Data Management Challenges of Data-Intensive Scientific Workflows

Ewa Deelman, Ann Chervenak
USC Information Sciences Institute, Marina Del Rey, CA 90292
deelman@isi.edu, annc@isi.edu

Abstract

Scientific workflows play an important role in today's science. Many disciplines rely on workflow technologies to orchestrate the execution of thousands of computational tasks. Much research to-date focuses on efficient, scalable, and robust workflow execution, especially in distributed environments. However, many challenges remain in the area of data management related to workflow creation, execution, and result management. In this paper we examine some of these issues in the context of the entire workflow lifecycle.

1. Introduction

Scientific applications such as those in astronomy, earthquake science, gravitational-wave physics, and others have embraced workflow technologies to do large-scale science [3]. Workflows enable researchers to collaboratively design, manage, and obtain results that involve hundreds of thousands of steps, access terabytes of data, and generate similar amounts of intermediate and final data products. Although workflow systems are able to facilitate the automated generation of data products, many issues still remain to be addressed [22]. These issues exist in different forms in the *workflow lifecycle* [16]. We describe the workflow lifecycle as consisting of a workflow generation phase where the analysis is defined, the workflow planning phase where resources needed for execution are selected, the workflow execution part, where the actual computations take place, and the result, metadata, and provenance storing phase.

During workflow creation, appropriate input data and workflow components need to be discovered. During workflow mapping and execution data need to be staged-in and staged-out of the computational resources. As data are produced, they need to be archived with enough metadata and provenance information so that they can be interpreted and shared among collaborators. This paper describes the workflow lifecycle and discusses the issues related to data management at each step. We describe challenge problems and, where possible, illustrate them in the context of the following applications: the Southern California Earthquake Center (SCEC) CyberShake [26], an earthquake science computational platform, Montage [7], an astronomy application, and Laser Interferometer Gravitational-Wave Observatory's (LIGO) binary inspiral search [8], a gravitational-wave physics application. These computations, represented as workflows, are running on today's national cyberinfrastructure and use workflow technologies such as Pegasus [17] and DAGMan [12] to map high-level workflow descriptions on to the available resources and execute the resulting computations. This paper describes the challenges, possible solutions, and open issues faced when mapping and executing large-scale workflows on current cyberinfrastructure. We particularly emphasize the issues related to the management of data throughout the workflow lifecycle.

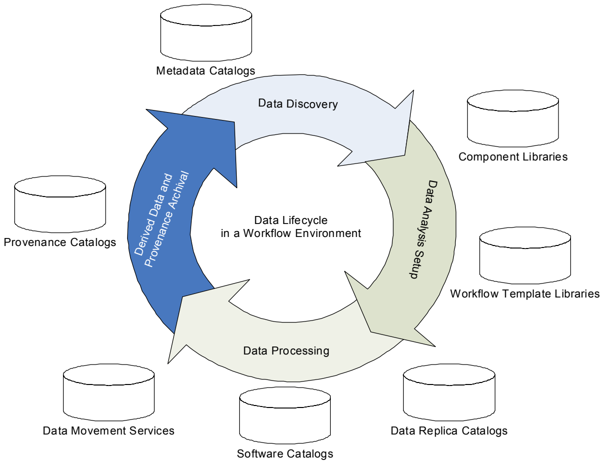

Figure 1: Data Lifecycle in a Workflow.

From the point of view of data, the workflow lifecycle includes the following transformations (see Figure 1): data discovery, setting up the data processing pipeline, generation of derived data, archiving of derived data and its provenance. Data analysis is often a collaborative process or is conducted within the context of a scientific collaboration. An example of such a large-scale collaboration is the

LIGO scientific Collaboration (LSC), which brings together physicists from around the world in a joint effort to detect gravitational waves emitted by celestial objects [6]. In astronomy, projects such as Montage develop community-wide image services. In earthquake science, scientists bring together community models to understand complex wave propagation phenomena.

