. All workstations had gigabit Ethernet NICs and the cluster switches were connected via dual bonded gigabit Ethernet to the exit routers. Wide area network traffic when through the nationwide HSCC (High Speed Connectivity Consortium) and NTON (National Transparent Optical Network) infrastructure provided by Qwest Communications, and finally across an OC48 connection to Lawrence Berkeley National Laboratory.

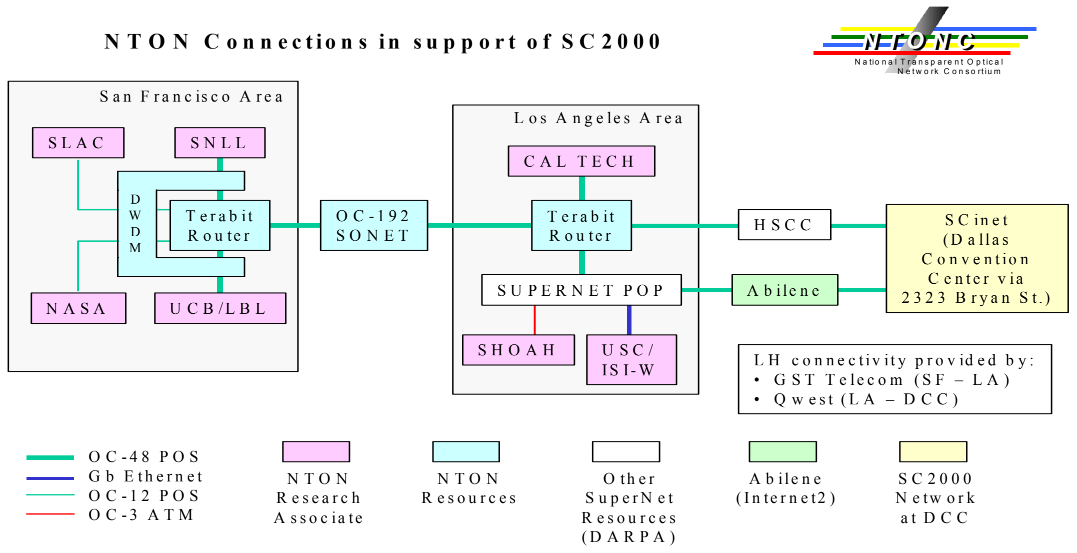


Figure 7: Graph From NTONC Showing Wide Area Network Connectivity During SC'00 [14].

We used *striped* transfers, with a 2-gigabyte file partitioned across the eight workstations. Each workstation actually had four copies of its file partition. On each server machine, a new transfer of a copy of the file partition was initiated after 25% of the previous transfer was complete. Each new transfer created a new TCP stream. At any time, there were up to four simultaneous TCP streams transferring data from each server in the cluster of eight workstations, for a total of up to 32 simultaneous TCP streams.

Proper TCP buffer sizes are critical to obtaining good performance in TCP wide area links [16]. The appropriate size is determined by calculating the bandwidth-delay product as follows: Buffer size in KB = Bandwidth (Mbs) * Latency (ms) * 1024/1000/8 (constants are for units). This value represents the maximum amount of data that could be in flight "in the pipe" over the network. For our purposes, latencies were in the 10-20 ms range, and we expected bandwidth to be between 200 Mbs and (an optimistic) 500 Mbs. We chose 1 MB as a reasonable buffer size for our transfers.

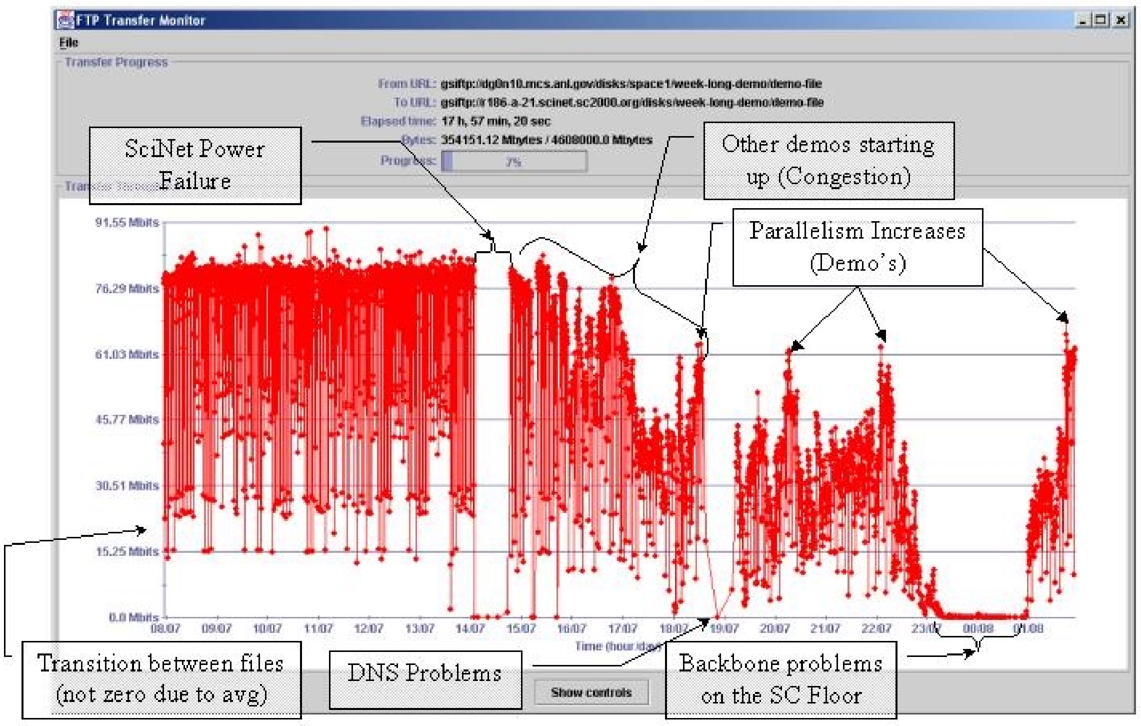
Any system can only be as fast as its slowest link. We used multiple disks with software RAID to ensure that disk was not the bottleneck. The NICs were rated at 1 Gbs, and the Network at 2.5 Gbs (although we were only supposed to use 1.5 Gbs). This two remaining bottlenecks are potentially the CPU or the PCI bus. Earlier work had shown (and the pattern repeated itself here) that the CPU was running at near 100%

