Grid Computing Enabled Web Processing Service

Bastian Baranski[1]

[1]Institute for Geoinformatics / University of Münster
Bastian.Baranski@uni-muenster.de

Abstract. The OGC Web Processing Service (WPS) specification defines a standardized interface to publish and perform geospatial processes over the web. Such a process can range from a simple geometric calculation to a complex simulation model. When computation large amounts of data and do complex calculations, the use of Grid Computing and related methods and technologies are a good choice for achieving high performance. At first glance, the WPS and its needs of computation resources is a perfect candidate to benefit from Grid Computing. This paper presents an architecture of a grid-enabled WPS. The presented proof-of-concept implementation demonstrates that Grid Computing and related methods and technologies increase calculation performance and improve service availability. Furthermore this paper presents an detailed outlook of interesting topics for further research activities.

1 Introduction

Current existing Spatial Data Infrastructures (SDI) are mainly focused on data retrieval, data processing and data visualization (Kiehle et al. 2006). An open standard based SDI mostly supports the retrieval and visualization of data through web services. But the data processing is normally done by human actors with more or less proprietary and monolithic Geo Information Systems (GIS). Some projects attempt to integrate stand-alone geoprocessing applications and their expert functionality into a web service environment and attempt to enable web services to execute geoprocessing tasks. One of these projects is the OGC Web Processing Service (WPS; see OGC 2007a). The WPS specification defines a standardized interface to publish and perform geospatial processes over the web. Such a process can range from a simple geometric calculation (e.g. a simple intersect operation) to a complex simulation process (e.g. a global climate change model).

When processing large amounts of data and do complex calculations (e.g. when doing a weather forecast simulation), situations with a lack of computing power may appear and so there is a need for advanced programming and data processing techniques. In such situations the use of Grid Computing or related methods and technologies are a good choice for achieving high calculation performance, for improving service availability and for guaranteeing different Quality of Service (QoS).

Even though Grid Computing or Distributed Computing are no new approaches, only little research had been done on combining OGC Web Services (OWS) with such a manner. Some research projects can be found at (Aloisio 2005), (Chen 2005), (Di 2005) and (GDIGRID 2008).

This paper presents a proof-of-concept implementation of a grid-enabled WPS which aims at increasing performance and improving availability of the underlying service. There is no need of making changes to OGC WPS specification or to existing OGC WPS compliant client applications. Chapter 2 provides an overview of the fundamental technology concepts behind this paper. Chapter 3 describes different categories of gridification of an OWS. Chapter 4 provides the architecture and the programming interface of the presented proof-of-concept implementation. Chapter 5 provides an overview of interesting topics for further research activities.

2 Background

This chapter introduces the OGC WPS specification and different meanings of the term 'Grid Computing'.

2.1 OGC Web Processing Service (WPS)

The OGC Web Processing Service (WPS) became an official OGC standard in late 2007. The specification defines a standardized interface to publish and perform geospatial processes over the web. Such a process can range from a simple geometric calculation to a complex simulation model. The WPS interface is based on three operations. The *GetCapabilities* operation provides service metadata and general informations about the offered processes. The process metadata including the corresponding input- and output-parameters are provided by the *DescribeProcess* operation. Via the *Execute* operation it is possible to run an offered process. The implementation of specific processes must be realized by the service provider itself

2.2 Grid Computing

This section introduces some meanings of the term 'Grid Computing'.

2.2.1 Definition

The term 'Grid Computing' is a diffuse phrase and there are many definitions available. This lack of a sole definition leads to many people working with Grid technology and having different views on what a Grid is.

Here are two examples of common used definitions of the term 'Grid':

> „A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive

> access to high-end computational capabilities.“, Foster and Kesselman (1998)
>
> „Whereas the Web is a service for sharing information over the Internet, the Grid is a service for sharing computer power and data storage capacity over the Internet. The Grid goes well beyond simple communication between computers, and aims ultimately to turn the global network of computers into one vast computational resource.“, CERN (2008)

Following the argumentation of Foster (2002), the term Grid Computing must be evaluated in „terms of the applications, business value, and scientific results that it delivers, not its architecture“. Foster captured a couple of definitions in his famous three-point-checklist of (technical) capabilities, according to which a Grid is a system that:

1. coordinates resources that are not subject to centralized control
2. using standard, open, general-purpose protocols and interface
3. deliver nontrivial qualities of service

There exist no global grid infrastructure that really fits all attributes and requirements described earlier. But there exist a lot of methods and technologies which are used when solving problems addressed by the given definitions of the term 'Grid'.

