GeS 2007
German e-Science

Available online at http://www.ges2007.de

Negotiation-based Choreography of Data-intensive Applications in the C3Grid Project

C. Grimme[1], T. Langhammer[2], A. Papaspyrou[1], and F. Schintke[2]

[1] Robotics Research Institute - Section Information Technology, Dortmund
University, 44221 Dortmund, Germany
email: {alexander.papaspyrou, christian.grimme}@udo.edu
[2] Zuse Institute Berlin (ZIB), 14195 Berlin-Dahlem, Germany
email: {langhammer, schintke}@zib.de

Abstract

We present a negotiation and agreement strategy and protocol for the efficient scheduling of data intensive jobs in the Grid. It was developed with the background of the Collaborative Climate Community Data and Processing Grid (C3Grid), which provides a comprehensive infrastructure for solving computational problems in Earth System Science. The presented solution is a subset of the overall C3Grid architecture and especially focuses on the collaboration of Data Management and Workflow Scheduling. We evaluate our approach on a case study representing a complex application typical for climate research. Finally, extensions for future work – especially on standardization efforts – are reviewed.

1 Introduction

Earth System Science investigates the highly dynamic processes and their chemical formation of the diverse subsystems like oceans, atmosphere or biosphere and their longterm changes, especially since the beginning of the industrialization in the 19th century. Therefore large amounts of measured and simulated data are acquired, selected, preprocessed, transported and analyzed by a researcher using highly demanding applications to produce results in various areas of research (e.g. short term weather forecast or storm track analysis).

Today, scientists do not have a coherent working environment for their studies which typically consist of a set of dependent tasks to get the result to a single scientific question: data that are relevant to the study have to be discovered, acquired, and then preprocessed manually and separately on heterogenous resources, which are distributed over several institutes with different access rights and different working environments. This is a tedious and time consuming task, which constricts international collaboration concerning seamless data exchange and analysis.

The Collaborative Climate Community Data and Processing Grid (C3Grid) is a cooperation of earth system science and computer science researchers that aims to provide a Grid technology solution to overcome the aforementioned deficiencies. Therefore, the following requirements have to be fulfilled:

- Standardized access to heterogeneous and distributed data archives, which includes the local selection and preprocessing to minimize wide area data transfers.
- Automatic co-allocation of compute and data resources to ensure data availability during processing time due to planned data transfers.
- Reusability of tools[1], that are already present in the earth system science community.
- Comprehensive support for geographic metadata according to ISO 19115 [6] and its XML embedding according to ISO 19139.
- A user-friendly interface to access the Grid.

In the course of this paper the central components for data management and scheduling which must be capable to agree on the future availability of data and compute resources are discussed in detail.

The remainder of this paper is structured as follows: in Section 2 we give an overview of C3Grid's architecture with a focus on the Data Management and Workflow Scheduler and their interaction. Thereafter, in Section 3 we apply our concept to a common scientific problem in climate research. Section 4 briefly concludes our work and shows future perspectives for the C3Grid development.

2 Architecture

First, we describe the overall C3Grid system architecture with a focus on the Data Management System (DMS) and the Workflow Scheduling Service (WSS). Afterwards we present our negotiation scheme for those services.

2.1 General Overview

The C3Grid follows the idea of a Service Oriented Architecture (SOA) [3, 7] where individual services are loosely coupled and can be accessed without knowledge of the system behind. From top to bottom, the architecture consists of three distinct layers (see Figure 1). The portal layer provides interfaces for the Grid end user, like a web site[2] or programming interfaces. The middleware layer consists of the distributed Grid infrastructure of information- and data-related services, processing resources and temporary disk stores. The bottom layer provides a unified abstraction for the local base data, metadata and compute providers.

Execution of submitted jobs and manipulation of data needed for those jobs are done in the Virtual Workspace, which is backed by DMS-managed storage space. Access to resources for DMS and WSS is provided by two underlying interfaces: the Data Provider Abstraction (DPA) and Compute Provider Abstrac-

[1] E.g. the Climate Data Operators [8].

[2] This is being realized using the GridSphere Portal Framework [5].

