Article under Review for the International Journal of Spatial Data Infrastructures Research, Special Issue GSDI-11, submitted 2009-03-01

Towards a grid-enabled SDI: Matching the paradigms of OGC Web Services and Grid Computing*

Alexander Padberg[1], Christian Kiehle[2]

[1]Institute of Geography, University of Bonn, Meckenheimer Allee 166, 53115 Bonn, Germany. apadberg@uni-bonn.de
[2]lat/lon GmbH, Aennchenstr. 19, 53177 Bonn, Germany. kiehle@lat-lon.de

Abstract

Spatial Data Infrastructures (SDI) have been widely adopted within the last years to provide access to spatial data and offer components to visualize maps based on distributed data. The implementation of Open Geospatial Consortium (OGC) specifications adds a layer of abstraction and interoperability to share and use data and spatial information among different communities. Integration of OGC-compliant interfaces into commercial off the shelf geospatial software is an indicator for the maturity of the standards beyond scientific scope. Whereas the usage of distributed data inside local Geographic Information Systems (GIS) is widely adopted, the distributed processing of data is not yet a common task. The recently defined Web Processing Service (WPS) specification strives to distribute geoprocessing functionality among the web. The paradigm of Grid computing has evolved parallel to distributed spatial information systems. The according standardization organization, the Open Grid Forum (OGF), is responsible for the development of specifications for Grid computing in general. A memorandum of understanding between the OGC and the OGF has been signed in 2007 (Lee 2008, Reed 2008) and should foster the information interchange as well as the development of specifications for spatial applications on the Grid. This way the geographic community is enabled to benefit from the superior storage and computing capacities of Grid infrastructures. The process of actually matching these two technological approaches leads to several conceptual and technical challenges, since both specification baselines differ significantly. The current paper examines the conceptual differences between the paradigm of OGC Web Services and Grid computing and provides an integration approach to overcome the gaps between SDIs and Grid infrastructures. Using the example of a grid-enabled WPS the steps needed to utilize the computing power of the Grid from an OGC context are explained. Furthermore an outlook of the potential of other grid-enabled OGC Web Services will be given together with an analysis on the challenges that arise during their implementations.

Keywords: SDI, OGC, WPS, WFS, WCS, Grid computing, Globus Toolkit

1 TOWARDS DISTRIBUTED SPATIAL INFORMATION SYSTEMS

A shift from Geo Information Systems (GIS) towards Spatial Data Infrastructures (SDI) has occurred behind the scenes. During the last decade data collections have been centralised and are available remotely, often as services. In Europe, initiatives like INSPIRE build the political framework to share geodata across organizations, boundaries and even worldwide. The technical framework is built on international standards as defined by the Open Geospatial Consortium and the International Organization for Standardisation (ISO). While a GIS provides components to input, manage, analyse and present (IMAP) spatial data, a SDI is first and foremost capable of the tasks data management and presentation. It is likely that most of the tasks related to data input are either done in situ (i.e. by capturing primary data on-site) or on a local workstation using advanced editing tools. Both cases do not take advantage of an SDI's means of data input, although the captured data will most likely be integrated into an SDI. The most unutilized task of a GIS inside SDIs is the aspect of analysis.

Analysis in spatial information systems means processing spatial data into information. To estimate the impact of a building activity to the environment, spatial data sets of different sources have to be analysed and processed. Processing can be very trivial, like calculating a buffer zone around a street segment, or very complicated, like the calculation of a groundwater model. In either case, a complicated model may be reduced to a certain number of tasks, worked off in a predefined sequence. A GIS is capable of processing and model building. A number of processes is integrated in most GIS, like buffer, union, Map Algebra (Tomlin 1990), or point-in-polygon-analysis. Chaining processes and spatial data leads to models, which can be processed using GIS. Depending on the amount of spatial data and the complexity of the single processes, the task of analysis requires immense computing power. Until now, most SDIs do not implement the task of analysis.

Reasons for the missing analysis components inside SDIs are several technical as well as organizational issues. Until end of 2007 there has been no standard for integrating processes or calculations inside Spatial Data Infrastructures. Solutions were implemented based on the actual requirements and involved techniques like custom-tailored implementations for Remote Procedure Calls (RPC), SOAP or proprietary solutions by major GIS/SDI vendors. In 2007 the OGC defined the Web Processing Service standard (WPS) to share processes in an OGC-compliant way. A process may be a single processing task or a bundle of tasks encapsulated inside a process. WPS promises to close the analysis gap for future SDIs.

