Grid-Based Processing of High-Volume Meteorological Data Sets

Guido Scherp[1], Jan Ploski[1], and Wilhelm Hasselbring[1,2]

[1] OFFIS, Escherweg 2, 26121 Oldenburg, Germany
{guido.scherp,jan.ploski}@offis.de
[2] University of Oldenburg, Software Engineering Group, 26111 Oldenburg, Germany
Hasselbring@informatik.uni-oldenburg.de

Abstract. Our energy production increasingly depends on regenerative energy sources, which impose new challenges. One problem is the availability of regenerative energy sources like wind and solar radiation that is influenced by fluctuating meteorological conditions. Thus the development of forecast methods capable of determining the level of power generation (e.g., through wind or solar power) in near real-time is needed. Another scenario is the determination of optimal locations for power plants. These aspects are considered in the domain of energy meteorology. For that purpose large data repositories from many heterogeneous sources (e.g., satellites, earth stations, and data archives) are the base for complex computations. The idea is to parallelize these computations in order to obtain significant speed-ups. This paper reports on employing Grid technologies within an ongoing project, which aims to set up a Grid infrastructure among several geographically distributed project partners. An approach to transfer large data sets from many heterogenous data sources and a means of utilizing parallelization are presented. For this purpose we are evaluating various Grid middleware platforms. In this paper we report on our experience with Globus Toolkit 4, Condor, and our first experiments with UNICORE.

1 Introduction

Regenerative energy sources are becoming more and more important for our energy supply. It is assumed that in the future the dependency on these sources will increase. However, the availability of these sources is highly influenced by meteorological factors which impose new challenges. Forecast models for simulations are needed to provide a near real-time estimation of the power generation (e.g., through wind power). Another scenario is the determination of appropriate locations for building power plants. For example, an analysis based on (archived) solar irradiation data combined with further geographical and commercial information (lakes, rivers, costs, etc.) can be used to find optimal spots for solar power plants. Each simulation or analysis is based on large heterogeneous data sets, which come from satellites, earth stations, data archives, or other sources. Due to the large amount of data, the computational power of a single computer

W. Lehner et al. (Eds.): Euro-Par 2006 Workshops, LNCS 4375, pp. 235–244, 2007.

is insufficient. Next generation satellites with higher resolution will further increase that amount of data. Today, approximately one terabyte of new data per month is received and continuously archived that are relevant for our project WISENT, which is introduced in the following.

A solution for speeding up the simulations and analyses lies in utilizing parallel computation. Therefore, the challenges of transferring large amounts of data as well as the parallelization of simulations and analyses have to be addressed. In the context of parallel execution and transfer of large data amounts, the term Grid [1,2] has become more and more familiar in the last years. In the project WISENT (wisent.d-grid.de) Grid technologies are employed to focus on these challenges in the domain of energy meteorology. The University of Oldenburg (ehf.uni-oldenburg.de), three departments of the German Aerospace Center (DLR DFD, DLR TT, DLR IPA, www.dlr.de) and the company meteocontrol (www.meteocontrol.de) are collaborating on WISENT with the institute OFFIS (www.offis.de) as project coordinator. The project started in October 2005 and is funded by the German Federal Ministry of Education and Research (BMBF, www.bmbf.de) over three years. One aim of the project is to set up a Grid infrastructure to support large data transfers and distributed processing.

Each project partner has different resources that are to be shared within the Grid infrastructure. The different types of resources and their utilization are shown in Figure 1. Firstly, data are received from many heterogeneous sources such as satellites and earth stations. Based on this data, various complex computations are performed utilizing resources such as desktop PCs, multi-processor servers or a dedicated cluster. Because of the planned integration into the German Grid initiative D-Grid (www.d-grid.de), we also intend to utilize D-Grid resources, for example high-performance computing (HPC) centers. The results of these computations are used for multiple purposes, for example to monitor the energy output of photovoltaic modules. To gather experience, we examined several Grid middleware platforms such as Globus Toolkit 4, Condor, and we are currently evaluating UNICORE.

