Journal of Computer and Communications, 2014, 2, 1-6
Published Online April 2014 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2014.26001

Note on Three Classes of Data Grid Operations

Arcot Rajasekar, Hao Xu, Reagan Moore

School of Information and Library Science, University of North Carolina at Chapel Hill, Chapel Hill, USA
Email: sekar@renci.org, xuh@cs.unc.edu, rwmoore@renci.org

Received 19 January 2014; revised 25 February 2014; accepted 5 March 2014

Abstract

Traditional grid computing focuses on the movement of data to compute resources and the management of large scale simulations. Data grid computing focuses on moving the operations to the storage location and on operations on data collections. We present three types of data grid operations that facilitate data driven research: the manipulation of time series data, the reproducible execution of workflows, and the mapping of data access to software-defined networks. These data grid operations have been implemented as operations on collections within the NSF DataNet Federation Consortium project. The operations can be applied at the remote resource where data are stored, improving the ability of researchers to interact with large collections.

Keywords

Component, Data Grid, Policy-Based, Operations

1. Introduction

Given the increasing amount of data, diversity of data types, diversity of storage systems, and emergence of software defined networks, computing with data needs new approaches. Traditional analysis models that rely on scheduling of jobs, reservation of time on super computers, and allocations of compute cycles need to be augmented with novel models of computation that handle the integration with large-scale data management.

Scientific workflows that integrate multiple processing steps are increasingly applied in current computational environments, as opposed to single job submission to a cluster or a super computer. Several domains from medicine and biology, to hydrology and ecology, to finance and e-commerce, to social networking and crowd sourcing need a paradigm shift to enable their data intensive analyses. Traditional, large monolithic blocks of code scheduled for days of operation on a cluster or a super computer rely on large static directories of data pre-staged to the computing sites. Emerging data intensive analyses are agile, with a short-lifetime, and sharded

How to cite this paper: Rajasekar, A., Xu, H. and Moore, R. (2014) Note on Three Classes of Data Grid Operations. *Journal of Computer and Communications*, **2**, 1-6. http://dx.doi.org/10.4236/jcc.2014.26001

across geographically distributed sites. They perform a chain of processing steps, often tweaked at run time. Such computations are highly data aware with computable agents distributed across the network to where the data are located, choreographed by distributed scientific workflow systems with fail-over capabilities.

A second aspect that is coming to the front in these novel settings, is the diversity of the characteristics of the data. Big data, characterized by the 5Vs—Volume, Velocity, Veracity, Variety and Value—become the challenges of this realm. Most often, the data are non-static, and are not resident as files on a file system. Increasingly, data are processed through an I/O stream, be it from twitter or stock market feeds, or from environmental or medical sensors, or from action sequences coming from High Definition movies taken from marine or astronomical cameras, or from human interactions with search engines at e-commerce sites. The timeliness of data mining in such settings needs computation done through multiple processing steps, guided through data flow computations. Dealing with time-series data analysis poses challenges not seen by static file analysis models of the compute grid paradigm.

A third aspect of data that is emerging is its decentralization. Data are dispersed across the communication network, accessible through heterogeneous protocols, stored under autonomous administrative management, controlled by privacy, security and authorization policies, and replicated and backed up in multiple locations. Performing computation in such a milieu needs an understanding of the whole network as a "computer". Computation has to be dispersed to the locality of the data. Data need to be streamed from sources to analysis stations to archives, through source-to sink streamlined parallel data transfers. Sharded, torrential data transfers through strategic caching and staging facilities are needed to achieve optimal performance in this global computational platform. A new look at how to share knowledge across the computing, storage, application and service systems through unifying networking, identification systems, and authorization systems is very much needed. Increasing computational awareness between the network world and the data management world (in both directions) is a shift in paradigm that is being spearheaded by emerging software defined networks.

The three novel aspects—workflows, time-series analysis, and network-enabled data intensive computing—provide a basis for the next generation of data grid computing.

The DataNet Federation Consortium (DFC) is an NSF funded project that is implementing national data cyber infrastructure through the federation of existing data management systems [1]. The DFC provides a federation hub that links community resources such as data repositories, information catalogs, web services, and compute grids. The federation hub serves as a collaboration environment that supports sharing of data collections and analysis workflows for six science and engineering domains: oceanography, engineering, hydrology, plant biology, cognitive science, and social science. Each domain has unique data management challenges that are driven by discipline specific data types, data formats, data access protocols, data analyses, and data management policies. The unifying goals of the DFC are supports for reproducible data-driven research, promotion of collaborative research, and engagement of students in analysis of "live" data collections.