2. Workflow Creation

Data and Software Discovery Scientists in a collaboration frequently submit workflows to process data sets and derive scientific knowledge. These collaborators may submit related workflows and build upon earlier work by other scientists. Thus, scientists need to be able to discover information about workflows that have been run in the past, identify data sets of interest, and locate analysis code and workflow templates. In the workflow creation stage, they identify data sets and analysis code of interest by unique logical identifiers or metadata, independent of where these data sets or analysis codes may be physically located in the distributed environment. Discovery of data sets, application codes, workflow templates, etc., is often done by querying various catalogs. *Metadata catalogs* store attributes that describe the contents of data sets. *Provenance catalogs* [28] store information about computations and workflows to provide a detailed record of how analyses are run, including information about inputs to computations, application parameters used, calibration values for equipment, versions of workflow and analysis software used, etc.

Community Standards A challenging aspect of setting up these discovery catalogs is the need for communities to agree on standards for specifying metadata and provenance. Often, scientists in an application domain spend great effort to agree on a metadata ontology that is rich enough to describe the meaning of data sets used and generated in the domain. Some of the most successful efforts have been made in bioinformatics, where scientists are not only defining metadata standards but also sharing descriptions of services used in workflows as well as workflows themselves, for example, as part of myExperiment [23]. Similar standards need to be defined in other scientific domains, including standards to describe the software characteristics, inputs, outputs, versions, etc.

Metadata Catalogs Many application communities have deployed their own metadata catalogs to store and query metadata attributes using relational databases or RDF triple stores [25], including LIGO and the Earth System Grid (ESG) [27], a scientific collaboration that supports climate modeling science. The schema for these databases corresponds to the metadata ontology defined by the community. Several systems provide general metadata catalogs that are independent of particular application communities. The Storage Resource Broker (SRB) system [33] provides a catalog called MCAT that stores metadata and is also used to coordinate data accesses, enforce access permissions and maintain consistency for replicated data. The Metadata Catalog Service [18] provides a set of generic and extensible metadata attributes. Even though metadata technologies exist, the biggest challenge is for the scientific communities to decide on common definitions of terms.

Data Provenance Data provenance technologies are still being developed [32]. A challenge in the provenance area is the ability of users and workflow systems to interpret provenance information produced by a different or unfamiliar workflow system. To facilitate data discovery where data have been produced by different systems, an effort to standardize on data provenance representation is underway [30, 31]. Once standards are in place, the challenge for the workflow systems will be to implement them.

Workflow Creation Provenance An interesting aspect of workflow creation is the ability to re-trace how a particular workflow has been designed, or to determine the provenance of the workflow creation process. A particularly interesting approach is taken in Vistrails [21] where the user is presented with a graphical interface for workflow creation and the system incrementally saves the state of the workflow as it is being designed. As a result, users may re-trace their steps in the design process, choose various “flavors” of the same workflow and try and retry different designs. Another challenge is to be able to capture not only the how but the why of the design decisions made by the users.

3. Workflow Planning and Execution

Workflow Data and Component Selection During workflow creation, scientists specify the applications or workflows they want to run and the input data sets for these computations using unique logical identifiers. In the workflow planning stage, these logical identifiers for applications and data must be mapped to resources in the distributed environment. For data sets that are inputs to workflows or analysis, this requires discovering the location of one or more copies of the desired data sets, selecting among them, and often copying or staging the data sets onto resources where computations will run. For analysis codes, this requires finding where the code exists and possibly transferring the code to the location where the computation will run. A scheduler is responsible for selecting among available data sets, selecting appropriate computational resources to run each task of a workflow, and orchestrating the movement of data sets and the execution of workflow tasks. Schedulers or workflow mappers need to be able to optimize the workflows based on some user-specified criteria [35, 37].

Data Dependencies In the workflow execution stage, an execution manager such as DAGMan keeps track of tasks that must run and the dependencies among them. Earlier tasks in the workflow may produce intermediate data products that are consumed by tasks that run later. These intermediate data products may need to be staged from the resource where the earlier task ran to the resource on which the later task will run. The workflow execution system delays execution of a particular task until all its input data products are available on the computational resources where the task will run.

