
University of Bonn, Department of Geography

GRIDIFICATION OF OGC WEB SERVICES: CHALLENGES AND POTENTIAL

M. Sc. Alexander Padberg, Prof. Dr. Klaus Greve

Abstract: Grid resources might provide a foundation for meeting the increasing demand for storage capacity and computing power in next-generation SDIs. This article gives an overview of the current state of research regarding combinations of grid service and OGC Web Service (OWS) technology. Because of the inherent complexity of grid computing it is necessary to simplify the access of grid resources for potential users. Therefore, the article describes approaches developed in the GDI-Grid project to bridge architectural gaps between both technologies without changing the well-known service interfaces specified by the OGC. After highlighting fundamental differences of grid infrastructures and SDIs, suitable methods for grid-enabling OGC processing (i.e. Web Processing Service) and data services (i.e. Web Feature Service, Web Coverage Service) are shown.

Keywords: Spatial Data Infrastructure, SDI, grid computing, OGC, OWS, WPS, GDI-Grid

// GRIDIFIKATION VON OGC WEBDIENSTEN: HERAUSFORDERUNGEN UND POTENTIAL

// Zusammenfassung: Grid-Ressourcen können eine Möglichkeit bieten, den steigenden Anforderungen an Speicherplatz und Rechenleistung in modernen Geodateninfrastrukturen zu begegnen. Dieser Artikel gibt einen Überblick über den gegenwärtigen Forschungsstand bezüglich der Verbindung von Grid-Diensten und OGC Web Services (OWS). Aufgrund der Komplexität von Grid Computing ist es notwendig, potenziellen Nutzern die Verwendung von Grid-Ressourcen so einfach wie möglich zu machen. In diesem Artikel werden Ansätze zur Anbindung von Grid-Technologien an Geodateninfrastrukturen beschrieben, die im GDI-Grid-Projekt entwickelt wurden. Die vom OGC spezifizierten Service Interfaces bleiben in den vorgestellten Ansätzen erhalten. Nach einer Beschreibung der grundlegenden Unterschiede zwischen Grid-Infrastrukturen und GDIs werden Methoden zur Gridifizierung von OGC Prozessierungs- (Web Processing Service) und Datendiensten (Web Feature Service, Web Coverage Service) geschildert.

Schlüsselwörter: Geodateninfrastruktur, GDI, Grid Computing, OGC, OWS, WPS, GDI-Grid

Authors

M. Sc. Alexander Padberg
Prof. Dr. Klaus Greve
University of Bonn
Department of Geography
Meckenheimer Allee 166
D-53115 Bonn
E: {apadberg, klaus.greve}@uni-bonn.de

1. INTRODUCTION

With improving spatial and temporal resolution the size and complexity of spatial data sets generated by advanced sensors is steadily increasing. Accessing and processing these vast amounts of data in adequate execution times requires resources beyond the capacities of most conventional spatial data infrastructures (SDI). In communities like the hydrological or geological communities there is also a demand for reducing the execution times of highly complex processes that have a plethora of spatial parameters (f.e. processing digital elevation models covering large areas). For these reasons other types of infrastructures employing distributed resources, such as cluster architectures and grid infrastructures, become a viable option for storing and processing geospatial data (Reed 2008). Unlike local cluster infrastructures the grid (Foster 2002) provides nearly unlimited scalability in conjunction with low initial costs. Organizations only pay for the time, they actually use the resources. Standardized access mechanisms support finding and accessing relevant data in complex and widely distributed archives. Purchasing potentially expensive hardware for computing-intensive tasks is thereby rendered unnecessary. In our research linking grid concepts and SDI concepts serves as a starting point for the development of geospatial cloud computing concepts.

This paper gives an introduction on utilizing grid infrastructures for geospatial purposes. In our ongoing research in the GDI-Grid Project we combine grid concepts with SDI concepts and implement SDI components (i.e. the Open Source framework deegree, Fitzke et al. 2004) in a grid middleware environment (Globus Toolkit 4, Foster 2006a). The GDI-Grid Project (GDI being the German acronym for Spatial Data Infrastructure) is part of the D-Grid initiative, the national German grid initiative, which strives for making grid infrastructures accessible to users from academia and industry. Participants of the GDI-Grid project include providers as well as users of data and services. In the project we provide the users of an OGC-compliant SDI with a whole set of generic grid services including data services and catalogue services, that make use of the superior storage and computing resources of a grid infrastructure while being addressed through the commonly known OGC service interfaces. Thus the complexity of the grid is kept hidden from the user.