Since openness is a characteristic of SDIs, not every organisation wants to share their processes or make them freely accessible. Security and billing mechanisms have to be implemented to prevent unauthorized access to processes (and data). To further advance the concept of SDIs the following aspects should be taken into account:

- How can processes be shared among organizations (securely)?
- How can huge amounts of data be shared among organizations (securely)?
- How can performance be optimized? (cf. Scholten 2006)
- How can small-and-medium-sized organisations share high-quality SDI components?

2 IMPLEMENTATION OF GRID-ENABLED SDI COMPONENTS

The integration of Grid technologies for spatial information systems has been explored in some research projects, but is not part of any known productive infrastructure. Chen et al. 2006 discuss integration of Grid technology for earth monitoring systems while Fritzsch & Hiller 2006 adopt Grid computing for climate change sciences. Anyway, an entire standards-compliant Spatial Data Infrastructure has not been implemented in the Grid.

This chapter provides an overview of the incompatibilities of SDIs and Grid infrastructures. The second part of the chapter shows a way to bridge the gap between these technologies by presenting a prototypical grid-enabled SDI.

2.1 Differences between OGC Web Services and Grid Services

The concept of Grid computing has been termed in the 1990ies by Ian Foster. “Grid computing has emerged as an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation.” (Foster 2001). It's relation to the concept of Spatial Data Infrastructures lies in the terms distributed computing, large-scale resource sharing and high-performance. It is conceivable that SDIs will reach limits in terms of resources when accessing Earth Observation data inventories or data bases of national and international data centres. This is especially true, whenever large-scale data sets have to be processed. The calculation of a continental Normalized Differenced Vegetation Index (NDVI) requires Map Algebra on multiple terabytes of data, a task which cannot be handled by local workstation computers. A Grid, providing processing power of thousands of Central Processing Units (CPU), seems to be the ideal platform to provide large scale processing capabilities. Potential advantages are:

- Means to share storage and computing resources as a virtual organization, therefore reducing the initial costs of hardware acquisition
- Means to enhance storage and computing resources on demand

- Means to handle all transactions securely

Because the concepts of conventional SDIs and Grid infrastructures differ significantly (Hobona 2007), an actual implementation of a grid-enabled SDI has to address several incompatibilities. Especially the subjects

- service description
- service interfaces
- stateful services
- security mechanisms

are handled in fundamentally different ways. A description how they are dealt with in the respective infrastructures is given below.

2.1.1 Service description

While Grid services always come with a Web Service Description Language document (WSDL, www.w3.org/TR/wsdl), OGC web services are described using Capabilities documents as well as service-specific metadata for different operations (e.g. DescribeFeatureType, DescribeCoverage etc.). WSDL documents and Capabilities documents differ significantly and are not easily converted into each other. To deploy an OGC web service in a Grid infrastucture it is necessary to create a WSDL descripition manually. There is not yet a method for doing this automatically.

2.1.2 Service interfaces

OGC service specifications define the set of operations a service supports. There are different ways to invoke these operations, but the preferred possbilities are key-value-pair requests via HTTP-GET and requests encoded in XML documents sent via HTTP-POST. Additionally most of the younger service specifications include instructions to utilize SOAP for invoking an operation.

Grid services are usually addressed through a Grid middleware. The Globus Toolkit 4 for example delivers service calls using SOAP. Services that do not support SOAP or have no WSDL-description may not be integrated in a Grid workflow.

2.1.3 Stateful services

Apart from using SOAP for communication and WSDL for description, some Grid services implement the Web Services Resource Framework (WSRF, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf) developed by the Organization for the Advancement of Structured Information Standards (OASIS). While conventional web services, including OGC web services, are stateless, the WSRF enables a service to manage state informations. These values are stored as resources at a so-called service endpoint. Every service endpoint has its own Uniform Resource Identifier (URI) that may be used to access the stored resources. Storing intermediary results necessary for further calculations is a common use case for stateful services.

OGC specifications apart from the WPS do not yet include any instructions regarding stateful services. An optional part of the WPS specification provides a request parameter for storing the results of a process at an external resource. But as this part of the specification is optional it is not widely supported at the moment.