Challenges of workflow planning and execution include finding available resources whose capabilities match the requirements of the workflow. This in turn requires up-to-date information about the current state of each resource, so that computational tasks or data transfer jobs are not assigned to resources that are already heavily loaded or are temporarily unavailable.

Distributed Data Environment A major challenge in today's applications is the physical management of data in the distributed environment. Although the processing power may be available, getting the data to that computational resource may be time consuming and error-prone. Figure 2 shows the distributed nature of scientific data. Most often it is stored in archives and staged to the computational sites on demand. In case of LIGO, data is kept archived at each computational site within the collaboration. If the workflows execute on the Open Science Grid [1], data needs to be staged-in from one of the LIGO sites. Within the computational site, often a cluster, we also distinguish between shared storage and storage local to a computational node.

Identifying the location of desired data sets is a challenge in this type of distributed environment. Typically, replica location [9, 11] or metadata catalogs [18, 33] record mappings from logical identifiers for data to one or more physical locations where copies of the data sets are stored. Based on knowledge of the state of resources (the latency, bandwidth and load of storage systems, network bandwidth among nodes, etc.) that may be provided by information services [13], the workflow planner selects among available data replicas. In particular, the planner may try to select copies of the data that are "close" to the computational resources where workflow tasks will run, with respect to network latency or other metrics.

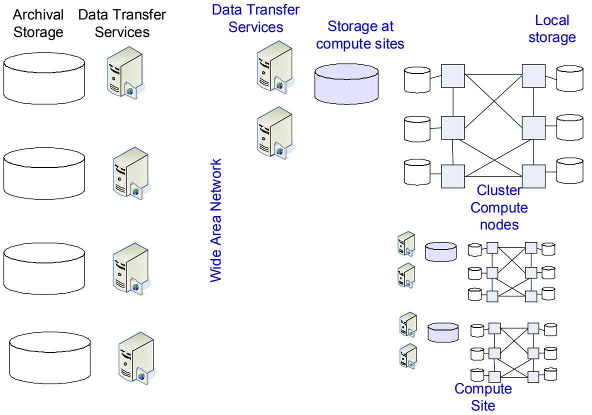

Figure 2: View of the Distributed Data Environment.

Asynchronous Data Placement It may be advantageous for workflow planning and execution services to coordinate with *data placement services*, whose role is to move data asynchronously with respect to workflow execution with the goal of improving the execution time of workflows. For example, a workflow engine might provide hints to a data placement service about required data sets as well as the expected ordering of data set access, based on knowledge or dependencies in the workflow. Based on these hints, the placement engine can asynchronously stage some of the data required by the workflow engine onto shared storage resources near where the workflow tasks will execute. In earlier work, we demonstrated the potential advantage of prestaging data sets that are needed for workflow execution [10]. We showed that such prestaging of input data reduced the execution time of data-intensive workflows considerably.

In current work, we are exploring a range of data placement algorithms for staging data off of storage resources after execution is complete as well as for prestaging data onto resources before execution begins. We are interested in various approaches to the design of placement services, ranging from workflow and placement services that are tightly integrated to those that have relatively little communication or interaction.

Data Transfer Challenges Workflows rely on a variety of data transfer mechanism over the wide area. These include such tools as GridFTP [5], the Fast Data Transfer (FDT) service [4], and others. In order to support the data transfer needs of their users and load-balance the requests, many grid installations deploy multiple data movement servers targeting the same storage system. However, failures and server timeouts still occur. We found many such errors when running the CyberShake workflows on the TeraGrid [15]. If errors occur due to problems accessing the input data, another data source (if it exists) can be chosen by the workflow system. This data source would be found in a data replica catalog either during workflow planning or as part of the fall-back mechanisms in the workflow execution. In order to deal with failures at the destination, a simple retry can be performed by the workflow system, or a different data transfer server can be chosen for the data movement. Retries are able to deal with temporary server overloads, transient network failures, and other intermittent problems. However, other types of failures are harder to deal with.