The challenges of transferring large data sets and parallelization in energy meteorology are described in Section 2. In Section 3, we report on possible solution paths we have examined or intend to examine, and experiments with Globus Toolkit, Condor, and UNICORE we have already performed or planned. We conclude with an overview of future work in Section 4.

2 Challenges to Be Addressed

In WISENT, we are faced with various challenges. In the present paper we focus on the transfer of large data sets and the utilization of parallelism to run complex computations on these data sets. To fulfill the requirements associated with it, we intend to examine appropriate Grid technologies. The challenges are described in more detail in the following subsections.

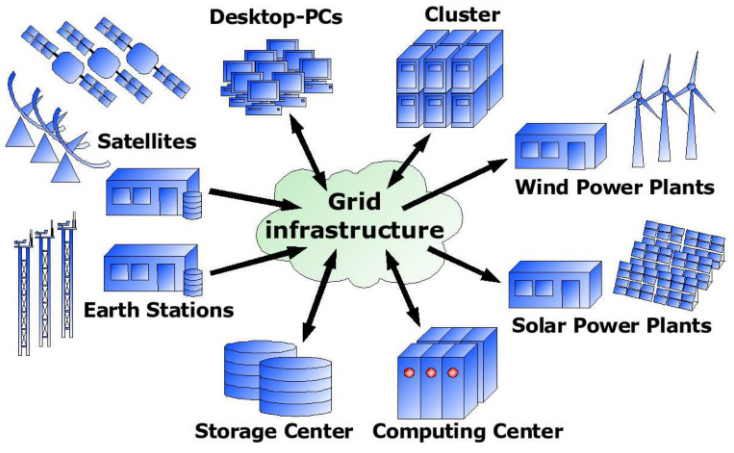

Fig. 1. Overview of the application scenario in WISENT

2.1 Transfer of Large Data Sets

One characteristic of the collaboration of our project partners is the exchange of data. Those data mostly consist of data products that are the results of computations on raw satellite data or previously processed data products. Data transfers are executed periodically (weekly, daily, hourly) or on demand. The number of daily data transfers at DLR-DFD concerning WISENT, for example, is up to 100, some of which are performed between the project partners and others which involve external facilities. The size of one transfer ranges from a few kilobytes up to several hundred megabytes. In the future, the number and size of data transfers will increase.

Today, a data transfer often relies on simple transfer protocols such as FTP or HTTP. Manual execution and monitoring are not uncommon. One goal is to execute these transfers reliably and automatically, which should result in less manual work and thus in a cost reduction. Each project partner intends to be able to easily share own data as well as access the data offered by any project partner within the Grid infrastructure. Furthermore, some transfers are used to deliver commercial data products. Thus, security for authentication and authorization as well as monitoring for accounting and billing are also required.

2.2 Parallelization

Our project partners use applications in the context of energy meteorology that can benefit from parallelization. The potential for parallel execution can be exploited in different ways, for example at data or program level. At data level, it means that the input data is first separated into several parts. The computation is next executed on distributed computing nodes, whereat the same program is running on each node without any network communication among the nodes.

Finally, after the parallel execution is completed, the corresponding output data parts are merged into the final result. In contrast, the parallelization at program level is used if a clear subdivision of input data for independent computation is not possible. Thus, parts of the programs are distributed and a frequent exchange of intermediate data is needed, for example via an implementation of the Message Passing Interface (MPI) [3]. The implementation of parallelization at data level is often much easier than parallelization at program level. Fortunately, we found out that most applications considered within our project can use parallelization at data level. No program sources have to be modified, which is an advantage because the source code of some programs concerned and libraries is not available. Our project partners already use parallelization at program level in a few applications. Thus, this parallelization technique will become relevant and has to be considered in the future.

3 Solution Approaches Employing Grid Technologies

Today, several Grid technologies and respective Grid middleware platforms are available. We gained initial experience with Globus Toolkit 4 and Condor and we are currently evaluating UNICORE, which we describe in the following sections. Our evaluation of each software considers the challenges mentioned in Section 2.