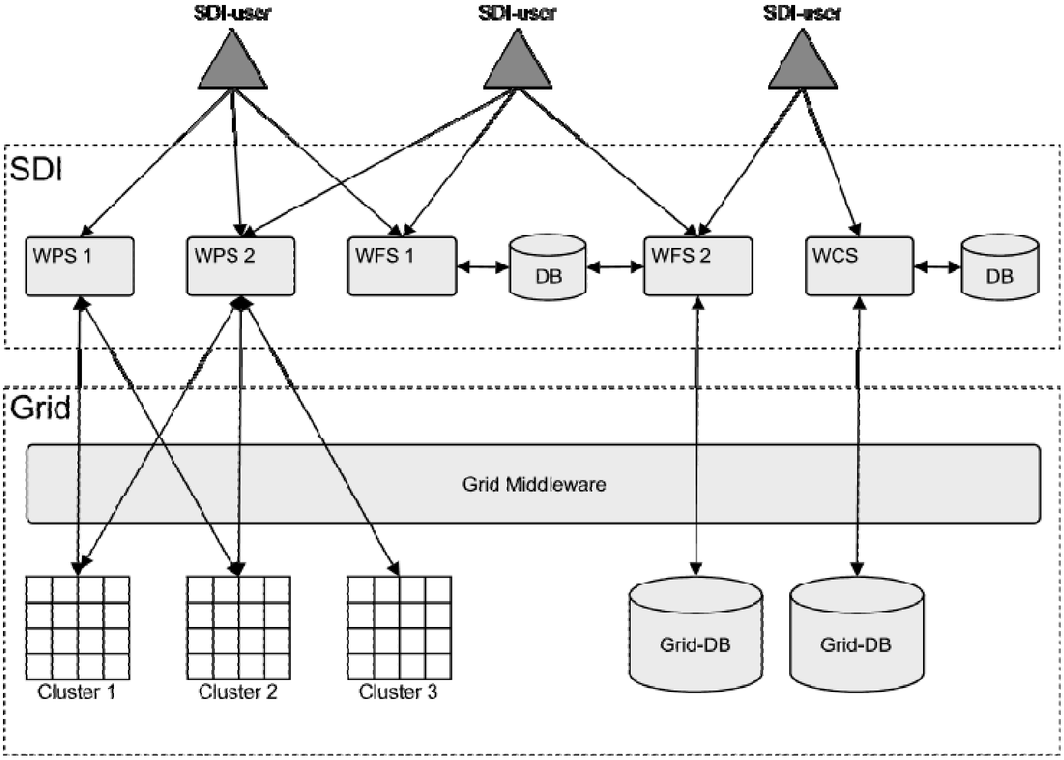
2.1.4 Security mechanisms

OGC specifications include no statements regarding security issues. Securing the transport protocol is usually done using HTTPS. Furthermore there is no specification on how to authenticate different users at a service. Security mechanisms in conventional SDIs are established project-specifically and different vendors handle security in different ways. For Grid infrastructures this low level of security does not suffice. It is necessary, that every resource access can be assigned to exactly one particular user, as the vast amounts of computing power and storage capacity of Grid infrastructures posess the potential for misuse. Therefore security is a significant element of Grid infrastructures (Foster 1998). If OGC web services are to be used in Grid infrastructures they have to provide the means to perform user authentication as well as encryption for communicating with Grid resources.

2.2 Prototype of a grid-enabled SDI

The following sections cover the prototypical implementations done to overcome the mentioned differences between conventional SDIs and Grid infrastructures. The aim of these implementations is to enable different aspects of a SDI to work in a Grid environment. All of the prototypes are based on the Open Source deegree framework (Fitzke et al. 2004). deegree is a java framework supplying all the main components for building a SDI. The Grid middleware used is Globus Toolkit 4 (Foster 2006). Figure 1 gives a thorough overview of the use cases the grid-enabled SDI shall support.