Based on user requirements, the DFC has identified and implemented collection-based operations that are essential for supporting current research initiatives. A collection-based operation is a function that is applied to a collection of digital objects in the aggregate, as opposed to operations that are performed upon individual files. Examples include the manipulation of a time series composed from a set of files that represent discrete time segments, the re-execution of a workflow using the original input data, and the distribution of replicated data through interactions with software-defined networks. In the case of time series data, the collection is the set of files that comprise the entire time series, and an operation is the manipulation of an arbitrary time interval within the collection. For workflow re-execution, the collection is the set of input and output files and the workflow and parametric files. The operation then corresponds to re-execution of the workflow with the creation of a new collection that holds the new input and output files. For managing interactions with replicas of files, the collection is the set of replicas which are typically located at multiple storage locations, and the operation is the identification of a set of disjoint network paths that optimize simultaneous retrieval from all of the replicas.

For each example, we detail the approach used within the DFC to simplify the collection operations, and we provide a use case from a science and engineering domain.

2. DFC Infrastructure

The DFC uses the integrated Rule Oriented Data System (iRODS) to implement a collaboration environment [2] [3]. The iRODS data grid provides the interoperability mechanisms that are needed to federate existing data management systems, and the virtualization mechanisms that are needed to build a shared collection across multiple institutions. The interoperability mechanisms include:

- Middleware drivers—which enable application of data grid operations at a remote community resource. This requires installing data grid middleware as a software service at each participating site. The middleware driver is used to execute the protocol of the remote resource.
- Micro-services—which enable execution of the protocol required to access a remote community resource. The micro-service is used by the data grid middleware to pull data from the remote site to an analysis platform.
- Policies—which control execution within the distributed environment. Essentially, the data grid must evaluate whether execution of the operation will occur faster at the remote site through a middleware driver, or faster at an analysis platform through the migration of the data. For sufficiently low complexity (number of operations per byte required to analyze the data set), it is always faster to do the computation at the remote community resource. The standard example is a data sub-setting command.

These interoperability mechanisms can be used to manipulate single files, or they can be applied to collections of files. Within the DFC, collections are used to provide a context for data, including organization structure, semantic information stored as collection metadata attributes, provenance information, and administrative information. Given the formation of a collection, one can then ask what operations should be performed on the collection as a whole, with the objective of simplifying the research process. Note that the interoperability mechanisms can be applied to both individual files and to collections.

3. Time Series Collections

The Ocean Observatories Initiative (OOI) manages time series data generated by sensors in ships, autonomous gliders, buoys, on-shore installations, and satellites [4]. The data are aggregated into files in caches, and then distributed to the user community for analysis. The sensor data are encoded in NetCDF files [5], with metadata for time stamps that delineate when the data stream started and stopped.

A significant challenge is the archiving of sensor data streams at the National Oceanic and Atmospheric Agency (NOAA) as climate records. This requires two specific tasks:

1) Automated archiving of the data stream files from the OOI cache to NOAA.

2) Automated retrieval of a specified time segment from the archived data.

Collection-based operations simplify both tasks. A collection can be defined as the set of files that are present in the cache, with an associated operation that synchronizes the files into a time series collection within the archive. The time series collection automatically indexes the files that are archived, constructing a start and stop time index across all of the files that are stored. A request for data access is then made based on the desired start and stop time.

A get operation is translated into a sequence of operations that concatenate the interior files associated with the time series, and subset the files at the beginning and end of the time series to generate the appropriate time sequence. The resulting data set is presented as a single time series sequence for the research analysis.

This approach requires the ability to parse the structure of each file, extract the desired descriptive metadata, apply data sub-setting commands on data arrays, and concatenate data streams. The OOI has chosen the NetCDF format for describing time series data, and has chosen the OPeNDAP protocol [6] for managing data access. The implementation within the DFC then required use of all three interoperability mechanisms:

1) Middleware drivers for OPeNDAP servers. Drivers were developed for both ERDDAP [7] and PyDAP [8] servers to retrieve data sets and send them to the climate record archive.

2) Micro-services for NetCDF manipulation. Micro-services were developed that apply the standard NetCDF operations on a NetCDF file for extracting dimension information, variable names, and metadata, and for applying sub-setting operations on arrays.

3) Policies for controlling the ingestion process. Since the data rate for acquiring sensor data may be very low, the cache is periodically harvested. A periodic iRODS rule is defined that controls the frequency of data archiving.