Our work focuses on grid services that implement the Web Services Resource Framework and are deployed inside a Globus Toolkit Middleware. The GOWS Working Group comprised of the projects CYCLOPS, GENE-SI DR and DORII is developing similar service implementations for the glite grid middleware. Furthermore, work on a grid-enabled processing service is also part of the recently finished OWS-6 testbed. Results from these initiatives are incorporated into our research. Other related works include Chen et al. (Chen et al. 2006) as well as Fritzsche and Hiller (Fritzsche, Hiller 2006). While the former examines the integration of grid computing in earth monitoring systems, the latter aims at using grid technologies in the climate change community.

2. DIFFERENCES BETWEEN OGC-COMPLIANT SDIS AND GRID INFRASTRUCTURES

There are several important differences between the paradigms of standards-compliant SDIs and grid infrastructures (Hobona et al. 2007). The standards used in grid infrastructures are defined by the Open Grid Forum (OGF, www.ogf.org), while the Open Geospatial Consortium (OGC, www.opengeospatial.org) develops the specifications for Spatial Web Services, which form the foundation for SDIs. SDIs provide data and services through standardized service interfaces. Furthermore, the OGC's Web Processing Service (WPS) specification allows for the integration of data manipulation and processing components.

While an individual data service instance is able to connect to several data backends, a single backend is usually made up of a single data source. In grid services on the other hand, the service is able to manage the access of distributed data resources with redundant data storage for increased availability. Using grid infrastructures and their superior transmission capabilities, access times might be reduced significantly. Processing services on the other hand may also be grid-enabled. Processes from conventional WPS instances are executed on a single resource, while grid-enabled processing services are able to split a process execution over a multitude of computing nodes. For problems that can be parallelized a simultaneous execution on more than a single resource could speed up the execution significantly. In the geospatial domain there are processes that can be parallelized through the use of simple spatial tiling mechanisms

on the input data. These processes are the most obvious candidates for gridification (i.e. modifying the process logic so instead of a single computing node a multitude of spatially distributed computing nodes work together to calculate the result of a process).

Matching the different kinds of paradigms is challenging. While both kinds of infrastructures are commonly realized as service-oriented architectures, the standards used differ significantly. The architecture of grid infrastructures is defined in the Open Grid Service Architecture specification (OGSA, Foster et al. 2006b). It is built on and extends the widely known Web Services Architecture (Booth et al. 2004).

At the time the OGC started creating specifications for service based SDIs, the Web Service (WS-*) standards were not yet established. Therefore, the OGC had to develop their own approach for creating service-oriented data distribution. Because all OWS (OGC Web Services) are based on this approach, WS-* and OWS are not directly compatible.

Before presenting an implementation approach for the gridification of geospatial processing, the fundamental gaps between the concepts of the OGC and the OGF are examined.

2.1 SERVICE DESCRIPTION

Services deployed in a grid infrastructure have to be described using Web Service Description Language (WSDL) documents. In this XML format services are described "as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information" (WSDL, www.w3.org/TR/wsdl.html). When the first OGC specifications were developed, WSDL was not yet a standard. Therefore, the OGC also created their own service description method. OGC Web Services return their according metadata in capabilities documents, the format of which is defined in service-specific schema documents.

To deploy an OGC-compliant service in a grid infrastructure, it is necessary to manually create a WSDL document from the service's capabilities, as there is not yet a way to convert a capabilities document into a WSDL description automatically.

2.2 SERVICE INTERFACE

There are several different methods to invoke the operations provided by an OGC

Web Service. Preferred ways are key-value-pair requests via HTTP-GET and XML-encoded requests via HTTP-POST. Grid services accessed through a grid middleware are addressed using SOAP (www.w3.org/TR/soap). While only younger OGC service specifications include instructions for using SOAP for service invocation, services that do not support SOAP, may not be integrated into a grid workflow.