Data Storage Challenges When applications access Terabyte-size data sets, storage available at the execution sites can be a limiting factor for the successful data staging and thus for successful workflow execution. This problem is particularly challenging because there are few systems deployed today that support disk space reservation, so applications compete with each other for space on a first-come-first-serve basis. Even if there are disk quotas present on the execution sites, these quotas are usually maintained at the Virtual Organization (VO)-level [20], and therefore, users within the VO compete for space. In LIGO, for example, binary inspiral workflows require a minimum of 221 GBytes of gravitational-wave data and approximately 70,000 computational workflow tasks [36]. The resources of the OSG provide on average 258 GB of shared scratch disk space. The shared scratch disk space is used by approximately 20 VOs within the OSG. Thus LIGO workflows need to be carefully mapped to the available OSG resources, and new algorithms are needed to manage the size of the workflow data footprint during execution.

It is possible for workflow systems to take into account the storage space available at a particular site when making task scheduling decisions [34]. The workflow system can find out how much space is available at a remote site, estimate the amount of space needed by the workflow tasks and consider only the sites that provide a suitable amount of space for resource selection. One of the challenges in this case is the ability to receive accurate information from the resources. Another challenge is the ability to estimate the amount of storage needed for the output data of workflow tasks. Also, because the available space can change before or while the data transfer is being done, workflow systems need to be able to recover from disk space failures and re-plan the workflow for execution elsewhere. The ability of the workflow system to clean up data sets when they are no longer needed can reduce the workflow footprint [36] and thus is an important factor in successful workflow execution.

Data Management inside the Resource In distributed environments, clusters are often managed by schedulers such as Condor [19] or PBS [24]. As the clusters grow larger, the issue of data management within the cluster becomes important. If workflow tasks access data via the shared file system, then the overall application may suffer if the file server becomes overloaded. A solution to this problem is for the workflow tasks to compute on data that reside on a local disk. The issue is then to provide mechanisms within the workflow to perform the staging of data from a shared location to the local file system. Most resource management systems support a way of specifying this type of data staging in the submission scripts. In a distributed environment these scripts are usually generated automatically by remote submission software. The challenge for the workflow system is to be able to identify the properties of the remote execution site and to pass the appropriate information to the submission software.

Dealing with Data too Large to Move In some cases, the data sets that workflows operate on are too large to move efficiently and process at a remote location. It is necessary for the workflow scheduling algorithms to take this into account when deciding which resources to use for the computation. A challenge for the algorithms is trying to figure out the costs involved in moving the data over the network, which can vary greatly based on network load and the latency and bandwidth of source and destination storage systems. Workflow systems depend on monitoring and information systems for current information on network and storage performance.

Virtual Data When dealing with large numbers of workflows and large VOs, it is often the case that multiple workflows may use the same input data sets or intermediate data products. For example, raw data managed by a VO is often in a form that needs to be calibrated first to be scientifically viable. Thus, many workflows incorporate the calibration step in their computations. As a result, the intermediate, calibrated data can be shared by other workflows and users within the collaboration, provided the data are correctly tagged with metadata and provenance information. The challenge for the workflow system is then to recognize when intermediate data already exist; to determine whether it is more efficient to access the existing data rather than recompute it; and to make use of this information to possibly reduce the workflow.

4. Derived Data and Provenance

Both final and intermediate workflow results are typically staged out to a permanent storage location. In order for this data to be interpretable both by the user and his/her colleagues, metadata and provenance information about the data need to be stored as well.

Metadata Management In some scientific disciplines, such as astronomy, there are standard data formats [2] that include metadata about an image as part of the image file header. Community codes then have the obligation to generate and save the metadata inside the files they generate. In Montage for example, the application reads-in FITS files that contain image data and write new images in the same format with new metadata included in the header. However, it may be difficult to search for specific data by opening and reading the files. Thus, additional workflow components can be provided to extract and save metadata in a metadata catalog. In general, it is very difficult for workflow systems to appropriately catalog metadata associated with derived data, as there is no standard way for software components to generate the metadata, and most of the time, the software components do not provide any metadata for the results. New capabilities need to be developed for communities to define standards and formats. One challenge is to determine how to retrofit existing legacy codes to provide metadata information, or more likely to wrap them with metadata capabilities. Another challenge is to have incentives for community members to develop new metadata-compliant codes.