The approach also required the development of an explicit collection operation to manage the creation of a time series index. A folder is "mounted" as a time series collection within the data grid. Interactions with the mounted folder automatically apply the following collection operations:

1) When a file is put into the mounted folder, the start and stop time is automatically extracted, and added to a time index for the time series data.

2) When a get operation is made on the folder, the time series stream is automatically composed and delivered as a single sensor data stream.

The user can then focus on research issues related to manipulation of sensor data streams, while the underlying data cyber infrastructure automates the archiving of the sensor data and the retrieval of desired time stream intervals.

4. Workflow Re-Execution

Reproducible data-driven research can be implemented through the ability to re-execute analysis workflows using the original input data and input parameters. Within the hydrology discipline, the analysis of a watershed requires the acquisition of data sets from federal agencies, the extraction of information appropriate to the spatial watershed of interest, and the execution of the hydrology analysis. The analysis involves thousands of data sets acquired from:

- US Department of Agriculture—soils data
- US Department of Transportation—roads, dams
- US Geological Survey—national land cover data
- NOAA—NEXRAD [9] precipitation data
- NASA—Landsat TM data [10], MODIS data [11]
- US Geological Survey—digital elevation maps
- NCEP/NCAR [12]—wind speed data

The acquisition and preparation of the data is a labor intensive task that requires interacting with separate protocols for each data source, and different data formats for each data type. Within the DFC, the knowledge required to interact with each community resource (execute the appropriate data retrieval protocol) is encapsulated in a data retrieval micro-service. The steps needed to extract relevant information are encapsulated in data manipulation micro-services. The micro-services are chained into a workflow that manages all of the steps needed to execute the hydrology analysis. The workflow is written in the iRODS rule language, and can be saved as a text file within the iRODS data grid along with the input files. The analysis will be reproducible if the workflow can be re-executed with the same input parameters and input files, and then generate the same output files.

This approach requires the ability to automate the tracking of the provenance of the files that are generated by workflows. Within the DFC this is accomplished by defining a special type of collection that can be mounted as a workflow provenance collection. A folder can be mounted as the collection that automates provenance tracking for a workflow. The steps include:

- Mounting a collection as a workflow provenance area for a specific workflow (.mss file)
- Deposition of an input file into the collection (which lists input parameters, input file names and output file names)
- Automated creation of a run file for executing the workflow
- Automated creation of an output directory for each execution of the workflow
- Automated deposition of the output files into the output directory
- Automated versioning of the output directory for multiple executions with the same input files

Within the data grid the workflow file, input files, and output files can be shared between researchers. It is possible for a second researcher to re-execute a workflow, and compare results with the first execution of the workflow. It is also possible to modify the input parameters or input files, execute the workflow and compare results between the two runs. The automation of workflow provenance acquisition, along with the automation of the retrieval and transformation of input data sets, makes it possible for researchers to focus on the physics behind the analysis instead of the data management.

5. Network Optimization

Workflow management and time series management are both accomplished by associating related operations with a folder (collection). The operations performed upon the folder minimize the data management labor normally associated with data driven research and minimize the amount of time needed to conduct research.

A similar approach can minimize the time needed to move massive data sets over the network. In this case, the collection corresponds to the set of replicas that have been created for data loss risk mitigation. Within the iPlant Collaborative [13], single data sets that are over a terabyte in size are transported over the network. Within the iRODS data grid, parallel TCP/IP streams are used to move the file. This enables the data grid to "fill the pipe", and not prolong the transport time while waiting for transfer receipt acknowledgements. In practice, the performance of parallel TCP/IP streams can be improved by sending each data stream over a separate network path, disjoint from the paths used for the other data streams. If the sources for the data streams are separate replicas, then the data delivery can be further optimized.

With the advent of software defined networks, and through use of the ability to control an OpenFlow switch [14] from an external data management application, the DFC project demonstrated optimization of network data delivery. The approach required developing the following capabilities:

- Creation of micro-services that use the OpenFlow protocol to control the selection of a network path.
- Creation of policies to select among available paths.
- Addition of policy enforcement points to control path selection for each I/O stream of a parallel TCP/IP transport request.

In this example, the operations that are performed to manage the selection of network paths are automatically enforced by the data grid through policies that are specified in a rule base. By modifying the controlling policy, the use of disjoint network paths can be limited to files that are very large in size. Based on current data grid experience, the use of parallel I/O streams leads to better transfer performance when file sizes are larger than 32 MegaBytes.