Furthermore, the OGC specifies a set of possible operations for every OGC Web Service. For example the operations for a WPS are GetCapabilities, DescribeProcess and Execute. Other OWS provide a similar set of operations. Outside of the OGC domain these types of service requests are not well-known though.

2.3 STATEFULNESS

Grid services are stateful services, they depend on the ability to store state information like intermediate results for later use. OGC specifications were developed in a tradition of stateless environments and protocols. While statefulness and the integration of SOAP and REST (Representational State Transfer, Fielding 2000) interfaces are under discussion, OGC specifications do not yet include a general definition of how to handle state information. The only service able to store intermediate results is the Web Processing Service (WPS, OGC 2007). An optional part of the WPS specification allows results from a WPS process execution to be stored at an external resource and used as input data in a later service call. Because of the ability to store state information the WPS specification is used as a starting point for our research.

2.4 SECURITY

Grid environments are very powerful processing tools that have to be protected against misuse. Therefore, security is a major concern in grid infrastructures (Foster et al. 1998). Without mechanisms for encrypted communication and proper methods for authorization and authentication grid computing shall not be possible. In the D-Grid environment it is effectively forbidden to use grid resources without a proper level of security. Security in grid infrastructures utilizing the Globus Toolkit Middleware is implemented using the Globus Security Infrastructure (GSI, Foster et al. 1998).

The OGC does not yet provide specifications on how to establish security in conventional SDIs. Currently security is handled in a project- or vendor-specific way. When integrating OGC Web Services in grid infrastructures it is essential to provide mechanisms to address encryption, authentication and authorization meeting the requirements of the GSI.

3. IMPLEMENTATION OF A GRID-ENABLED WPS

The WPS specification (OGC 2007) basically provides users with an interface to invoke (geospatial) processes. Neither type nor complexity of the processes are specified, only the set of possible operations and their invocation methods are defined, thus providing process developers with maximum flexibility. Operations provided by the WPS include GetCapabilities, returning metadata about the service, such as information about the service provider and a list of supported processes, and DescribeProcess, returning metadata about specific processes. The Execute-operation is used to invoke one of the processes provided by the service instance.

3.1 GRIDIFICATION OF THE SERVICE

To utilize the superior storage and computing power of the grid, the implementation approach for a grid-enabled WPS has to address the differences described in the previous chapter. The gridification has to be performed on the process level to ensure that the WPS implementation and the service interface are not affected at all. Thus it is guaranteed, it won't be necessary to modify the requests sent to the service, when connecting the WPS to the grid.

To grid-enable the deegree3-WPS, we split the WPS process into two parts (see figure 1): One part inside the grid infrastructure plus a conventional WPS interface. For this we created a grid service using the Grid Development Tools from the Marburg Ad-Hoc Grid Environment middleware (MAGE, mage.uni-marburg.de). This tool generates a skeleton for a grid service, where only the process logic needs to be inserted. If the final WPS process is for example supposed to transform a digital elevation model into a triangulated irregular network (TIN), the according methods and functions have to be implemented in this grid service. The service

is then deployed into a Globus Toolkit container, acting as a runtime environment. To maximize the potential benefit of utilizing grid computing resources, the implementation should provide the means for parallel execution of the process logic on a multitude of worker nodes.

The other part of the grid-enabled process is set up like a conventional WPS process, that is a Java process class and a configuration document defining process-specific metadata, such as input and output parameters, are created. No process logic apart from an invocation of the grid service is inserted into this process class. This WPS process now acts as a client invoking a grid service.