After this very short introduction, there are still some unanswered questions. Is a local cluster with a grid middleware on top a Grid? What actually makes a software or a service a “Grid software” or a “Grid Service“? Where to draw the line between Web Services and Grid Services?

3 Gridification of OGC Web Services (OWS)

This chapter describes different categories of gridification of an OGC Web Service (OWS). The term 'Gridification' means the adaption of existing applications and services to the requirements and expectations of a grid environment.

Following the argumentation of Krüger (2008), it is possible to distinguish between two categories of gridification of an OWS. In a simple approach the existing application or the existing service (in this context an OWS) stay primarily unchanged and the grid is used as a computation- or a data-resource (a low level gridification; see Figure 1a). It is possible to do calculation tasks distributed in the grid and to use grid services for accessing data. In this scenario the existing not-gridified application becomes the bottleneck because it doesn't obtain all qualities of a general grid service.

This kind of gridification is very easy to realize and the implementation is nearly independent of the underlying grid middleware. In a more complex approach the existing application or the existing service (in this context an OWS) is full embedded into the grid middleware (e.g. as a stateful service inside WS-Resource Framework, WSRF) and obtains all qualities of a general grid service (a high level gridification). It is possible to do calculation tasks distributed in the grid and to use grid services for accessing data. This solution is not easy to implement and there is a need of a proxy to stay OWS compliant and to handle the communication (see Figure 1b).

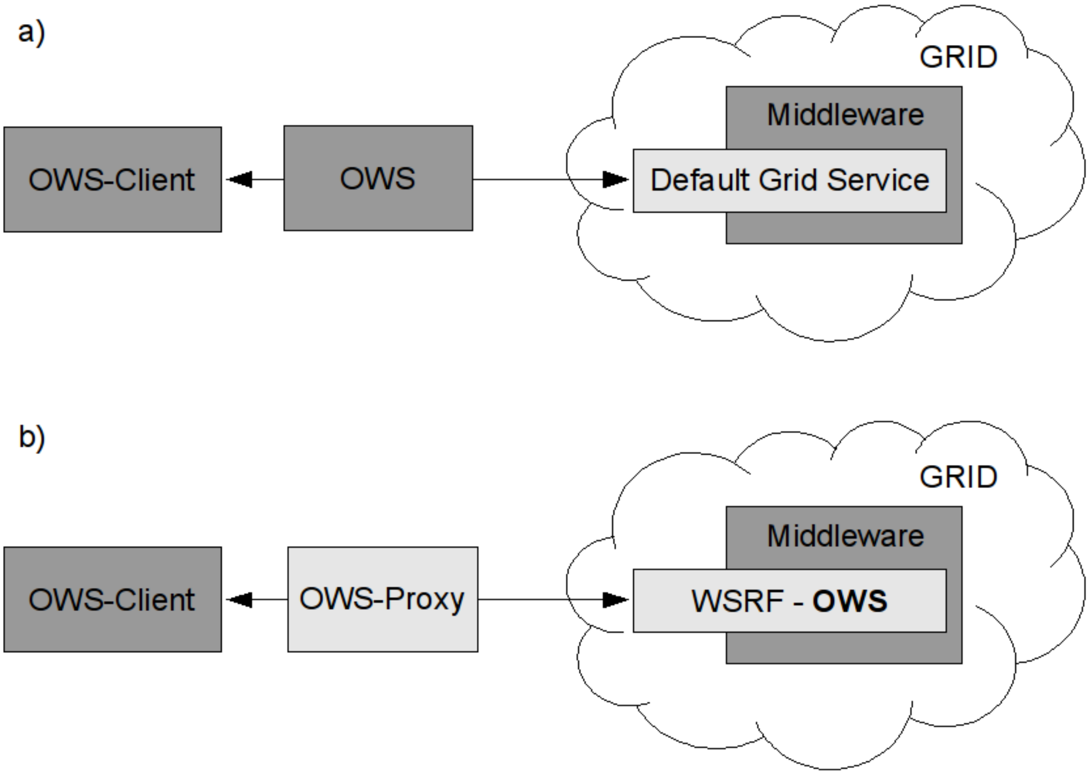


Figure 1: It is possible to distinguish between two categories of gridification of an OWS: a) low level gridification and b) high level gridification.