Provenance Management Having metadata information is often not sufficient to fully validate scientific results or to reproduce them. Additional information—*provenance information*—-is needed to support both scientific and engineering reproducibility [22]. Provenance captures information about which data were used during the workflow execution, which software was run, and what were the computing, storage, and other resources used to obtain the results. Detailed information will include the various parameter settings, environmental variables, etc. All this information can and should be captured by the workflow management system while the workflow is executing. In terms of scientific reproducibility, where one wants to share and verify their findings with a colleague inside or outside the VO, the user may need to know what data sets were used, what type of analysis and with what parameters. However, in cases where the results need to be reproduced "bit-by-bit", more detailed information about the hardware architecture of the resources, environment variables used, library versions, etc. are needed. Finally, provenance can also be used to analyze workflow performance, as was done for example in the context of CyberShake [15], where the provenance records were mined to determine the number of tasks executed, their runtime distribution, where the execution took place, etc.

In some cases, the workflow management system may modify the executable workflow to the point that it is not easy to map between what has been executed and what the user specified [14]. As a result, information about the workflow restructuring process needs to be recorded as well [29]. This information allows us not only to relate the user-created and the executable workflow but is also the foundation for workflow debugging, where the user can trace how the specification they provided evolved into an executable sub-workflow.

One of the challenges in managing provenance, and especially workflow provenance where the information can be significant in size (peta-byte scale), is the ability and necessity to determine what to store. Workflow management system designers need to work with application developers to define the important provenance components. Also, some other methods of periodically reducing, compressing, and otherwise managing provenance information may need to be employed. As a result, there is a risk of not having the needed information and thus being unable to verify and thoroughly analyze it or reproduce it. Another challenge related to the size of provenance is the efficient navigation of the information. Tools such as PASOA [28] and others are addressing these issues.

5. Conclusions

In this paper we examined the data lifecycle as it relates to the scientific workflow lifecycle. We discussed challenges in data and software discovery, data and component selection, physical data movement, and derived data, metadata and provenance management. We believe that data management issues are critical to scientific workflows, and although many technologies and point solutions exist today, much work remains to be done in that area. With the advent of multi-core processors, data management is increasing in importance. The need to bring data reliably and fast to where the computation takes place is critical. In cases where the cost of data transfer is too expensive, we have the need to bring software and the necessary computation environment to the data. In either case, issues of metadata and provenance, and workflow mapping techniques remain.

6. Acknowledgements

This work was supported in part by the NSF OCI-0438712 grant and by the Department of Energy's SciDAC II program under grant DE-FC02-06ER25757. The authors would like to thank the ESG, LIGO, Montage, and SCEC collaborators for helpful discussions and fruitful collaborations.

7. References

[1] "Open Science Grid." www.opensciencegrid.org

[2] "Flexible Image Transport System." http://fits.gsfc.nasa.gov/

[3] *Workflows in e-Science*. I. Taylor, E. Deelman, D. Gannon, and M. Shields, Eds.: Springer, 2006.

[4] "Fast Data Transfer (FDT) service," 2007.

[5] W. Allcock, et al., "Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing," *Mass Storage Conference*, 2001.

[6] B. C. Barish, et al., "LIGO and the Detection of Gravitational Waves," *Physics Today,* vol. 52, 1999.

[7] G. B. Berriman, et al., "Montage: A Grid Enabled Engine for Delivering Custom Science-Grade Mosaics On Demand," in *SPIE Conference 5487:* 2004.

[8] D. A. Brown, et al., "A Case Study on the Use of Workflow Technologies for Scientific Analysis: Gravitational Wave Data Analysis," in *Workflows for e-Science*, I. Taylor, et al. Eds.: Springer, 2006.

[9] A. Chervenak, et al. "Giggle: A Framework for Constructing Sclable Replica Location Services," in *SC2002 Conference*, Baltimore, MD, 2002.

[10]A. Chervenak, et al. "Data Placement for Scientific Applications in Distributed Environments," in *Grid 2007*

[11]A. L. Chervenak, et al. "Performance and Scalability of a Replica Location Service," in *HPDC-13*, 2004.