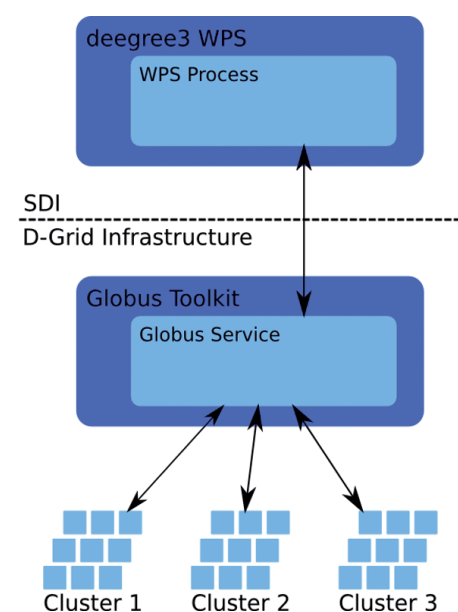


Figure 1: Model for a grid enabled WPS

By using the MAGE's Grid Development Tools to create the grid service, the required WSDL document is created automatically. Splitting up the WPS process into a conventional process and a grid service solves the problems with service description, service interfaces and statefulness, as the grid service is used to address the internal grid resources while the user still only communicates with an OGC-compliant WPS.

3.2 IMPLEMENTATION OF SECURITY FEATURES

Every invocation of a service, deployed in a Globus Toolkit Container, must be assignable to exactly one user. As the GSI (Globus Security Infrastructure) is based on public-

key cryptography, such a grid service may only be invoked by a user with valid grid credentials. To simplify the usage of grid services from an OGC-WPS an online credential repository called MyProxy (Novotny et al. 2001) is employed. A user creates proxy-credentials at the MyProxy-repository using his or her long-lived grid credentials and assigns a username-password combination to these proxy-credentials. The WPS process from the grid-enabled WPS is extended to include a MyProxy-call, a corresponding JavaAPI is part of the Globus Toolkit. When sending a request to the WPS the user includes the username-password combination in the request. This identification is then used by the WPS process to retrieve a short-lived credential from the MyProxy-repository, which in turn may be used to invoke grid services.

The use of short-lived credentials in grid projects is further evaluated in the Gap-SLC project, that is also part of the D-Grid initiative.

4. OTHER OGC WEB SERVICES

Besides the WPS there are other OGC Web Services that might benefit from gridification. Furthermore, if a WPS instance is to work efficiently in a grid environment, it has to access data sources using the grid's capacities. In the GDI-Grid Project a concept for grid-enabling OGC-compliant data services similar to the gridification of processing services was developed. Both the Web Feature Service (WFS) and the Web Coverage Service (WCS) were altered to use a grid datastore. For this task the OGSA-DAI (Open Grid Services Architecture Data Access and Integration, www.ogsa-dai.org.uk) middleware is used.

The WFS from the deegree framework is extended to include an OGSA-DAI datastore implementation. This datastore implementation creates an SQL-statement, that is sent to an OGSA-DAI datastore set up inside a grid infrastructure. Thus a user can invoke the grid-enabled WFS in an OGC-compliant way and still access data stored inside the grid.

The prototype for a grid-enabled WCS, that was developed in the GDI-Grid Project, uses OGSA-DAI's GridFTP implementation. The data served by the WCS consists of several files deployed inside the grid. When a GetCoverage operation is invoked the grid-enabled WCS obtains the necessary files

through a GridFTP activity. The URL's of the hosts where the files are deployed are specified in the WCS configuration. As spatial queries are not supported by GridFTP, whole files are transmitted to the grid-enabled WCS, that extracts the requested sections.

Figure 2 shows an example for a SDI using both grid-enabled processing services as well as grid-enabled data services. On the left side of the figure there is a conventional WPS, executing a process on a single compute resource. The service in the middle utilizes a multitude of compute resources inside a grid infrastructure. The WFS on the right side accesses storage resources inside and outside the grid infrastructure. All three services are invoked by the SDI user using conventional OGC-compliant requests.

