ISSN 1430-3663

gis.SCIENCE

DIE ZEITSCHRIFT FÜR GEOINFORMATIK

www.gis-biz.de 3/2009

GRID COMPUTING AND GIS | E-INFRASTRUCTURES | WEB PROCESSING | AGILE 2009 WORKSHOP

EDITORIAL

SDI und // and GRID

Werte Leserinnen und Leser,

mit dem aktuellen Heft legen wir einen thematischen Fokus auf die Kopplung von GDI- mit Grid-Technologien. Geoinformationen, weltweit verteilt und von gigantischem Datenumfang, verlangen bei der Prozessierung nach neuen Konzepten, wie z.B. dem Grid-Computing, bei dem hohe Rechenleistung durch ein Cluster lose gekoppelter verteilter Computer bereitgestellt wird.

Schwerpunktmäßig speist sich dieses Heft aus Beiträgen, die zur AGILE-Konferenz in Hannover Anfang Juni 2009 auf einem Workshop zu GDI und Grid präsentiert wurden. Die ausgewählten Artikel geben einen hervorragenden Überblick über dieses aktuelle und interessante Thema. Motiviert wird der Themenschwerpunkt durch einen einleitenden Übersichtsartikel von Christian Kiehle und Patrick Mauè zu Grid-Technologien sowie von Craig A. Lee vom Open Grid Forum und George Percivall vom Open Geospatial Consortium, die auf die zahlreichen Initiativen zu GDI und Grid weltweit hinweisen. Werder/Krüger identifizieren sechs wesentliche Forschungsfragen im Zusammenhang mit der Parallelisierung der Verarbeitung räumlicher Daten. Sowohl Padberg/Greve als auch Woolf/Sharon untersuchen den Web Processing Service (WPS), wie dieser im Grid verteilt und die Gridverarbeitung gekapselt werden kann. Kurzbach et.al. nutzen Grid-Technologien zur Hochwassermodellierung mittels serviceorientierter Geodateninfrastrukturen. Foerster et.al. beleuchten wie das kartographische Generalisierungsproblem durch Grid-Computing ergänzt werden kann. Ein Überblick über die 12. AGILE-Konferenz 2009 von Schiewe sowie den Workshop GDI und Grid durch Kiehle/Mauè rundet den Themenkomplex ab.

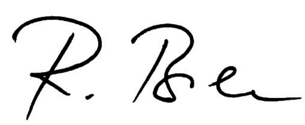
Wir, das Editorial Board der GIS, wünschen Ihnen interessante Einblicke mit diesem Heft.

// Dear Readers,

in this issue we have a thematic focus on the coupling of SDI and Grid technologies. The processing of geoinformation, distributed worldwide and with a gigantic data volume, requires new concepts, such as Grid-Computing, by which the high processing capacity is made available through a cluster of loosely-coupled distributed computers.

The papers covering the main theme of this issue are extended versions of papers presented during a workshop on SDI and the Grid at the AGILE conference in Hanover in early June 2009. The articles selected give an excellent coverage of this current and interesting topic. The motivation for the theme is presented by an introductory overview on Grid technologies given by Christian Kiehle and Patrick Mauè. Craig A. Lee from the Open Grid Forum and George Percivall from the Open Geospatial Consortium refer to the many worldwide initiatives towards SDI on the Grid. Werder/Krüger identify six pertinent research questions with regard to the parallelisation of spatial data processing. Padberg/Greve and Woolf/Sharon investigate the Web Processing Service (WPS), how this may be distributed in the Grid and how the Grid processing may be encapsulated. Kurzbach et al. use Grid technologies for flood modelling using service-oriented spatial data infrastructures. Foerster et al. illustrate how the problem of cartographic generalisation may be assisted using the Grid. A summary of the 12th AGILE Conference 2009 from Schiewe and of the workshop on SDI and the Grid from Kiehle round off the central theme.

We, the editorial board of GIS.Science, hope that you find this issue interesting.



Für das Editorial Board
// For the editorial board – Ralf Bill, Rostock
GIS.SCIENCE | DIE ZEITSCHRIFT FÜR GEOINFORMATIK

Foto: posterdeluxe Fotolia

gis.SCIENCE
DIE ZEITSCHRIFT FÜR GEOINFORMATIK

GRID COMPUTING AND GIS
// GRID COMPUTING UND GIS

EDITORIAL.

SCIENTIFIC PAPERS. // WISSENSCHAFTLICHE BEITRÄGE.