A typical classification of Grid infrastructures distinguishes Intra-, Extra-, and Inter-Grids (Figure 2). The term Intra-Grid refers to a Grid set up within a single organization. Several Intra-Grids of different organizations are connected to an Extra-Grid, which requires stronger security policies, such as virtual private networks (VPN). An Extra-Grid often provides access to a Grid infrastructure for a certain user group with established working relationships, thus it is typical for community grids. The final extension is Inter-Grid, a global Grid infrastructure for a wide range of independent users. The planned Grid infrastructure for our project conforms to the Extra-Grid definition. Grid technologies can be employed at Intra-Grid level, Extra-Grid level, or both levels, which is taken into account by our evaluation.

3.1 Globus Toolkit 4

As a start, we have chosen to evaluate the Grid middleware platform Globus Toolkit 4 [4]. This decision was made because of the comprehensive available documentation [5,6], the long development history since Globus Toolkit 1 (released in 1996), and the implementation of the Grid standard Web Services Resource Framework (WSRF) [7]. Globus Toolkit 4 offers comprehensive services for data transfers which motivates, due to the requirements of data transfers described in Section 2.1, the examination of these services. In this section we report on our first experiences.

Application Scenario. Our project partner DLR DFD is involved in most data transfers exchanging data products, which may be performed in two ways.

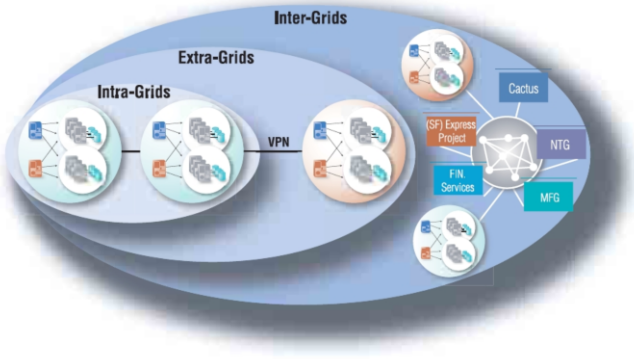

Fig. 2. Intra-, Extra-, and Inter-Grid (Source: IBM)

Either DLR DFD offers the desired data product for download on their "public" FTP server or the data product is transferred directly to an FTP server of a project partner or another external facility. This FTP-based approach has some disadvantages because errors are handled manually. For example, a transfer initiated by a cronjob might fail due to a temporary network timeout.

The manual error handling of failed FTP transfers gives rise to potential problems. The required data products are stored together with raw satellite data in an archive called *Digital Information and Management System* (DIMS), which is secured behind a firewall. Thus, a data product is periodically transferred to the "public" FTP server, which is located in a demilitarized zone (DMZ). The recipient is also periodically looking for changes on this FTP server and downloading the desired data products. This procedure does not include checks whether the required data products are completely available. If the series of uploads to the "public" FTP server was interrupted, an incomplete product might be downloaded, which might be discovered too late.

We have chosen five data transfers and their corresponding data products as an initial test scenario. Our approach is an incremental migration of each transfer toward Grid technologies in multiple steps. First, test data transfers are to be executed internally between DLR DFD and the OFFIS institute. Based on this experience, these transfers will be extended step-wise to cover the real-world scenarios.

Data Transfers with Globus Toolkit 4. Globus Toolkit 4 offers a layered architecture of services for data transfers. The GridFTP service belongs to the lowest layer. It is an enhanced FTP protocol with features such as the support of encryption and authentication based on X.509-certificates and several options to increase the data troughput, e.g. using several parallel data channels. The use of certificates allows Singe Sign-On (SSO), which is an important advantage. By default, GridFTP does not encrypt the data channel, but activation of encryption reduces the data throughput significantly. Globus Toolkit 4 comes with a

standalone GridFTP server, which supports so-called *Third Party Transfers* between two hosts, whereby the initiator of the transfer can be located anywhere. With `globus-url-copy`, Globus Toolkit 4 offers a command-line GridFTP client.

GridFTP provides only the basic service for transferring files from a source to a destination. Higher-level services are supposed to build upon this capability. One practical application of this idea is the WSRF-compliant Grid service named Reliable File Transfer (RFT). Besides several configurable properties, the RFT service accepts a list of files that are to be transported. The data transfer is processed in a reliable manner. That means that the RFT-Service retries interrupted transports up to a limited number of times before reporting an error. For this purpose, the state of each transport is recorded in a database with transactional control (such as PostgreSQL). Unfortunately, the RFT service in the actual version of GT4 is not utilizing the support of data channel encryption in GridFTP, but this feature is proposed in the coming versions.