In context of complex service chains, you can create a more detailed classification for each gridification level (see Krüger 2008). Within our approach we used the grid as a computation resource to increase calculation performance and to improve service availability.

4 Implementation

This chapter provides an architecture overview and describes the programming interface of the presented proof-of-concept implementation.

4.1 Architecture

A 'typical' WPS implementation supports all features of OGC's WPS specification and the offered processes and algorithms normally run local inside an application-server. There is use of geo-specific third-party libraries or other mainstream GIS, but there is typically no use of Grid Computing or related methods and technologies. The 52° North WPS (52N 2008) is such an implementation. The current release is compliant to OGC's WPS specification version 1.0.0 and former versions of this implementation perform reliable in many research and commercial projects. Schäffer (2004) presents an extension of the WPS specification which allows the chaining of multiple WPS to a single geoprocessing workflow.

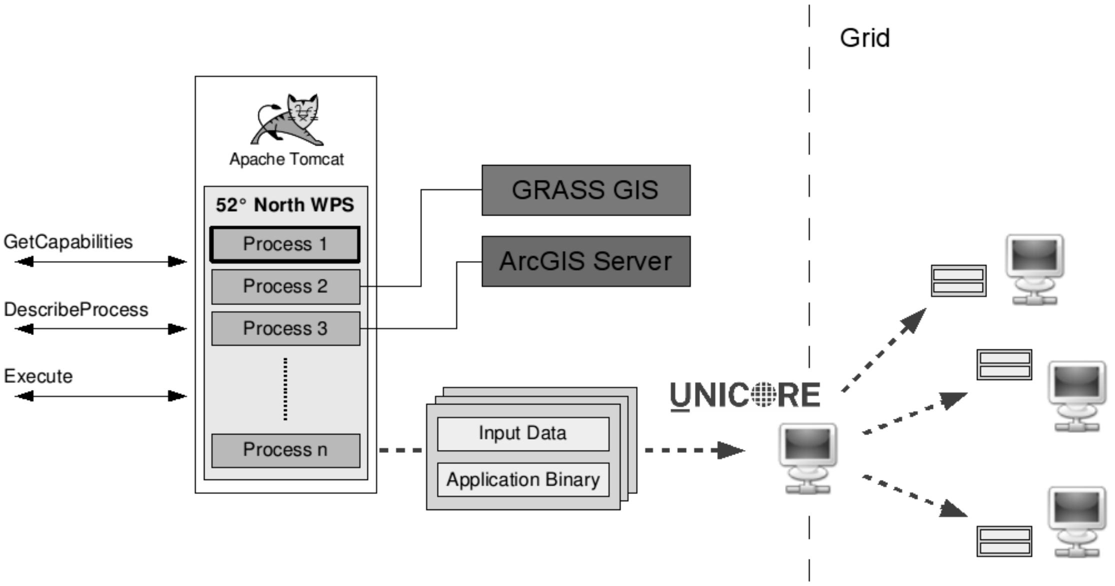


Figure 2: A brief architecture overview of the 52° North WPS-G.

The presented grid-enabled proof-of-concept implementation (the so-called 52° North WPS-G) is based on 52° North WPS and a brief architecture overview of the approach is shown in Figure 2. Inside the 52° North WPS implementation the offered processes and algorithms are managed inside an algorithm repository. Each process can make use of standard Java libraries, other third-party libraries or use expert functionality of other mainstream GIS applications (see Figure 2). When executing distributable processes, the process input data and the process application binaries are copied to the computation nodes and the process application binaries are executed parallel on each computation node. During the calculation phase, the CPU of the WPS server is idle and the WPS is waiting for the calculation results. When all calculation tasks are finished, the WPS fetches the resulting datasets and concatenates them all together.