Figure 1: Logical View of SDI-Grid-Architecture



2.2.1 Data Exchange

The standards-compliant retrieval and distribution of data is one of the fundamental functionalities of an SDI. OGC standards for accessing different kinds of data have been around for several years and are widely adopted by the geospatial community (Kiehle 2007). Data backends for these services range from spatial databases like Oracle Spatial or PostGIS to file-based data like shapefiles. Accessing data backends that reside inside a Grid infrastructure is not yet addressed by OGC standards. With the increase in temporal and spatial resolution due to improvements in data acquisition technologies like sensors the size of spatial data sets is growing. While storing such huge amounts of data might be expensive, transmitting them for visualisation or analysis could lead to serious bottle-necks in a geospatial workflow. To utilize the vast storage capacities and the superior transmission speed provided by a Grid infrastructure while preserving the service interface (i.e. set of operations, request format) the exisiting implementations of the Web Feature Service (WFS) and the Web Coverage Service (WCS) specifications were modified.

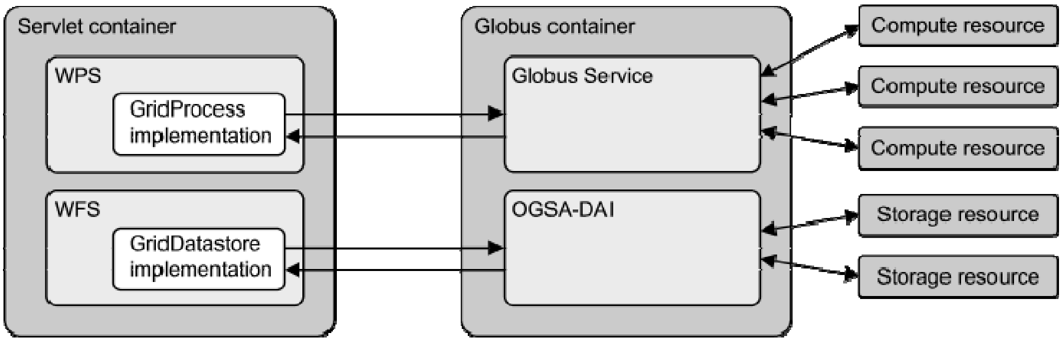
The WFS (OGC 2005) is used for accessing vector data and presenting it in GML-format. The types of data backends a WFS may access depend on the implementation of the service. The deegree WFS for instance supports PostGIS-, Oracle Spatial-, shapefile-, ArcSDE-backends as well as being able to access other WFS instances and forward their results to the user. It is possible to create additional kinds of datastores for the deegree WFS by extending the `Datastore`-class from the package `org.deegree.io.datastore`. For a non-transactional WFS the only method to implement here is `performQuery( Query query, MappedFeatureType[] rootFts )`. After updating the according properties document, the WFS can access this new custom datastore.

To create a grid-enabled WFS a custom datastore that combines an OGC-compliant WFS-frontend with a grid-datastore-backend was implemented (see Figure 2), the so-called `GridDatastore`. The WFS-frontend is covered by a deegree WFS, the datastore of the prototype is a PostGIS-database located inside a Grid infrastructure. For accessing relational databases via the Grid the OGSA-DAI-middleware can be used (www.ogsadai.org.uk). Elements of OGSA-DAI were integrated into the custom `GridDatastore` thus enabling the WFS to access a PostGIS-database via the Grid. The datastore implementation acts like a OGSA-DAI-Client passing an SQL request to the PostGIS-database.

The prototype is a non-transactional WFS, so InsertFeature-, UpdateFeature- and DeleteFeature-operations are not supported yet.

Extending the WCS (OGC 2003) to access data inside a Grid infrastructure was done using the same methods, but a different data backend. Instead of collecting data from a PostGIS database, OGSA-DAI was used to retrieve data in raster format from a coverage file via GridFTP. GridFTP extends the widely-known FTP-standard to include Grid-specifics like security mechanisms and parallel data transport (Taylor 2004).

Figure 2: Conceptual Deployment Diagram



2.2.2 Data Processing

Although not as widely adopted as data retrieval and exchange the processing of spatial data is one of the functionalities, that may be provided by a SDI. OGC's main resource for the description of standardized data processing is the Web Processing Service specification published in 2007 (OGC 2007). The specification provides the flexibility to implement processes with any degree of complexity, mainly defining the set of available operations to describe and invoke these processes. Highly complex processes with large input data sets require vast amounts of computing power to finish in a reasonable span of time. As this degree of computing power may not be readily available at a single workstation, technologies like Grid computing become a viable option.