When transporting smaller files, sites have improved performance by aggregating small files in a staging area into a tar file, then moving a large tar file over the network. The tar file can be manipulated by mounting a folder as a tar file collection. The data grid can then apply operations upon the tar file to list the contents, and extract individual files. A policy can be implemented that manages a staging area, periodically checking for a high-water mark. When a sufficient number of small files have been deposited, the policy can force the aggregation of the files into a tar file, move the tar file using parallel I/O streams to an archive, replicate the file to minimize risk of data loss, and then delete the original small files from the cache.

6. Evaluation of Data Grid Operations

The national projects that are participating in the DataNet Federation Consortium have implemented iRODS data grids as production systems. Each of the discussed data grid operations meets a critical requirement for at least one of the projects. The evaluation of data grid operations is primarily driven by the reduction in researcher time that would otherwise be involved in data management. For the hydrology case, the effort needed to acquire, transform, and prepare the input data sets can be measured in months. By automating the processing steps within the data grid, the time can be reduced to hours. Furthermore, by automating the management of the derived data sets, the analysis can be re-executed by other researchers to verify the results.

For the oceanography example, the intent is to automate archiving of data records. The mechanisms that apply the archival processes are run as a rule that is periodically executed. On each execution, any new climate records that have been deposited into a staging area are replicated to the archive. This is an idempotent process, with any records that have transfer failures being automatically discovered and transferred on the next execution of the periodic rule.

For the software defined network example, a demonstration was given at Supercomputing'13. The ExoGeni experimental network [15] was used. Four parallel I/O streams were moved through the network over three disjoint network paths. A speedup of a factor two was observed, since two of the streams had to go over the same network path. In this demonstration, network bandwidth was the limiting factor for the data transfers.

7. Future Work

The theory behind data grid operations is being generalized to define the minimal set of fundamental operations from which the above examples can be composed. The approach is based on the expression of data grid operations as a composition of a pre-process rule, the fundamental operation, and a post-process rule. The pre-process and post-process rules are implemented as distributed workflows that are managed by the data grid. Thus processing done at the remote data resource may be implemented as a pre-process policy. Processing done at an analysis platform may be implemented as a post-process policy. Since the datagrid can manage descriptions of arbitrary workflows, this approach promises to provide a general solution for all types of data management systems.

8. Summary

The DataNet Federation Consortium has used the concept of operations on collections to minimize the labor associated with data-driven research. The approach involves mounting a folder to support a specific type of operation, depositing files into the folder, and then applying the operation on the folder. Through this approach the DFC has successfully managed time series data streams, workflow provenance capture, workflow re-execution, and data transport optimization.

Acknowledgements

The research results in this paper were funded by the NSF Office of Cyber Infrastructure OCI-0940841 grant, "DataNet Federation Consortium", (2011-2016).

References

[1] DataNet Federation Consortium. http://datafed.org

[2] Rajasekar, R., Wan, M., Moore, R., Schroeder, W., Chen, S.-Y., Gilbert, L., Hou, C.-Y., Lee, C., Marciano, R., Tooby, P., De Torcy, A. and Zhu, B. (2010) iRODS Primer: Integrated Rule-Oriented Data System. Morgan & Claypool.

[3] Ward, J., Wan, M., Schroeder, W., Rajasekar, A., de Torcy, A., Russell, T., Xu, H. and Moore, R. (2011) The Integrated Rule-Oriented Data System (iRODS 3.0) Micro-Service Workbook. DICE Foundation, Amazon.com.

[4] Ocean Observatories Initiative. http://oceanobservatories.org

[5] NetCDF Network Common Data Form. http://www.unidata.ucar.edu/software/NetCDF/

[6] OPeNDAP Protocol. http://www.opendap.org

[7] ERDDAP Environmental Research Division's Data Access Program. http://coastwatch.pfeg.noaa.gov/erddap/index.html

[8] PyDAP Python Data Access Protocol. http://www.pydap.org

[9] NEXRAD Next Generation Radar. http://www.ncdc.noaa.gov/oa/radar/radarresources.html

[10] LandSat Thematic Mapper Data. http://landsat.gsfc.nasa.gov

[11] NASA MODIS Moderate Resolution Imaging Spectroradiometer Data. http://modis.gsfc.nasa.gov

[12] NCEP/NCAR National Centers for Environmental Prediction/National Center for Atmospheric Research Reanalysis. http://www.esrl.noaa.gov/psd/data/reanalysis/reanalysis.shtml

[13] The iPlant Collaborative. https://www.iplantcollaborative.org

[14] OpenFlow Switch. https://www.opennetworking.org/sdn-resources/onf-specifications/openflow

[15] ExoGeni Global Environment for Network Innovation Testbed. http://www.exogeni.net