The current implementation has full support of the UNICORE 6 grid middleware, but the connection to other grid middlewares (e.g. Globus Toolkit) or other Distributed Computing frameworks (e.g. GridGain) could easily be done. There is no need of administrative access to the grid, only a valid certificate for sending jobs is required.

4.2 Development

The 52° North WPS-G is a pluggable framework for offering (distributable) geospatial processes via the the OGC WPS interface and it is completely based upon the Java programming language. The development of distributable processes could easily be done for nearly all geoprocessing problems and consist of the following three steps.

4.2.1 Process Description File

The DescribeProcess operation of a WPS deliver process metadata including unique process identifier, human-readable process description and detailed format information of corresponding input- and output-parameters. On account to this, you have to create a process description file for each implementation of a distributeable process. This file must be full compliant to OGC WPS specification.

4.2.2 Algorithm Implementation

Next you have to implement the process logic itself. Therefor you have to implement the *IGridAlgorithm* interface. The interface consist of three functions. The run-method includes the basic process logic (e.g. a simple intersect calculation). The split-method is very process-specific and divides the whole problem into smaller sub-problems by splitting the input data into smaller chunks. The concatenate-method is very process-specific as well and concatenates the resulting datasets of the distributed and parallel executed sub-problems.

4.2.3 Application Description File

At last you have to specify the libraries needed by your implementation. When calling the Execute operation for the specific process, these libraries are copied to each computation node before executing the run-method remotely.

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions version="1.0.0">
<ProcessDescription wps:processVersion="2"
     statusSupported="true" storeSupported="true">
     <ows:Identifier>
           org.n52.wps.grid.algorithm.SimpleBufferAlgorithm
  </ows:Identifier>
  <ows:Title>
    Simple Buffer
  </ows:Title>
  <ows:Abstract>
```

```
    Create a buffer around a single polygon.
  </ows:Abstract>
  <DataInputs>
    <Input minOccurs="1" maxOccurs="1">
    <ows:Identifier>polygon</ows:Identifier>
    <ows:Title>Polygon to be buffered</ows:Title>
    <ows:Abstract>The Geometries to buffer</ows:Abstract>
    <ComplexData>
    (...)
```

Example 1: The Process Description File is compliant to WPS specification.

```
package org.n52.wps.grid;

public interface IGridAlgorithm
{
     /* run algorithm */
     ProcessOutput run(ProcessInput pInput);

     /* split input data into smaller chunks */
     List<ProcessInput> split(ProcessInput pInput, int pCount);