An implementation of the WPS 1.0.0-specification is part of the deegree framework and was used as basis for the prototype of a grid-enabled WPS. The design of the deegree WPS enables users to implement their own processes and deploy them without changing the service's implementation itself. The gridification of the WPS is based on splitting the process implementation in two parts. At first a Grid service, that is going to be deployed into a Globus Toolkit 4 container, is created using the MAGE Toolkit developed at the University of Marburg (http://mage.uni-marburg.de). The GT4 container acts as a runtime environment for Grid services. The actual logic of the final process is now inserted into this Grid service. If for example the process is aimed at calculating a buffer polygon around a polyline, the actual buffering has to be implemented in this Grid service. To maximize potential benefical effects of using Grid compute resources, the actual implementation should utilize parallelisation, so the calculation may be split up across several worker nodes (see Figure 2). Submission of jobs to worker nodes can be done using a Grid Resource Allocation Manager (GRAM).

The second part of the grid-enabled WPS consists of a conventional WPS process, that contains a service call to the aforementioned Grid service along with all grid-specific security settings. Apart from this service call the WPS process contains no further process logic. This way the process acts as a Grid-client, invoking the Grid service whenever an Execute-request is sent to the WPS (see Listing 1).

Listing 1: WPS process invoking a Grid service running in local GT4 container

```
String serviceURI = "https://127.0.0.1:8443/wsrf/services/GridService";
EndpointReferenceType endpoint = new EndpointReferenceType();
endpoint.setAddress(new Address(serviceURI));
GridServiceAddressingLocator locator = new
      GridServiceAddressingLocator();
GridPortType gridService = locator.getGridPortTypePort(
      new URL(serviceURI));

// Setting level of security
((javax.xml.rpc.Stub)gridService)._setProperty(
      org.globus.wsrf.security.Constants.CLIENT_DESCRIPTOR_FILE,
      CLIENT_DESC);
((javax.xml.rpc.Stub)gridService)._setProperty(
      org.globus.gsi.GSIConstants.GSI_TRANSPORT,
      org.globus.wsrf.security.Constants.ENCRYPTION);

// Executing the process logic of the grid service
DoBuffer buffer = new DoBuffer(parameter1, parameter2, parameter3);
DoBufferResponse response = gridService.doBuffer(buffer);
```

Apart from enabling a user from the geospatial community to take advantage of the resources offered by a Grid infrastructure, the presented approach at grid-enabling a WPS has another possible value. By inserting the whole process logic inside a Grid service deployed into a globus container, this service may be included in a series of service calls creating a complete workflow. Such a workflow might be made available through an OGC-compliant WPS itself, thus offering a possibility to invoke entire chains of service calls in a standardized way.

2.2.3 Security Mechanisms

While there are not yet mandatory security specifications published by the OGC, security is one of the main concerns in Grid computing (Foster 1998). Without following precise security rules Grid computing is not possible in a meaningful way, as resource providers need to be given some level of control over who is using their resources for how long. Furthermore if authentication and authorization of users are not taking place, accounting and billing are rendered impossible.

The Globus Toolkit Security Infrastructure (GSI, Foster 1998) is based on public-key cryptography. GSI uses X.509 certificates for the confirmation that a given public key belongs to a certain user. The open source software MyProxy is used for managing credentials (i.e. Certificates and private keys). A Java API for communicating with MyProxy servers is part of the Globus Toolkit.
MyProxy is an online credential repository for the Grid (Novotny 2001). Web services or Grid portals, that do not initially support the GSI, can easily be empowered to interact with Grid resources using MyProxy. When utilizing

MyProxy for a portal or web service, a user is required to create a proxy-credentials at the MyProxy-repository using his long-lived Grid credentials. The user specifies a username and a password for this proxy-credential. From now on until the proxy-credential has expired, this credential in combination with its according username and password can be used to generate short-lived Grid credentials, which in turn grant access on the same resources as the user's long-lived certificate, albeit for a shorter span of time.

As OGC Web Services do not integrate any means of interacting with Grid security systems, the aforementioned deegree implementations for WPS, WFS and WCS had to be enhanced to communicate with a MyProxy-repository. Whenever a WPS containing a process, that utilizes Grid compute resources, or a WFS or WCS serving a datastore, that represents a Grid storage resource, is deployed, a new `MyProxy`-object is instantiated. Every time either a request is sent to the WPS process or the WFS/WCS datastore is addressed, the `get(String username, String passphrase, int lifetime)`-method of this `MyProxy`-object is executed, returning a `GSSCredential`. This credential is created from the proxy-credentials matching the given username and passphrase. The `GSSCredential` is valid for the given lifetime in seconds and may be used in this time period to access Grid resources.