In our test scenario, we installed Globus Toolkit 4 on a host at DLR DFD and configured it to provide the RFT service. For testing purposes our institute assumed the role of a Certificate Authority (CA) to support authentication with signed X.509-certificates. For the future we plan to use D-Grid certificates. The first data transfers from DLR to Offis failed due to the strong firewall policies at DLR. Based on the FTP-based protocol GridFTP uses a new dynamic port for each data channel it creates. Typically, these ports are in a higher range and blocked by the firewall. However, the port range used by both client and server can be restricted by environmental variables. Because of the strong firewall policies DLR develops an Application Level Gateway (ALG), a proxy placed in the DMZ to support transparent communication for using (Grid) software such as GridFTP. These experiences are the basis for the incremental extension of data transfers based on GridFTP and RFT. Also planned is the full integration of the DIMS archive into the Grid infrastructure. Finally, a detour via the "public" FTP server should be avoided.

Currently, we are investigating the Monitoring and Discovery System (MDS) of Globus Toolkit 4. Each Grid service has properties which can be exploited with MDS, for example the RFT service reports the number of active transfers and transferred bytes as properties. This information can be used for live monitoring, accounting and billing.

Globus Toolkit 4 also supports distributed computing. However, in the context of connecting desktop PCs and PCs within clusters, the features of Globus Toolkit 4 do not appeal to us. The main reason is the apparent lack of a good scheduling mechanism available out-of-the-box. However, we have not fully evaluated the Community Scheduler Framework (CSF) it offers yet. Moreover, Globus Toolkit 4 does not specifically support pooling resources of desktop PCs. For example, some useful features such as the discovery of mouse and keyboard activities and job migration are not available. Consequently, we are currently considering using Globus Toolkit 4 only at the Extra-Grid level to improve the execution of data transfers within the Grid infrastructure.

3.2 Condor

Several work packages in WISENT focus on improving computational performance through data parallelization. One promising approach in that context is provided by Condor [8], a freely available software package developed at the University of Wisconsin-Madison. In this section, we briefly describe a parallelization scenario for which Condor appears to be one likely solution based on the insights gathered so far.

Application Scenario. Two- and three-dimensional computational models of radiative transfers are currently employed by our project partners to determine the amount of irradiation reaching the Earth's surface based on satellite observation. Apart from images, the algorithms rely on input data consisting of various atmospheric parameters, such as cloud profiles and water vapor distribution.

The three-dimensional radiative transport solver MYSTIC [9] developed by DLR utilizes a compute-intensive Monte Carlo simulation to track individual photons through multiple layers of the atmosphere. The simulation time varies depending on the covered geographic area and the number of simulated photons, which affects the precision of the results. For example, a modestly sized simulation for a 100x100 km area with 10,000 photons consumes more than 5 hours of CPU time on a 1 GHz PC.

From the parallel programming point of view, an attractive feature of both the 2D and 3D radiative transfer models is the ability to compute results for multiple distinct columns of the atmosphere separately, with a certain amount of input data overlap. The results can be combined into one large output image.

Our project partner DLR IPA intends to reduce the computation time of radiative transfer models by harnessing the power of a Linux cluster consisting of sixteen 2,4 GHz Pentium Xeon nodes, several servers and 10 to 15 Linux desktop workstations. Based on experiences gathered in the project, the network will be expanded to support other resource-intensive scientific computations. The initial experiments will process only several gigabytes of locally available input data and produce a fraction of that as output, meaning that large-scale data transports will not be of primary concern.