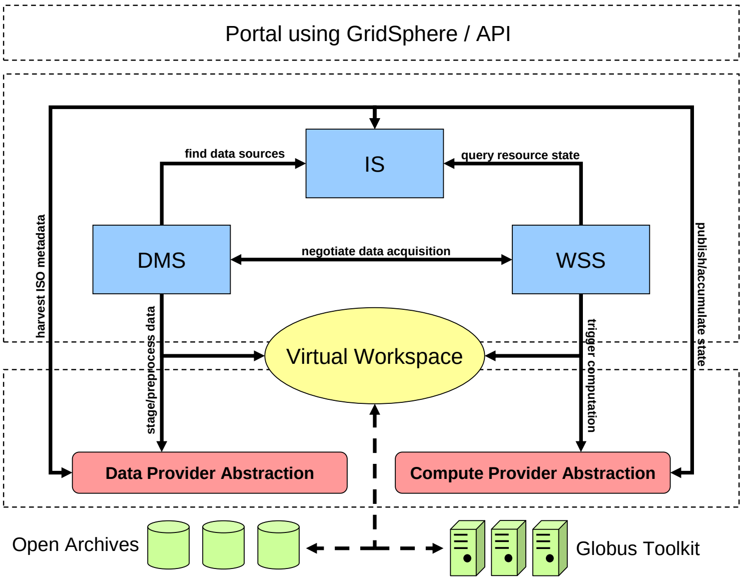

Figure 1: Overview of the C3Grid SOA.

tion[3] (CPA). They facade an arbitrary set of storage or computational resources with a common interface to enable standardized use of underlying providers' services. That is, a resource provider can act as data provider, compute provider or both depending on the implemented interfaces.

The management of data preparation (staging and preprocessing) as well as data transfer between providers is done by the DMS. However, decisions on data source selection and job processing locations are made by the WSS. The interaction of both services is regulated by a negotiation and agreement protocol which is describe in Section 2.4.

Data discovery and resource state accumulation is provided by the Information Service (IS). It is queried by the DMS to find sources of required data and publishes state information about compute resources to be requested by the WSS. Here, data related information is harvested [4] as ISO metadata through the DPA while status information is registered from compute providers directly.

2.2 Data Management Service

The DMS provides operations to access data archives, to manage distributed, replicated files and respective metadata information, and to transfer data between Virtual Workspaces.

[3] Currently an adapter to the Globus Toolkit 4.x [2] is implemented.

Staging from Distributed Archives

In order to support the great diversity of different archive systems (tape archives, disk file systems, databases,...), we designed a Data Provider Abstraction which is specified by a generic web service interface. If an institute wants to share its data in the Grid has to implement this interface and has to establish thereby the connection to its local storage system. The interface design takes into account that the majority of climate data is stored in relational database systems. Data staging requests may contain typical climate data selection arguments (geographic cuts, climate parameter selection,...), which, on the provider side, can be mapped for instance to respective database queries. This preprocessing functionality is essential for an early reduction of data during workflow execution.

Management of Virtual Workspaces

Virtual Workspaces are distributed disk shares for intermediate processing data. Any modifying access to Virtual Workspaces is under the control of the DMS, which decides on the creation, replication and deletion of files.

In order to prevent outdated data from polluting Virtual Workspaces, each file is being assigned a limited lifespan. The DMS guarantees availability within this lifespan and removes the file after it has expired.

The DMS supports file-level replication by two abstractions: logical files, which have attributes independent from any physical location, and physical files, which are ordinary files in a local UNIX file system. Replication is implemented as 1-to-N relation between logical and physical files, i.e. each logical file has a number of replicas as identical copies of physical files on Virtual Workspaces.

Planning of Future Transfers

Our Grid data management distinguishes from state-of-the-art systems by its active planing of data stagings and remote replications. The DMS is able to make assumptions about transfer times based on current information about resource availability.