The OGC service specifications also do not yet include a standard way for inserting username-password-combinations into requests. The prototypes include these parameters as vendor-specific-parameters attached to the list of parameters in key-value-pair requests. Sending requests via SOAP (www.w3.org/TR/soap/) is also a possible option, in this case the username-password-combinations are included inside the header of the SOAP-document.

3 THE GDI-GRID-PROJECT

The implementations presented in the previous chapter were created as a part of the GDI-Grid-Project (GDI is the german acronym for spatial data infrastructure), a project the authors of this paper are involved in. In this chapter the project and the research topics addressed therein will be introduced along with some possible use-cases for the research results.

The GDI-Grid-Project's goal is to merge the components of a standard-compliant SDI with a Grid infrastructure. Combining these technologies serves two purposes: On the one hand it will enable users of SDI-technologies to access the superior storage and compute resources of a Grid infrastructure in a standardized way, on the other hand it will enable users of Grid technologies to integrate geospatial service calls into their workflows. The project is funded by the german ministry of education and research, the BMBF (www.bmbf.de). It started 2007 and will continue until 2010. Project participants include several universities, the computing center of Lower Saxony and commercial partners ESRI, lat/lon GmbH and Stapelfeldt GmbH.

3.1 GDI-Grid Objective

One part of the project aims at adapting the architectural building blocks of spatial data infrastructures and Grid infrastructures to enable communication between them. By combining the knowledge of members of the geo-spatial community and members of the Grid community the challenges of matching both worlds are addressed from two angles: examining how to extend Grid components to address OGC Web Services and identifying ways how to modify OGC Web Services, so they may communicate with Grid resources.
Further research topics addressed include the creation and validation of automated Grid workflows, that contain calls to geospatial services, as well as developing methods for data preprocessing using Grid compute resources. Integration of data from different sources, generalization and enrichment of data are some of the preprocessing steps that are going to be implemented as WPS processes.

Possible use cases for the research results are addressed in scenario working groups. The scenarios represent tasks that could also be fulfilled by conventional SDIs thus being adequate for the determination of potential advantages of Grid based SDIs. A detailed overview of the scenarios is given below.

3.1.1 Noise propagation

Calculation of noise propagation is a valuable instrument for urban planning and due to the EU-directive 2002/49/EC (Cox 2002) a necessity when constructing highways or railways. This scenario aims at transferring existing workflows for creating elaborate simulations of noise propagation into the Grid. Especially inside cities the propagation of noise is a complex phenomenon with a multitude of influencing factors. Due to the vast amounts of data and the extensive algorithms involved, the need to accelerate the computation of an exhaustive acoustic simulation arises. The approach to speed up this task followed in the GDI-Grid-project, consists in dividing the investigation area into a set of tiles. These tiles are processed individually, thereby splitting up the calculation, so it can be run on several compute resources at the same time. Therefore the workflow needs to be modified to include the necessary preprocessing and postprocessing steps i.e. splitting and merging of data sets. In addition to the Grid-enablement of the service, the creation of an appropriate user interface is planned. The simulation of noise propagation is done using the “LimA” software (www.stapelfeldt.de) developed by Stapelfeldt GmbH.

3.1.2 Flood simulation

TU Hamburg-Harburg is adapting an existing application for the generation of flood forecasting models (www.tu-harburg.de/wb/forschung/software_kalypso.html) to operate inside a Grid environment. Flood forecasting models are used to determine the extent of flooding events, allowing local authorities to do an estimation of the possible damages or facilitating effective early warning measures for residents of the area at risk. In urban regions not only the underlying terrain but also a detailed 3d city model has to be taken into account when simulating floods. For an accurate simulation these data sets are needed in a very high spatial resolution, increasing the amount of data to be processed exponentially. During the project a sophisticated mechanism for parallelizing the computation to speed up the generation of flood forecasting models is sought.