     /* concatenate the results of distributed processes */
     ProcessOutput concatenate(List<ProcessOutput> pOutput);
}
```

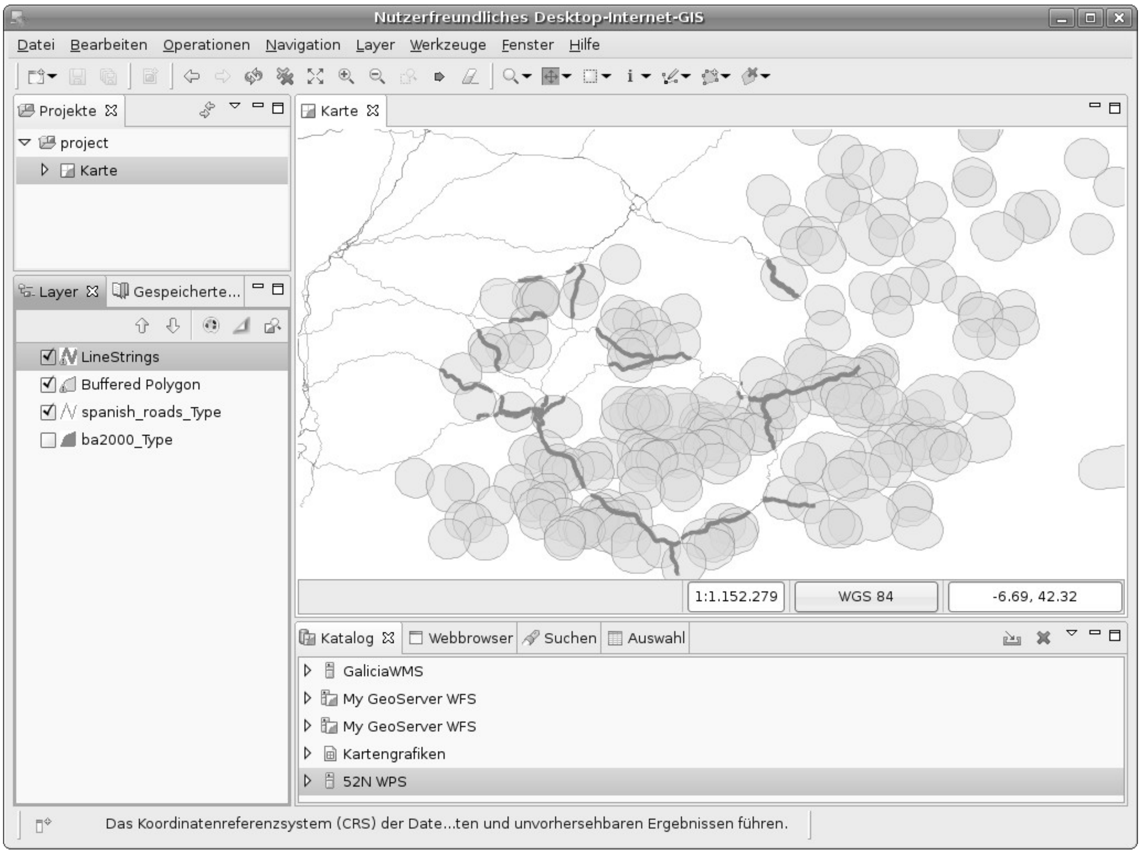
Example 2: Each process must implement the IGridAlgorithm interface.

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
  <app_name>org.n52.wps.grid.algorithm.SimpleBuffer</app_name>
  <version_num>1.0</version_num>
    <file_ref>
      <file_name>52n-wps-server-1.0.0.jar</file_name>
    </file_ref>
    <file_ref>
      <file_name>gmlpacket-2.0-0.4.jar</file_name>
    </file_ref>
    (...)
</application>
```

Example 3: The Application Description File references to all libraries which are needed by the developed distributable process.

4.3 Demo Scenario

The assumed demo scenario from Schäffer and Förster (2008) aims at producing a readable map which indicates the impact of recent fire threats to a transport infrastructure. The scenario involves data about recent fire threat areas and road data. It inherits aspects of a real-time risk-management application, as different data sources have to be integrated and processed in order to improve decision making. The chosen location for this scenario is the North-West of Spain. The data for the burnt areas are provided by the courtesy of the Joint Research Center in Italy and served through a OGC Web Feature Service (WFS). The roads of Spain are taken from CORINE data, provided and served through a WFS from the International Institute for Geo-Information Science and Earth Observation (ICT). In order to produce a satisfying map, some CORINE landcover data, served through an OGC Web Map Service (WMS), is also incorporated.



☑ LineStrings
☑ Buffered Polygon
☑ spanish_roads_Type
□ ba2000_Type

1:1.152.279 | WGS 84 | -6.69, 42.32

GaliciaWMS
My GeoServer WFS
My GeoServer WFS
Kartengrafiken
52N WPS

Das Koordinatenreferenzsystem (CRS) der Date...ten und unvorhersehbaren Ergebnissen führen.

Figure 3: A screenshot of the uDIG client after calculating the impact of recent fire threads on a transport infrastructure.

The calculation, which roads are at stake by a fire threat, is designed as a workflow of several smaller calculation steps. At first, buffers are created around the burnt areas. Secondly, as the roads are too detailed to be displayed on a general overview map, a simplification is applied to the roads and finally the roads are intersected with the burned areas. The first two steps in the workflow refer to a classical WPS which offers the needed processes. The final step in the workflow refers to a grid-enabled WPS which split the calculation problem into smaller sub-problems and distribute the intersection calculation as several parallel jobs over the grid. It is important to note, that this demo scenario is completely based upon distributed web services and that one of these web services do the calculation job in the grid.