One source of information are the primary data themselves. Depending on the type of storage system and the size of data, accessing archive systems may require seconds (direct disk access) up to hours (several tape accesses). The generic provider interface provides operations, which can be called by the DMS to get estimations of the staging time for a given request. These estimations may be of most variable quality. In an hierarchical storage system a file cached on disk can be delivered instantly whereas files spread over several tape archives may take hours to access. Nevertheless, also worst-case estimations given as order of magnitude can be important information for the DMS. Other resource information is much more precise and predictable. For example the size of available workspace is known in advance because the DMS has total control over all file store there.

Transfer planning is initiated by agreements between DMS and WSS. We describe these agreements in detail in Section 2.4.

Metadata Handling

ISO 19115 and ISO 19139 metadata is ubiquitous in C3Grid. We use it not only to describe archive data, but also to identify data in grid workspaces. Processing tasks in the C3Grid are designed to write new output data with a complete set of ISO descriptions. The DMS is aware of this fact and cares about the registration in the data-related part of the IS (DIS).

It is worth mentioning that all information about available data in the C3Grid is provided by the DIS. This refers to data at local institutes as well as files located in Grid workspaces. Therefore, two alternatives exist for scientists using the Grid for data analysis to share their results:

- Intermediate results of limited lifetime may stay in grid workspaces. This requires that the processing tools have sufficiently described the data with metadata which the DMS can automatically register in the DIS.
- Final results can be downloaded and archived at the users institute. Thus, the creation of meta information is outside the scope of the C3Grid but, nonetheless, necessary for re-publication.

2.3 Workflow Scheduling Service

A *workflow* comprises a set of atomic tasks (i.e. process execution, data transfer and so on) which may have dependencies between each other. In our case, such a workflow consists of processing and transfer tasks which generally represent the execution of staging, preprocessing or analysis as well as data transportation respectively. A more detailed example of a workflow is given in Section 3.

The Workflow Scheduling Service (WSS) provides functionality for the choreography of the execution of such workflows. That is, it exports an interface to the Portal layer which allows the submission, control, and monitoring of a workflow, and manages the ordered dispatch of tasks depending on the user's requests.

Therefore, the WSS has interfaces to the different (possibly heterogeneous) compute resources in order to expose a uniform set of functionality to the system. This is essential as advanced reservation capabilities are crucial to the choreography services, but most job submission systems do not implement this feature [9]. Furthermore, the incompatible methods for job management (submission, cancellation, etc.) are unified here.

Next, it is necessary that the user-requested data is available in the correct Virtual Workspace at task execution time. To achieve this, the WSS steers the DMS based on the data requested by the user and the execution plan of the whole workflow. This includes finding suitable data sources and agreeing on matching provisioning offers.

In this context, the WSS also resolves the task inter-dependencies within the workflow, determines the correct execution order, schedules the tasks on a set of

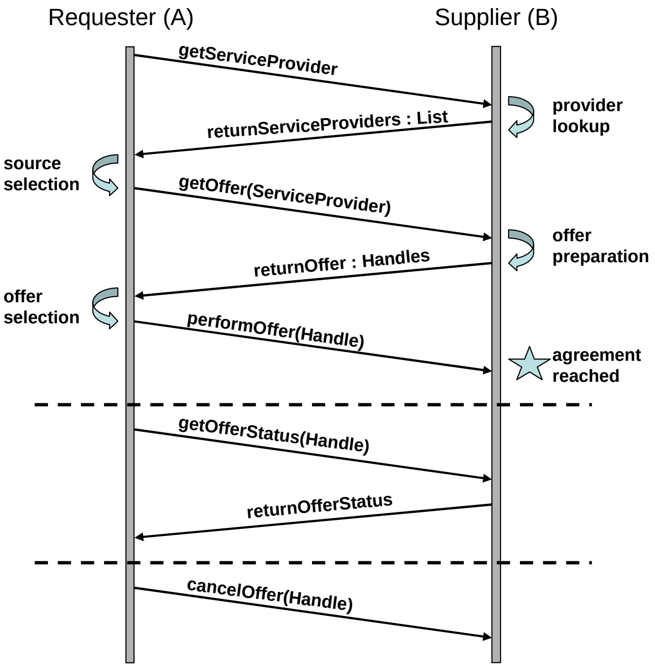

Figure 2: Schematic diagram of the proposed General Negotiation and Agreement Protocol in C3Grid.

compute resources and triggers the submission at execution time [10].