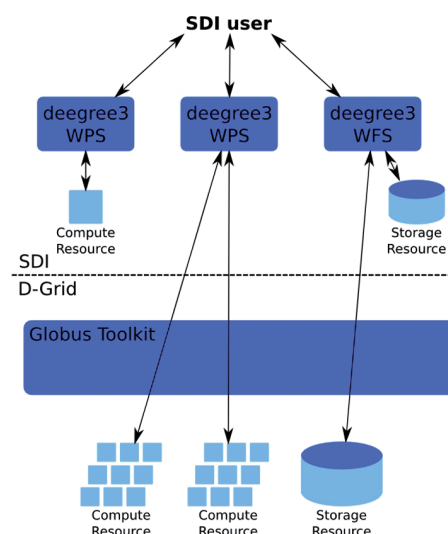


Figure 2: Architecture of an SDI including grid-enabled services

5. CONCLUSION

The aforementioned method for grid-enabling OGC Web Services is a straightforward approach encapsulating a grid service invocation inside a conventional OGC-compliant service. By utilizing the know-how of grid specialists and the MAGE framework it is possible to create the according grid services semi-automatically. Thus it is easy to modify existing conventional data services or processing services to integrate resources from grid infrastructures. At the moment this integrated approach is useful, because it is possible without changing the current specifications. There is not yet a finalized proposal

for a more sophisticated approach on combining OWS and grid services developed by the OGC or OGF. Further research on how to combine the paradigms at the specification level is necessary. For an evaluation of the potential of grid computing for the geospatial data community however the proposed low-level approach is a feasible first step.

6. OUTLOOK

The impact of grid-enabled OGC Web Services on the performance of geospatial applications is not yet clear. The available series of measurements do not yet suffice, to determine the actual benefits of utilizing grid technologies in the geospatial domain. Grid technology is a powerful tool to organize complex and effective cloud computing environments. It is assumed that with an increasing number of processing units and an increasing amount of data, that has to be processed, the gain in terms of speed justifies the additional overhead associated with grid computing. Further data with algorithms of varying complexity needs to be collected. Thus a function for the determination of the point where the speed gain outweighs the overhead may be established (see figure 3).

Potential approaches for improving the performance in conventional SDIs are similar to the methods used in Web Services Architectures. The methods focusing on efficiency and reliability include increasing the available bandwidth, improving server availability and response time, caching of requests and intermediate results. Whether some of these or other methods to address performance issues are feasible in a grid-enabled SDI is a topic for future investigation.

Whether gridification is feasible for a WPS process depends on whether there is a suitable way to parallelize its process logic. There is no generic parallelization method that fits every geospatial process. Therefore, it is not possible to create the process logic of the grid service belonging to a grid-enabled WPS process automatically. But some general approaches on parallelization of geospatial processes do exist. Practicable alternatives for splitting up the calculation of the process results include partitioning the input data sets by object type, by geometry type, by operation type, by spatial criteria and by Level of Detail (Chaudry and

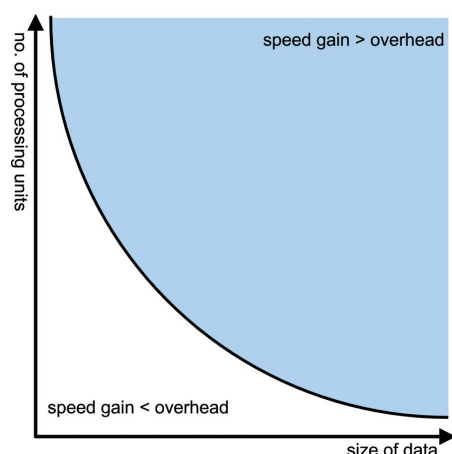


Figure 3: Estimation of a breakeven function

Mackaness 2008, Werder and Krüger 2009). Spatial tiling is often a suitable method for dividing input data (f.e. 3 dimensional laser scanning data sets) into several data sets. The input data are split over a raster (regular tiling) or according to administrative, geomorphologic or other criteria into a set of tiles with a specified overlap at the

borders of the tiles (Lanig et al. 2008). It is very important to apply partitioning with great care, as the assembled process results from different partitioning mechanisms may vary (Chaudry and Mackaness 2008). Furthermore, the most suitable partition type depends not only on the input data but also on the process. The data sets are distributed among the computing nodes and processed individually. The result of the process invocation is assembled from the node's individual results. Due to this additional assembly step some computational overhead is created. This overhead has to be taken into account when deciding whether to execute a process on the grid or not. Although the spatial tiling approach fits very well to grid computing, it is not applicable for all geospatial processes. Especially complex processes with high correlation between different parts of the input data can't be sped up this way, because dependencies influencing intermediate results of different nodes have to be applied before the final result is assembled.