A screencast of this demo scenario is online available at homepage of 52° North (see 52N 2008). A hands-on tutorial which explains how to to reproduce the steps from the presented demo scenario (with live access to a grid at Research Centre Jülich) and a cookbook which explains how to develop your own grid-enabled processes in the 52° North Framework will be online available as well.

5 Conclusion and future work

Based on easy and careful extensions of existing 52° North's WPS, the presented proof-of-concept implementation is still a user-friendly framework for offering geospatial processes and algorithms over a network. Compared to other implementations of OGC's WPS specification, the presented approach make use of Grid Computing for parallel processing the data. So the calculation performance and service availability could be enormously improved. To speed up the process calculation time, the calculation tasks are splittet into smaller calculation tasks. Afterwards the sub-problems is distributed in the grid and then processed parallel. In addition to occurring performance and availability advantages, the use of Grid Computing offers the ability to out-source geoprocessing tasks easily to an external (commercial) data processing center. It is important to note, that the presented 52° North WPS-G implementation has full support of OGC WPS 1.0.0 specification and widely adopted UNICORE 6 grid middleware. There is no need of making changes to OGC's WPS specification or to existing OGC WPS compliant applications. There is no need of administrative access to the grid as well. Only a valid certificate for sending jobs is required.

The following list is a short outtake of interesting topics for further researches in context of Grid Computing and spatial processing in GI applications. The last presented topic – the use of Service Level Agreements (SLA) for guaranteeing Quality of Service (QoS) in a Spatial Data Infrastructure (SDI) – is a major topic for the author's further research activities. Other aspects, which are not applicable for geospatial processing and GI applications at first glance, might be of great interest as well.

5.1 High-Level Gridification

Using the grid as a computation resource is only a first step forward into a full grid-enabled SDI. To increase the benefit from Grid Computing, real services in the grid has to be developed (e.g. a stateful WSRF service inside your grid middleware). But there is currently no common approach available, how to gridify OWS and already stay OGC compliant. Another aspect is the moving code and moving data paradigm used by the presented approach. Transferring the whole input data to the computation nodes is generally not applicable in all usecases. The OWS must be embedded very close into grid infrastructure, as well as the input data. A lot of research activities - together with international standardization bodies and with respect to customers and their practical needs - has to be carried out.

5.2 Adaption of mainstream GIS

There exist a lot of mainstream GIS applications which offering professional geoprocessing functionality via proprietary SOAP/WSDL based interfaces and/or Java API, etc. The ESRI ArcGIS Server is one of the major server-based GIS that comes with applications and services for spatial data management, visualization and processing. The GRASS GIS is an open-source based GIS used for geospatial data management, image processing, visualization, etc. To integrate such GIS into an OGC-based SDI, they have to be accessible via a standardized and OGC compliant interface. The WPS specification offers such an interface. Many research to integrate proprietary mainstream GIS into an open architecture had been done at the author's department at University; see Brauner et al. (2008) and their approach of an integration of GRASS GIS functionality into WPS.

To invoke mainstream GIS applications via a standardized and open interface (e.g. WPS) is one first aspect for further research activities. The gridification of those applications – how to access them via the WPS interface and how to process them in the grid - is another important aspect which has to be realized. It is important to note, that the moving code paradigm used by the presented approach - transferring application binaries or source code to the computation node - is not suitable for all kind of GIS. Sometimes the application configuration (e.g. startup a web-service or setup a database) is too complex to make it remotely. A lot of general work on the question, how to bring existing application easily into the grid, has to be carried out. Virtualization might be a possible solution for this kind of problems.

5.3 Desktop Grid Computing

A lot of computing resources are idle on current desktop computers. The increasing appearance of multi-core CPU's enforces such a situation. The motivation behind Distributed Computing projects like BOINC (which is used by SETI@home) is to use idle CPU time on desktop computers, to do e.g. scientific computing or other useful jobs. On the one hand, the advantages of such an approach are evident. At first there is an existing hardware infrastructure and the administration needs for extending this infrastructure with grid capabilities are low. Secondly a broader community is addressed for sharing their ideas and for sharing their computation resources. On the other hand, such techniques are not applicable in time-critical situations when there are heavy requirements on performance and availability of the underlying service (e.g. in emergency situations).