3.1.3 Emergency routing

Routing algorithms based on high-quality data tend to be highly complex. In a typical disaster management scenario the complexity of routing increases because of constant changes in trafficability of danger areas. Propagating plumes of toxic gases or spreading fires may result in arterial roads being impassable thus rendering long standing plans for evacuation useless. In such circumstances speeding up routing algorithms becomes necessary to guarantee up-to-date evacuation plans. Incorporating real-time sensor measurements and simulations like the flood forecasting model described in the previous paragraph furthermore increase the complexity of routing algorithms and the amount of relevant data. In such a scenario Grid computing might prove to be one way to satisfy the need for additional compute resources. It has to be noted that the gain in speed comes with a caveat: Because of the way multiple users are sharing the same Grid resources, actual real-time-applications are not possible using Grid technologies at this stage. For reliable real-time-applications new scheduling algorithms ensuring the needed quality of service have to be developed.

4 EVALUATION AND OUTLOOK

This work shows a way to combine storage and computing backends from Grid infrastructures with an OGC-compliant service frontend, enabling SDI-users to access the vast storage and processing capacities the Grid has to offer. By preserving the conventional service interfaces, the complexity for the users stays comparably low. While OGC's data access services WFS and WCS benefit from the storage resources provided by a Grid infrastructure, the data processing service WPS might benefit from the computational resources.

A combination of SDI and Grid components has been shown and implemented. This implementation overcomes the main obstacles faced along the way towards a grid-enabled SDI: it integrates a common Grid middleware (Globus Toolkit 4)

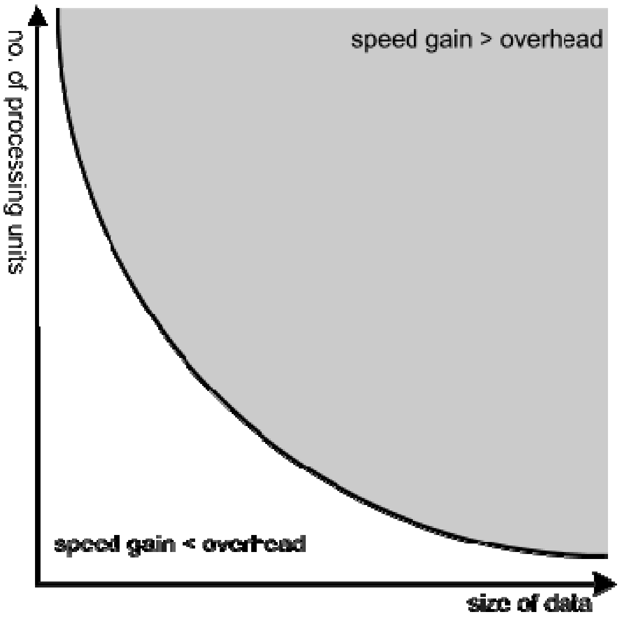
while still adhering to OGC standards. Following this approach the underlying complexity of the Grid infrastructure is encapsulated.
Since the OGC service infrastructure does not implement secure web services, especially the integration of Grid Security Infrastructure (GSI) has been a crucial task. Additional harmonization between Grid- and spatial standards has to be undertaken.

While the advantages of Grid computing sound promising for large data sets, the usage of Grid technologies comes with a lot of overhead. Authentication and GSI-integration delay common request-response-cycles. This is particularly significant for processing rather small data sets, especially when there are no parallel algorithms available. The parallelisation of processes is complicated since there are several problems to solve. Is the process itself capable of being parallelised? If not, is it sensible to infer spatial parallelisation by dissecting spatial data and stitching the results after processing?

A meaningful performance evaluation is not accomplishable inside the current Grid environment, since all processing jobs are scheduled by the Grid middleware depending on the current workload. In some cases, processing jobs will reside for a couple of hours inside a stack before being processed. This fact also hinders the integration of real-time data provided by sensors. Anyway it is likely that large amounts of data and complex processing routines lead to situations where Grid infrastructures are superior to standard computing plattforms (see figure 3). This “break-even-point”, where the speed gain achieved compensates for the overhead induced through Grid technologies, has to be defined based on case studies. Furthermore the type of the function for the determination of this point is not yet certain. The number of processors, that perform a parallel computation, and the size of the input data being the input parameters for this function, it could be linear, as well as an exponential or logarithmic function.