capacity. This high CPU usage is common with Gigabit Ethernet and is caused by the numerous interrupts that must be serviced. Interrupt coalescing (waiting to see if more interrupts come in and combining them) can help reduce this problem. We were, in fact, using interrupt coalescing at SC. A second way of reducing the CPU load is by using Jumbo Frames. If each interrupt is able to dispatch more data, fewer interrupts are required. However, one of the routers did not support jumbo frames, so we were unable to evaluate the impact of this mechanism.

Table 1 summarizes the results of our testing. Instrumentation provided by SciNET revealed that we achieved a peak transfer rate of 1.55 Gbs over an interval of 0.1 seconds and 1.03 Gbs over an interval of 5 seconds. Over a one hour-long period, we sustained an average data rate of 512.9 Mbs per second. This corresponded to a total data transfer during that hour of 230.8 GB.

Table 1: Configuration and performance results.

Striped servers at source location	8
Striped servers at destination location	8
Maximum simultaneous TCP streams per server	4
Maximum simultaneous TCP streams overall	32
Peak transfer rate over 0.1 seconds	1.55 Gbits/sec
Peak transfer rate over 5 seconds	1.03 Gbits/sec
Sustained transfer rate over 1 hour	512.9 Mbits/sec
Total data transferred in 1 hour	230.8 Gbytes



From URL: gsiftp://dg0n10.mcs.anl.gov/disks/space1/week-long-demo/demo-file
To URL: gsiftp://r186-a-21.scinet.sc2000.org/disks/week-long-demo/demo-file
Elapsed time: 17 h, 57 min, 20 sec
Bytes: 354151.12 Mbytes / 4608000.0 Mbytes
Progress: 7%

Figure 8: Bandwidth measured for a series of transfers performed over a 14 hour period, between Dallas and Chicago.

Experiments were also conducted to test the fault tolerance features of GridFTP. A reliability plug-in was written that monitored performance and if data transfer rates dropped below a certain, user configurable, point, an alternate replica would be selected. The hardware configuration for this experiment consisted of a Linux workstation with a 100 Mbs NIC transferring a 2 GB file repeatedly to a similar workstation at Argonne National Laboratory in Chicago, via commodity internet access. Figure 8 shows the results of our experiment. We show aggregate parallel bandwidth for a period of approximately fourteen hours on November 7, 2000. This data corresponds to parallel (multiple TCP stream) transfers using varying levels of parallelism, up to a maximum of eight streams. The graph was produced with the NetLogger system [13]. Bandwidth between the two hosts reaches approximately 80 Mbs, somewhat lower than achieved in previous experiments, most likely due to disk bandwidth limitations. Figure 8 shows drops in performance due to various network problems, including a power failure for the SC network (SCiNet), DNS problems, and backbone problems on the exhibition floor. Because the GridFTP protocol supports restart of failed transfers, the interrupted transfers continued as soon as the network was restored.

Toward the right side of the graph, we see several temporary increases in aggregate bandwidth, due to increased levels of parallelism. The frequent drop in bandwidth to relatively low levels occurs because the GridFTP implementation used at SC'2000 destroys and rebuilds its TCP connections between consecutive transfers. Based on this observation, we identified the need for and have since implemented *data channel caching*. This mechanism allows a client to indicate that a TCP stream is likely to be re-used soon after the existing transfer completes. In response to this input from the client, we temporarily keep the TCP channel active and allow subsequent transfers to use the channel without requiring costly breakdown, restart, and re-authentication operations.

Another feature that was added to GridFTP after SC'2000 is 64-bit addressing to allow file sizes greater 2 gigabytes. Lack of support for large files limited the bandwidth we achieved at SC'2000. Large files will be common in data grid environments and must be supported efficiently by GridFTP.