While parallel 3D computation models are being implemented by our colleagues with background in physics and meteorology, our task is to assist them with the technology to improve their existing parallel computing solution for 2D radiative transfer. The currently deployed solution, used to drive the Linux cluster mentioned above, distributes libRadtran [10] solver processes to multiple cluster nodes through a combination of shell scripts implemented in-house, the Parallel Virtual Machine (PVM) package [11] and a third-party PVM-based extension of the Unix `make` tool called `ppmake`. This solution does not scale to utilizing desktop resources nor does it permit unsupervised execution. Problems such as difficult-to-explain non-termination of PVM processes are common and call for manual interventions. Finally, the present solution does not stand software engineering scrutiny. For example, the `ppmake` tool, which is suspected to cause the encountered problems, is no longer actively maintained.

The Condor Approach. Condor is a software package that has been designed with the aim to support "cycle scavenging", that is, taking advantage of computational resources during idle periods without interfering with their primary users. A typical Condor installation consists of a central coordinator machine and one or more machines acting as compute and/or job submission nodes. Both classes of nodes communicate with the coordinator to announce the type and availability of resources or the availability of jobs to be computed. On each compute node, a Condor process monitors and regularly reports to a coordinator static attributes such as the operating system and processor architecture, as well as dynamic information such as the current CPU load, amount of disk space and keyboard and mouse activity.

A job submission node announces resource requirements for user-submitted batch jobs, which consist of binary executables along with a specification of input/output files. When an available resource has been matched with a job request by the coordinator, it arranges direct communication between the submitting and computing node. A shadow process is started on the submitting node in order to monitor and report the execution status of the job process on the computing node; both processes communicate using a proprietary protocol.

A job may be interrupted if its current computing node becomes unavailable or at its owner's discretion. If the executable program was linked with the Condor library and adheres to some restrictions, Condor is able to migrate a snapshot of the job process to another compatible compute node and continues its execution there. Furthermore, I/O system calls made by executable programs can be automatically re-routed to the original submission machine, which may be used to eliminate the need for explicit data transfer operations before and after job execution. Alternatively, Condor offers built-in file transport mechanisms.

Our tests with Condor consisted of migrating the current PVM-based scripts to utilize the Condor submission client instead of ppmake. They performed to our satisfaction on a network of several different Linux PCs. The small amounts of data we utilized for the tests helped us understand Condor's opportunistic approach to scheduling. In fact, because individual fragments of computations were very short, the measured clock time for the Condor run exceeded that of a sequential execution. We were able to influence the loss factor by modifying the Condor configuration, particularly the communication intervals between individual processes. Further improvements are expected by adjusting the job granularity, as we have already discovered in experiments on behalf of another project partner.

Based on our experience so far, the main weaknesses of Condor appears to be the platform dependence, caused in part by the restricted availability of its source code and in part by its operating-system dependencies, the complexity reflected in a daunting number of (well-documented) configuration settings, and the demands it places on the network connectivity. Condor's main advantages are the flexibility, long product history, solid documentation, wide user base and built-in features which facilitate the construction of "desktop Grids", which we intend to utilize at the Intra-Grid level. Those "desktop Grids" are very

suitable for running our applications using parallelization at data level. Condor also supports parallelization at program level through MPI, but in this scenario the network bandwidth could quickly become a major bottleneck. Thus, the access to HPC centers is planned as well in order to support future demands for executing those applications within the Grid infrastructure.

3.3 UNICORE

Our evaluation of UNICORE has just begun. So far, we have successfully connected two UNICORE hosts and performed first tests using the UNICORE client. Thus, an assessment of whether we will use UNICORE at the Intra-Grid or Extra-Grid level is not yet possible. As UNICORE was originally used to connect distributed computing centers, its classification into the Extra-Grid category seems reasonable. One great advantage of UNICORE is that the communication between the client and the gateway is encrypted and uses only one fixed port for both job submissions and data transfers. This is very compliant with stronger firewall policies. Each connection of a GridFTP client in contrast needs at least one exclusive port for the data channel (c.f. Section 3.1). But a data transfer through the UNICORE Protocol Layer (UPL) has a noticeably less data throughput than GridFTP and thus it seems only suitable for small data transfers.