2.4 Co-scheduling of Data and Jobs

In the design of the negotiation process between scheduler and DMS, we identified two typical types of request. On the one hand, there are requests which can be performed with few resources, like status requests or manipulation of file attributes. Those can be performed directly. On the other hand, there are requests which entail costly operations and thus require more sophisticated planning regarding their execution. To this end, we propose a negotiation mechanism for agreeing on processing policies for such operations.

The sequence diagram shown in Figure 2 illustrates the General Negotiation and Agreement Protocol we have developed. In this protocol, service A starts by requesting an operation. Service B, which provides the operation, responds with a concrete offer meeting the constraints as well as a handle for identification purposes. In order to complete the agreement, service A must confirm the offer with the identifying handle. Nevertheless, service B may also ignore a confirmation if too much time has passed since the offer has been made.

After these three steps, the agreement is completed. Service A can now track the execution by requesting a status report or cancel it by sending a respective message. Note that A can send cancel and status requests at any point in time, even before the agreement's negotiations have been completed.

In C3Grid, the WSS takes the requesting part, while the DMS acts as provider of transfer and staging operations. In the first step, the WSS gives a time slice in which the operation must be finished. The DMS responds with the earliest time it can guarantee the availability of the data at the requested destination. Finally, the WSS confirms this offer and the agreement is completed.

3 Case Study

To show the collaboration of DMS and WSS in more detail, a proof-of-concept implementation of a typical complex application in climate research – namely Stormtrack Analysis – has been build. In Stormtrack Analysis, filtered geopotential heights are examined in order to track and predict the movement of low-pressure areas over time with regard to given climate models. This is essential as storms and cyclones typically cross such areas [1].

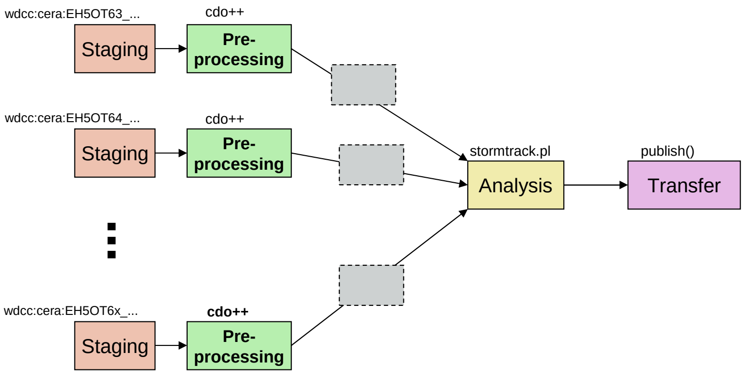

Figure 3: An exemplary dependency graph representing a workflow for Stromtrack Analysis. The dashed boxes denote implicit transfers, that is tasks inserted automatically by the WSS depending on the chosen processing sites without user interaction.

The workflow describing such an analysis can be broken down to a dependency graph as depicted in Figure 3. The result of this application is the output (in this case an image file) of the last step which is then presented to the submitter via the Portal component.

After submission, the WSS has to analyze the workflow description and generate corresponding tasks for each step. Although the number of tasks within the workflow might vary (depending on the amount of data requested by the user), four major steps can be identified for this type of application:

1. data preparation (staging and preprocessing)

2. data transfer (implicit or explicit)
3. data analysis (tool execution)
4. data publishing (result provision)

Obviously, data preparation is a prerequisite to data analysis. As such, the WSS first triggers the staging and preprocessing part via the DMS. That is, it requests a list of data providers (providers that offer a service which satisfies the specific data request). In the first implementation, a random selection from these is done; however, more complex selection schemes are obviously possible. For example, the WSS could select primarily those data providers which also offer compute resources at the same site (in order to minimize network data transfers). For the selected data provider, the WSS currently requests an offer obeying the *as soon as possible* rule only. This offer is also accepted, and the earliest completion time returned by the DMS is stored. An advanced implementation would request several offers for different providers and/or timeslots and select one out of these depending on a criteria set. This set could, for example, include *earliest availability*, *locality regarding free compute resources*, and others. The whole procedure is then repeated for each requested input files. Note that, at this point preparation is done in parallel for each data set.