[12]P. Couvares, et al., "Workflow Management in Condor," in *Workflows in e-Science*, I. Taylor, et al. Eds., Springer, 2006.

[13]K. Czajkowski, et al., "Grid Information Services for Distributed Resource Sharing," in *HPDC* 2001

[14]E. Deelman, et al., "Pegasus: a Framework for Mapping Complex Scientific Workflows onto Distributed Systems," *Scientific Programming .l,* vol. 13, 2005.

[15]E. Deelman, et al., "Managing Large-Scale Workflow Execution from Resource Provisioning to Provenance Tracking: The CyberShake Example," e-*Science* 2006.

[16]E. Deelman, et al., "Managing Large-Scale Scientific Workflows in Distributed Environments: Experiences and Challenges," in *Workflows in e-Science*, 2006.

[17]E. Deelman, et al., "Pegasus: Mapping Large-Scale Workflows to Distributed Resources," in *Workflows in e-Science*, I. Taylor, et al. Eds.: Springer, 2006.

[18]E. Deelman, et al., "Grid-Based Metadata Services," in *16th International Conference on Scientific and Statistical Database Management*, 2004.

[19]D. H. J. Epema, et al., "A Worldwide Flock of Condors: Load Sharing among Workstation Clusters," *Future Generation Computer Systems,* vol. 12, 1996.

[20]I. Foster, et al., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int. J. of High Performance Computing Applications,* vol. 15, 2001.

[21]J. Freire, et al., "Managing Rapidly-Evolving Scientific Workflows.," *IPAW,* vol. 4145, pp. 10-18, 2006.

[22]Y. Gil, et al., "Examining the Challenges of Scientific Workflows," *IEEE Computer,* December 2007.

[23]C. A. Goble, et al., "myExperiment: social networking for workflow-using e-scientists," *WORKS,* pp. 1-2, 2007.

[24]R. Henderson, et al., "Portable Batch System: External Reference Specification," 1996.

[25]O. Lassila, et al., "Resource Description Framework (RDF) Model and Syntax Specification," 1999.

[26]P. Maechling, et al., "Simplifying construction of complex workflows for non-expert users of the Southern California Earthquake Center Community Modeling Environment," *SIGMOD Record,* vol. 34, 2005.

[27]D. E. Middleton, et al. "Enabling worldwide access to climate simulation data: the earth system grid (ESG)," *Scientific Discovery Through Advanced Computing (SciDAC 2006), Journal of Physics: Conference Series,* vol. 46, pp. 510-514, June 25-29, 2006

[28]S. Miles, et al., "The Requirements of Using Provenance in e-Science Experiments," *J. of Grid Computing,* 2006.

[29]S. Miles, et al., "Connecting Scientific Data to Scientific Experiments with Provenance," in *e-Science*, 2007

[30]L. Moreau, et al., "The Open Provenance Model," University of Southampton 2007. http://eprints.ecs.soton.ac.uk/14979/1/opm.pdf

[31]L. Moreau, et al., "The First Provenance Challenge," *Concurrency and Computation: Practice and Experience,* 2007.

[32]L. Moreau, et al., "Concurrency and Computation:Practice and Experience, Special Issue on the First Provenance Challenge," 2007.

[33]A. Rajasekar, et al., "Storage Resource Broker-Managing Distributed Data in a Grid," *Computer Society of India Journal, Special Issue on SAN,* vol. 33(4), 2003.

[34]A. Ramakrishnan, et al., "Scheduling Data -Intensive Workflows onto Storage-Constrained Distributed Resources," in *CCGrid* 2007.

[35]R. Sakellariou, et al., "Scheduling Workflows with Budget Constraints," *Integrated Research in Grid Computing, CoreGrid series, Springer-Verlag, 2005.*

[36]G. Singh, et al., "Optimizing Workflow Data Footprint " *Scientific Programming Journal,* vol. 15, 2007

[37]M. Wieczorek, et al., "Scheduling of Scientific Workflows in the ASKALON Grid Environment," *SIGMOD Record,* vol. 34, pp. 56-62, 2005