RUBRICS. // RUBRIKEN.

lat/lon GmbH, University of Muenster, Institute for Geoinformatics

GRID TECHNOLOGIES FOR GEOSPATIAL APPLICATION AN OVERVIEW

Patrick Maué, Dr. Christian Kiehle

Abstract: This special issue is all about Grid computing, the idea of splitting up complex and resource-intensive tasks into smaller chunks and let many processors perform the desired computations in parallel. In the following two pages we briefly try to give a more sophisticated definition of Grids. We introduce initiatives which drive the development of Grid technologies and standards, and discuss the current state of Grid computing and its use for Geospatial Technologies.

Keywords: Grid Computing, Spatial Data Infrastructure, Globus Toolkit, Webservices

// GRID-TECHNOLOGIE FÜR RAUMBEZOGENE ANWENDUNGEN - EIN ÜBERBLICK

// **Zusammenfassung:** Die vorliegende Ausgabe ist dem Themenbereich Grid-Computing gewidmet. Grid-Computing liegt die Idee zu Grunde, aufwändige Berechnungen in kleinere Teile aufzubrechen und diese von verschiedenen Rechnern bearbeiten zu lassen. Auf den folgenden Seiten wird in aller Kürze in diesen für die Geoinformatik neuen Bereich eingeführt. Es werden einige Projekte hervorgehoben, die im Spannungsfeld zwischen Grid und Geodateninfrastruktur agieren.

Schlüsselwörter: Grid Computing, Geodateninfrastruktur, Globus Toolkit, Webservices

Authors

Dr. Christian Kiehle
lat/lon GmbH
Aennchenstrasse 19
D-53177 Bonn
E: kiehle@lat-lon.de

Dipl.-Geoinf. Patrick Maué
University of Muenster
Institute for Geoinformatics
Weseler Strasse 253
D-48151 Muenster
E: pajoma@uni-muenster.de

The image of an computing infrastructure available anytime from anywhere reminded Ian Foster and his colleagues of power grids and resulted consequently in the term Grid Computing (Foster et al., 1998). They didn't claim to be the first thinking of the various underlying requirements. In fact, the idea of supplying computing resources without any "inflexible dependence on predetermined programs" (Licklider, 1960), has existed for many decades now. A simple – and one of the best – introduction is given in the GridCafé (Chevalier et al., 2009), a service originally coming from CERN: *Grid computing is a service for sharing computer power and data storage capacity over the Internet.*

Simulating global climate change has to consider various data coming from different observations. The earth atmosphere has to be modeled, but also its biosphere, hydrosphere or the human impact on the land use. This eventually results in enormous amounts of data, and unfortunately also quite complex workflows to correctly simulate the dependencies between these different factors. Here, the need for shared computer power and data storage capacity is self-explaining. Of course, existing supercomputers can also provide these resources, and today simulations used to compute weather forecasts are usually coming from such powerful computers. Why then going through the hassle of dividing processes and data – and anticipating and catching eventual side effects which can destroy the results – if everything can be computed on one computing unit? Obviously not everyone has access to these powers, and only very few have the required resources. Grid computing has to be understood as a service from the Internet, available to everyone who is authorized to use the existing resources. Depending on the complexity of the task, this service can (ideally endlessly) scale up by adding more nodes to the current process, making it eventually much more powerful then supercomputers. The fastest Grid-based virtual computer (BOINC) has currently a processing power of around 2.3 Peta Flops, whereas the fastest supercomputer (Roadrunner) can merely perform 12.8 Giga Flops.

The idea of Grid as a service implies also an easy access to its storage and processing capabilities. A Grid middleware like the Globus Toolkit or Unicore allows for submitting tasks to the Grid. All subsequent tasks (distributing and monitoring jobs and collect the results in the end) are managed by the various components within the Grid and remain largely hidden away from the user. Two aspects are very important on this level: security and standards. Grid users are grouped in so called "Virtual Organizations"(Foster et al., 2001), which help to identify who is authorized to access which resources in the Grid. Trust is one crucial aspect for the Grid's acceptance. After all, the user is asked to upload his precious data and algorithms into an unknown network hidden behind a single interface provided by the middleware. Implementing open standards like the Web Services Resource Framework (WSRF, Czajkowski et al., 2004) or the Open Grid Services Architecture (OGSA, Foster et al., 2005) ensures the interoperability between clients and the Grid's exposed interfaces. And using open standards together with well-established middlewares with a good reputation in quality and security builds up the so much needed trust.

The Grid's flexibility, security, and nearly free availability sounds promising especially for science, where researchers are constantly coming up with new detailed models. And also commercial applications recognized the benefits, and Grid computing – here tightly integrated, extended, and re-branded to Cloud computing (Buyya et al., 2009)– is getting popular especially among small companies.