Another benefit of UNICORE is the possibility to model jobs and subjobs in a workflow-style. As mentioned in Section 3.1, the DIMS archive of DLR DFD stores both raw and post-processed satellite data. Some data products are computed regularly and stored in the archive for further purposes while others are computed on demand without persistent storage. The computation of each data product consists of several steps that form so-called process chains. However, as of today, each process chain is implemented individually and no consistent description format exists. We intend to assess the UNICORE client's utility for modeling and executing such process chains in WISENT. With the growing complexity of the process chains and their automation requirements, more sophisticated workflow elements supported by UNICORE's workflow description language may become useful. One disadvantage of the workflow model is that each workflow element must be bound to one specific resource (Virtual Site). For our scenario a dynamic resource selection with an optional limitation to specific resources ought to be possible as well.

4 Conclusions and Future Work

This paper has shown how the use of Grid technologies at different levels can address the challenges in the context of energy meteorology, including transfers of large data sets and parallelization of programs at data level. We are still in an evaluation phase, thus beside further tests with Globus Toolkit 4 and Condor we have to gain more experience with UNICORE for a more comprehensive comparison. Currently, we are also considering evaluating gLite [12] and the

commercial platform Sun N1 Grid Engine [13]. Another task is to investigate the interoperability between these Grid middleware platforms. It is very likely that we will use different Grid middleware platforms for the Intra-Grid and Extra-Grid levels, thus interoperability is important.

In the view of the considerable data heterogeneity, one of our next steps will be to support a uniform access method. To this end, we plan to examine the OGSA-DAI (Data Access and Integration) [14], which provides a Grid-related standardized access to different data resources using Web Services technologies.

Moreover, we plan to further examine security issues. A critical success factor for a Grid infrastructure lies in achieving trustworthiness [15]. The project partners need to be sure that the Grid infrastructure is secure, and they intend to control their own participating systems. Therefore, services for monitoring, access control and logging are required. Finally, easy access to the Grid infrastructure as well as fast installation and deployment of new Grid nodes are also important concerns.

References

1. Foster, I.: What is the Grid? A Three Point Checklist. Grid Today **VOL. 1 NO. 6** (2002)
2. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: Enabling scalable virtual organizations. Lecture Notes in Computer Science **2150** (2001) 2–13
3. MPI: Message Passing Interface. (`http://www-unix.mcs.anl.gov/mpi/standard.html`) Retrieved: 2006-06-11.
4. Globus: The Globus Alliance. (`http://www.globus.org/`) Retrieved: 2006-06-11.
5. Globus: Globus Toolkit 4.0 Release Manuals. (`http://www.globus.org/toolkit/docs/4.0/`) Retrieved: 2006-06-11.
6. Jacob, B., Brown, M., Fukui, K., Trivedi, N.: Introduction to Grid Computing. (`http://www.redbooks.ibm.com/redbooks/pdfs/sg246895.pdf`) Retrieved: 2006-06-11.
7. WSRF: Web Services Resource Framework. (`http://www.globus.org/wsrf/`) Retrieved: 2006-06-11.
8. University of Wisconsin: Condor High Throughput Computing. (`http://www.cs.wisc.edu/condor/`) Retrieved: 2006-06-11.
9. Mayer, B.: I3RC phase 1 results from the MYSTIC Monte Carlo model. (I3RC (Intercomparison of 3D Radiation Codes) workshop, Tucson, Arizona, 1999)
10. Mayer, B., Kylling, A., Hamann, U.: libRadtran – library for radiative transfer. (`http://www.libradtran.org`) Retrieved: 2006-06-11.
11. Geist, A.: PVM: Parallel Virtual Machine: A Users' Guide and Tutorial for Network Parallel Computing. MIT Press (Scientific and Engineering Computation) (1994)
12. EGEE: gLite Lightweight Middleware for Grid Computing. (`http://glite.web.cern.ch/glite/`) Retrieved: 2006-06-11.
13. Sun: Sun N1 Grid Engine. (`http://www.sun.com/software/gridware/`) Retrieved: 2006-06-11.
14. OGSA-DAI: Open Grid Services Architecture - Data Access and integration. (`http://www.ogsadai.org.uk/`) Retrieved: 2006-06-11.
15. Hasselbring, W., Reussner, R.: Toward trustworthy software systems. IEEE Computer **39**(4) (2006) 91–92