Apart from the technical challenges there is yet a similarly important task for grid-enabling SDIs. At the moment most of the users from the geospatial domain are not fully aware of grid computing technologies and the potential advantages of their use. Either the technology is not known at all or perceived as being unneeded or too complicated. Therefore, it is necessary to conduct further studies for identifying use cases that are prone to benefit from distributed computation and storage. The results from these studies then have to be promoted to the geospatial community while at the same time making sure there is a simple way for this community to utilize grid technologies.

ACKNOWLEDGEMENTS

The work presented in this paper is part of the GDI-Grid project (Spatial Data Infrastructure Grid) funded by the German Federal Ministry of Education and Research (BMBF). The support is gratefully acknowledged. ◀

References

- Booth, D.; Haas, H.; McCabe, F. (2004): Web Services Architecture. W3C, Working Group Note, 2004.
- Chaudry, O.; Mackaness, W. (2008): Partitioning Techniques to Make Manageable the Generalisation of National Spatial Datasets. In: 11th ICA Workshop on Generalisation and Multiple Representation, 2008.
- Chen, A.; Di, L.; Wei, Y.; Bai, Y.; Liu, Y. (2006): An optimized grid-based, OGC standards-compliant collaborative software system for serving NASA geospatial data. In: Proceedings of the Thirtieth Annual IEEE/NASA Software Engineering Workshop SEW-30, p. 159–166, 2006.
- Fielding, R. (2000): Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine, 2000.
- Fitzke, J.; Greve, K.; Müller, M.; Poth, A. (2004): Building SDIs with Free Software – the deegree Project. In: Proceedings of GSDI-7, 2004.
- Foster, I.; Kesselman, C.; Tsudik, G.; Tuecke, S. (1998): A Security Architecture for Computational Grids. 5th ACM Conference on Computer and Communication Security, 1998.
- Foster, I. (2002): What is the Grid? A Three Point Checklist, Argonne National Laboratory & University of Chicago, 2002.
- Foster, I. (2006a): Globus Toolkit Version 4: Software for Service-Oriented Systems. IFIP International Conference on Network and Parallel Computing, 2006, p. 2–13.
- Foster, I.; Kishimoto, H.; Savva, A.; Berry, D.; Grimshaw, A.; Horn, B.; Maciel, F.; Siebenlist, F.; Subramaniam, R.; Treadwell, J.; Von Reich, J. (2006b): The Open Grid Services Architecture Version 1.5, 2006.
- Fritsch, B.; Hiller, W. (2006): Das Collaborative Climate Community Data and Processing Grid – Gridtechnologien für die Erdsystemforschung. In: Strobl, J.; Blaschke, T.; Griesebner, G. (Hrsg.): Angewandte Geoinformatik 2006, p. 171–176, 2006.
- Hobona, G.; Faribairn, D.; James, P. (2007): Workflow Enactment of Grid-Enabled Geospatial Web Services. 6th UK e-Science All Hands Meeting, 2007.
- Lanig, S.; Schilling, A.; Stollberg, B.; Zipf, A. (2008): Towards Standards-Based Processing of Digital Elevation Models for Grid Computing through Web Processing Service. In: Gervasi, O. et al.: ICCSA 2008, Part II, LNCS 5073, pp. 191–203, 2008.
- Novotny, J.; Tuecke, S.; Welch, V. (2001): An Online Credential Repository for the Grid: MyProxy. Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing, 2001.
- OGC (2007): OpenGIS Web Processing Service, OpenGIS standard: OGC 05–007r7, 2007.
- Reed, C. (2008): Grid computing in the geospatial web: the role of standards. In: Proceedings of the 15th ACM Mardi Gras Conference: From Lightweight Mash-Ups To Lambda Grids, 2008.
- Werder, S.; Krüger, A. (2009): Benefit from Parallelization in Grid-Computing. In: AGILE 2009 Pre-Conference Workshop: Grid Technologies for Geospatial Applications, 2009.