8 Related Work

DODS, the Distributed Oceanographic Data System [12], has focused on the complementary problem of providing remote access to a data file. DODS has a multi-tier client/server architecture. On the client side, DODS APIs may be linked into an application, thus enabling it to access remote data via URL. DODS servers provide filters for a number of different data formats (e.g. netCDF, HDF, JGOFS, MATLAB, and Freeform) that provide subsetting and format translation of managed data in response to client requests. DODS was designed with a heavy emphasis on generality and relies solely upon HTTP as a transport protocol. While this approach facilitates easy deployment, it is not well-suited to HPC applications or very large data movement over high-bandwidth wide-area networks. In addition, DODS does not currently address wide-area security requirements, replica management, access to secondary storage, or distributed catalog functions.

The Storage Resource Broker (SRB) [3] from the San Diego Supercomputing Center is middleware infrastructure that provides a uniform, UNIX-style file I/O interface for accessing heterogeneous storage resources distributed over the wide area network. Using its Metadata Catalog (MCAT), SRB provides collection-based access to data based on high-level attributes rather than on physical filenames. SRB also supports automatic replication of files on storage systems controlled by SRB. In contrast to the layered Globus architecture with direct user and application control over replication, SRB uses an integrated architecture, with all access to data via the SRB interface and MCAT and with SRB control over replication and replica selection.

9 Conclusions and Future Directions

We have described the ESG prototype, which provides a complete, end-to-end implementation of an interactive analysis and visualization system for data replicated in a wide area data grid. This prototype has been used to demonstrate a number of innovative features, including interactive analysis of remote data, performance data-based replica selection, and high-performance wide area data movement.

We are now starting work, in collaboration with the National Center for Atmospheric Research, on ESG-II, a next-generation system that supports (1) distribution of data analysis and visualization pipelines, so that some data analysis operations (at least extraction and subsetting, similar to those available with DODS) can be performed local to the data before it is transferred over the network; (2) more sophisticated discovery capabilities; and (3) access to data and analysis capabilities from lightweight clients such as browsers, and portals—including access via DODS protocols and mechanisms.

Acknowledgements

This research was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. This work was also supported by the Director, Office of Laboratory Policy and Infrastructure Management of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098. This research used resources of the National Energy Research Scientific Computing (NERSC), which is supported by the Office of Science of the U. S. Department of Energy. This work was also performed under the auspices of the U.S. Department of Energy by the University of California at Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

References:

[1] "Climate Data Analysis Tool," http://www.pcmdi.llnl.gov/software/cdat/index.html.

[2] W. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke, "Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing," presented at Mass Storage Conference, 2001.

[3] C. Baru, R. Moore, A. Rajasekar, and M. Wan, "The SDSC Storage Resource Broker," presented at Proc. CASCON'98 Conference, 1998.

[4] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, "The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Data Sets," *J. Network and Computer Applications*, pp. 187-200, 2001.

[5] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid Information Services for Distributed Resource Sharing," presented at IEEE International Symposium on High Performance Distributed Computing, 2001.

[6] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *International Journal of Supercomputer Applications*, vol. 11, pp. 115-128, 1997.

[7] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A Security Architecture for Computational Grids," in *ACM Conference on Computers and Security*, 1998, pp. 83-91.

[8] I. Foster and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure,".: Morgan Kaufmann, 1999.

[9] I. Foster and C. Kesselman, "Globus: A Toolkit-Based Grid Architecture," in *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, Eds.: Morgan Kaufmann, 1999, pp. 259-278.

[10] I. Foster and C. Kesselman, "A Data Grid Reference Architecture," GriPhyN 2001-6, 2001.

[11] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Intl. J. Supercomputer Applications*, vol. (to appear), 2001.

[12] P. A. Fox, J. Garcia, and P. Kellogg, "The HAO Data Service: Experience in Interdisciplinary Data Delivery," presented at Proc. of the CODATA 2000 Workshop, US National Academy, 2000.

[13] D. Gunter, B. Tierney, B. Crowley, M. Holding, and J. Lee, "NetLogger: a toolkit for distributed system performance analysis.," presented at 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2000.

[14] NTONC, "NTON Connection in support of SC2000," http://www.ntonc.org/docs/NTON_ConnectionsForSC2000v1.1.ppt, 2000.

[15] L. Qiu, Y. Zhang, and S. Keshav, "On Individual and Aggregate TCP Performance," presented at 7th Intl. Conference on Network Protocols (ICNP'99), Toronto, Canada, 1999.

[16] B. Tierney, "TCP Tuning Guide for Distributed Applications on Wide Area Networks," presented at Usenix ;login, 2001.

[17] S. Vazhkudai, S. Tuecke, and I. Foster, "Replica Selection in the Globus Data Grid," presented at International Workshop on Data Models and Databases on Clusters and the Grid (DataGrid 2001), 2001.

[18] R. Wolski, "Forecasting Network Performance to Support Dynamic Scheduling Using the Network Weather Service," in *Proc. 6th IEEE Symp. on High Performance Distributed Computing*. Portland, Oregon, 1997.