The mentioned drawbacks, especially the need to do a sophisticated parallelization of the process logic, should not cover the fact, that a distribution of a calculation over a multitude of computational resources can be used to decrease processing time significantly. But this gain in speed is not a Grid-specific improvement. The same results could also be achieved using distributed computing mechanisms provided by a local cluster. There are other characteristics that set Grid computing apart from using a local cluster. The most significant difference between conventional clusters and Grid computing is the ability of the Grid to provide a far larger number of computational resources as the need arises. If a given process can be split up into thousands of subprocesses that may be executed independently, a Grid infrastructure could possibly execute all subprocesses simultaneously thus significantly speeding up the process as a whole. A cluster on the other hand has a natural upper limit of the subprocesses that can be executed simultaneously. The same scaling advantages of Grid infrasructures apply to storage resources. Furthermore Grid infrastructures not only offer a high level of security, it is one of the main concepts of Grid computing. Without security Grid computing would not be possible at all, therefore Grid infrastructures inherently support sophisticated security mechanisms. To establish a comparable level of security in local clusters takes a lot of time and effort.

Figure 3: Estimation of a break-even function



Another concept inherent to Grid computing is the definition of virtual organizations (VO). VOs consist of a group of users working together on the same data. This data is not accessible for any user that is not a part of the VO. Inside a VO it is possible to define special roles like administrator with a set of rights different from those of a standard user. The creation of a VO is a valuable tool for facilitating the cooperation of users, who work on the same project but not in the same place. Whether these different concepts outweigh the additional overhead induced by Grid computing has to be decided based on each particular case.

REFERENCES

Chen A, Di L, Wei Y, Bai Y, and Liu Y (2006). An optimized grid-based, OGC standards-compliant collaborative software system for serving NASA geospatial data. *Proceedings of the Thirtieth Annual IEEE/NASA Software Engineering Workshop SEW-30 (SEW'06), Columbia, Maryland: 159–66.*

Cox, P.; Matas I Palou, J. (2002). Directive 2002/49/EC of the European Parliament and of the Council of June 2002 relating to the assessment and management of environmental noise, *Official Journal of the European Communities, Luxembourg*.

Fitzke, J.; Greve, K.; Müller, M.; Poth, A. (2004). Building SDIs with Free Software – the deegree Project, *Proceedings of GSDI-7, Bangalore.*

Fritzsch, 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. Beiträge zum 18. AGIT Symposium, Salzburg. S. 171-176.*

Foster, I.; Kesselman, C.; Tsudik, G.; Tuecke, S. (1998). A Security Architecture for Computational Grids, *5th ACM Conference on Computer and Communication Security.*

Foster, I.; Kesselman, C.; Tuecke, S. (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications, 15(3): 200-222.*

Foster, I. (2006). Globus Toolkit Version 4: Software for Service-Oriented Systems, *IFIP International Conference on Network and Parallel Computing, LNCS 3779: 2-13.*

Hobona, G.; Faribairn, D.; James, P. (2007). Workflow Enactment of Grid-Enabled Geospatial Web Services. *6th UK e-Science All Hands Meeting 2007, Nottingham, U.K.*

Kiehle, C.; Greve, K.; Heier, C. (2007): Requirements for Next Generation Spatial Data Infrastructures – Standardized Web Based Geoprocessing and Web Service Orchestration. *Transactions in GIS 11(6): 819-834.*

Lee, C.; Percivall, G. (2008). Standards-Based Computing Capabilities for Distributed Geospatial Applications. *Computer 41(11): 50-57.*

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.*

OGC (2003). Web Coverage Service (WCS) Implementation Specification, *OpenGIS standard: OGC 03-065r6.*

OGC (2005). Web Feature Service Implementation Specification, *OpenGIS standard: OGC 04-094.*

OGC (2007). OpenGIS Web Processing Service, *OpenGIS standard: OGC 05-007r7.*

Reed, C. (2008). Grid computing in the geospatial web: the role of standards. *Proceedings of the 15th ACM Mardi Gras Conference: From Lightweight Mash-Ups To Lambda Grids: Understanding the Spectrum of Distributed Computing Requirements, Applications, Tools, infrastructures, interoperability, and the incremental Adoption of Key Capabilities. Baton Rouge, Louisiana.*

Scholten, M.; Klamma, R.; Kiehle, C. (2006). Performance evaluation of spatial data infrastructures for geoprocessing. *IEEE Internet Computing* 10(5): 34–41

Taylor, I. (2004): *From P2P to Web Services and Grids: Peers in a Client/Server World.* Springer Verlag.

Tomlin, C.D. (1990): *Geographic Information Systems and Cartographic Modelling.* Englewood Cliffs, NJ, Prentice Hall