5.4 Service Level Agreements

As the availability of OWS rapidly grow for the last decade, they'll play a major role in emerging e-commerce models. To make OGC based services and OGC-based SDI ready for commercial usage, the Geospatial Digital Rights Management Reference Model (GeoDRM RM; see OGC 2007a) aims at establishing a trusted infrastructure for purchasing, managing and protecting rights to digital content. But until now, there is still a lack of specifications and techniques which intend to guarantee a certain Quality of Service (QoS) in an OGC based infrastructure.

However, monitoring the performance of services and the ability to react quickly on quality fluctuations is an essential skill for service customers and service providers as well. With respect to these needs, normally a formal contract - the Service Level Agreement (SLA) - between a service customer and a service provider for guaranteeing quantifiable service qualities on defined levels will be closed. This agreement contains information about the service level objectives (e.g. limits of maximum and average service response times) and guarantee terms (e.g. rights of service customer if the service provider fails to match defined service qualities over specified time periods). A service consumer will most probably only agree to pay charge for service usage, if the service e.g. is available for a certain percentage of time. In some cases, service consumers will demand the service provider to pay a violation fee, if the agreement terms are violated and such a fee has been stipulated within the agreement. Faced with these scenarios it is evident that the fulfillment of agreements must be monitored by a trustful third-party.

For the reasons explained above, there is a need for establishing a trusted infrastructure for negotiation, managing, monitoring and reporting SLA's in an OGC-based SDI. Grid Computing related methods and standards (e.g. the WS-Agreement Specification; see GGF 2007) can be helpful for ensuring these qualities. When enforcing SLA's for complex geoprocessing workflows, which are composed of several single OWS, the use of active resource reservation is required. Especially in situations when making great demands on OWS, the use of Grid Computing may be a good choice to achieve high QoS's.

6 References

52N (2008), 52° North Initiative for Geospatial Open Source Software GmbH, Website: http://www.52north.org

Aloisio G. et al. (2005), “A Grid-enabled Web Map Server”, Proceedings of the International Conference on Information Technology: Coding and Computing, (ITCC’05)

Brauner J., Schäffer B. (2008), „Integration von GRASS Funktionalitäten in web-basierten GDI Diensteketten“, Presentation, FOSSGIS 2008

Buyya R. (2005), "Grid Computing: Making the Global Cyberinfrastructure for eScience a Reality". CSI Communications 29 (1). Mumbai, India: Computer Society of India (CSI). ISSN 0970-647X.

CERN (31th March, 2008), http://gridcafe.web.cern.ch/gridcafe/

Chen A. et al. (2005), “Grid Computing Enabled Geospatial Catalogue Web Service”, ASPRS 2005 Annual Conference

Di L. et al. (2005), “The Development of a Geospatial Grid by Integrating OGC Web Services with Globus-based Grid Technology ”, The 15th Global Grid Forum – GGF15 (Community Activity)

Foster I., Kesselman C. (1998), „The Grid: Blueprint for a New Computing Infrastructure.”, Morgan Kaufmann Publishers

Foster I. (2000), „What is the Grid? A Three Point Checklist“

GDIGRID (2008), Geodateninfrastruktur-Grid, Website: http://www.gdi-grid.de

GGF (2007), „Web Services Agreement Specification (WS-Agreement)“, Open Grid Forum, OGF Grid Final Documents (GFDs), GFD.107

Kiehle C., Greve K., and Heier C. (2006). “Standardized Geoprocessing – Taking Spatial Data Infrastructures one Step Further”. Proceedings AGILE 2006, 273-282.

Krüger A. (2008), „Web Processing Service (WPS): grid-enabling spatial processing capabilities“, OGC-OGF Collaboration Workshop, The 22nd Open Grid Forum - OGF22

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

OGC (2007b), Geospatial Digital Rights Management Reference Model (GeoDRM RM), OpenGIS reference model: OGC 06-004r3

Schäffer B. (2007), „Integrated Web Geoprocessing Workflow Composition and Deployment “, Diploma Thesis, University of Münster

Schäffer B., Foerster T. (2008). „A Client for Distributed Geo-Processing and Workflow Design“