Next, depending on the location of the prepared data and selected compute site, the WSS has to introduce *implicit transfer tasks*. These tasks are not user-specified, but inserted automatically by the WSS to ensure input data availability at the target processing site where the analysis will take place. The negotiation on these transfers is handled similarly to the data preparation step described above. The start time for a transfer task is equal to the completion time of the depending preparation task; the ending time is also recorded. Due to the structure of the case study workflow, the negotiation step of finding a suitable service provider can be omitted since both source and target of the transfer are already known. In more complex scenarios, the source and / or target for the data transfer step might not be known in advance or be ambiguous (because of data replication, multiple target processing sites, etc.). For the offer selection itself, again the simplest strategy (*as soon as possible*) is currently used.

In the current prototypic implementation of the DMS, offers are given based on static knowledge about the testing environment. In an advanced version it will use dynamic information from different sources: It will ask the data provider to give a time estimation for the retrieval and preprocessing of archive data, it will take existing replicas into account for selecting a transfer source, it will check how a new transfer would interfere with those already scheduled, etc.

The data analysis task is started with the arrival of the input data at the compute site. Finally the data publishing is an *explicit transfer task* copying the output data to a user defined location. This again is done corresponding to the aforementioned negotiation phase for data transfer.

4 Conclusion and Future Work

We developed a service oriented architecture for the distributed analysis of huge amounts of earth system data in a distributed Grid environment. Our strategy is a tight co-operation between workflow scheduling and data management, which find agreements over future transfers and availability of data. Furthermore, all components are aware of comprehensive ISO 19115 and ISO 19139 metadata. It identifies raw data at a technical and scientific level and can also contain provenance information. Furthermore, we demonstrated the applicability of our design by an implementation of the interfaces and a prototypic distributed execution of a typical earth science application.

In our next steps, we will focus on the analysis of the inherent unreliability of dynamic resource information.

Acknowledgements

We thank all partners in C3Grid for the intensive collaboration. This work was partly funded by the German Federal Ministry of Education and Research (BMBF).

References

1. M. L. Blackmon. A Climatological Spectral Study of the 500 mb Geopotential Height of the Northern Hemisphere. *Journal of Atmospheric Sciences*, 33:1607–1623, August 1976.
2. I. Foster and C. Kesselman. Globus: A Toolkit-Based Grid Architecture. In *The Grid: Blueprint for a Future Computing Infrastructure*, pages 259–278. Morgan Kaufman, San Mateo, 1st edition, 1998.
3. H. He. What Is Service-Oriented Architecture. online, September 2003.
4. The Open Archives Initiative. *The Open Archives Initiative Protocol for Metadata Harvesting*, protocol version 2.0 edition, June 2002.
5. Jason Novotny, Michael Russell, and Oliver Wehrens. GridSphere: An Advanced Portal Framework. In *Proceedings of the 30th Euromicro Conference*, pages 412–419. IEEE Press, September 2004.
6. International Organization of Standardization. ISO 19115:2003 Geographic Information – Metadata. TC 211; ISO Standards, May 2003.
7. Bertrand Portier. SOA terminology overview, Part 1: Service, architecture, governance, and business terms. Online: http://www-128.ibm.com/developerworks/webservices/library/ws-soa-term1/index.html, November 2006.
8. Uwe Schulzweida and Luis Kornblueh. *CDO Userś Guide, Climate Data Operators, Version 1.0.6*, 2006.
9. W. Smith, I. Foster, and V. Taylor. Scheduling with Advanced Reservations. In *Proceedings of International Parallel and Distributed Processing Symposium*, 2000.
10. Jia Yu and Rajkumar Buyya. A Taxonomy of Scientific Workflow Systems for Grid Computing. *SIGMOD Records*, 34(3):44–49, 2005.