The more details we are able to simulate, for example in the mentioned climate models, the more precise and trust-worthy are the derived conclusions. Research is in need for Grid computing, and so are many other users coming from the public or private sectors. A Grid has to be considered as an infrastructure which can and wants to be used by anyone who is in need of its capabilities. Like the power grid, its availability has to be assured by well-established authorities, preferably coming from the public sector. Infrastructures are nation-wide projects, and consequently we see a large number of different Grid initiatives around the world (Gentzsch et al., 2007). Today, nearly every country has its own network, and current developments (like the European Grid Initiative EGI) are actually focusing in linking existing infrastructures to build a Grid spanning across the various nations.

The question we raised and want to find answers for here is how Grids are able to support GIScience, the science of understanding, processing and communicating geographic information. From a European perspective is has been obvious to discuss the ideas of Grid-/GIScience integration within the largest European conference in the area of GIScience: the 12th Association of Geographic Information Laboratories in Europe (AGILE). Within a pre-conference workshop we discussed ideas on "Grid Technologies for Geospatial Applications" (see Grid Technologies for Geospatial Applications article by Maué & Kiehle in this issue). The current special issue summarizes the European state-of-the-art concerning Grid-Technologies within the GIScience domain.

Good starting points for studying the interconnection between Grid and GIScience are some of the recent research-and development projects in this area:

- The British SEE/SAW-GEO project (http://edina.ac.uk/projects/seesaw/) focuses the integration of security mechanisms into spatial data infrastructures by utilizing Grid technologies. SEE-GEO is an acronym for Secure Access to Geospatial Services). SAW-GEO on the other hand focuses workflows within spatial information systems and is the short form form Development of Semantically-Aware Workflow Engines for Geospatial Web Service Orchestration.
- The European project Cyber-Infrastructure for Civil protection Operative Procedures (CYCLOPS, www.cyclops-project.eu) utilizes Grid technologies for civil protection and also operates on spatial data sets.
- The German GDI-Grid (www.gdi-grid.de) integrates OGC and OGF standards on the German eScience infrastructure D-Grid (www.dgrid.de). Based on three selected use cases (flood simulation, noise propagation and emergency routing) a spatially-enabled Grid middleware will be developed. Also running on D-Grid is the

Collaborative Climate Community Data and Processing Grid (C3Grid, www.c3grid.de). C3Grid develops a grid-based research platform for earth systems research.
- The Global Earth Observation Grid (GEO-Grid) is a world-wide initiative for the Earth Science community. It aims to integrate a great variety of Earth Science Data (satellite imagery, geological data, etc.) through virtual organizations while keeping use restrictions on classified data sets. GEO-Grid maintains an extensive website: www.geogrid.org.

As you will encounter within the articles presented in this special issue, different underlying assumptions regarding security or interfaces lead to partly conflicting standards and solutions. The GIScience community (well-organized under the umbrella of the Open Geospatial Consortium OGC) and the Grid community (represented by the Open Grid Forum OGF) are currently working closely together to overcome these obstacles. The joint article of Craig Lee (President of the Open Grid Forum) and George Percivall (Chief Architect of the Open Geospatial Consortium) is a good example of the close cooperation between the both relevant standardization organizations.

Standardization will provide a solid ground for application developers and service providers. Interoperability between Grid computing environments and spatial applications is the first step towards an even broader acceptance of Grid computing within the spatial sciences and beyond. ◀

References

Baranski, B. (2008): Grid Computing Enabled Web Processing Service. Proceedings of the 6th Geographic Information Days, pp. 243–256.

Buyya, R. ; Yeo, C. S. ; Venugopal, S. ; Broberg, J. ; Brandic, I. (2009): Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. In: Future Generation Computer Systems vol. 25, Nr. 6, pp. 599–616.

Lee, C.; Percivall, G. (2008): Standards-Based Computing Capabilities for Distributed Geospatial Applications. In IEEE Computer vol. 41, Nr. 11, pp. 50–57.

Chevalier, C. ; Burne, C. ; Grey, F. ; Mondardini, R. : Grid Café.

Czajkowski, K. ; Ferguson, D. F. ; Foster, I. ; Frey, J. ; Graham, S. ; Sedukhin, I. ; Snelling, D. ; Tuecke, S. ; Vambenepe, W. ; et al.: The WS-Resource Framework : Global Grid Forum (GGF), 2004.

Foster, I. ; Foster, I. ; Kesselman, C. : The Grid: Blueprint for a New Computing Infrastructure : Morgan Kaufmann, 1999.

Foster, I. ; Kesselman, C. ; Tuecke, S. (2001): The Anatomy of the Grid – Enabling Scalable Virtual Organizations. In: International Journal of Supercomputer Applications vol. 15, Nr. 3.

Foster, I. ; Kishimoto, H. ; Savva, A. ; Berry, D. ; Djaoui, A. ; Grimshaw, A. ; Horn, B. ; Maciel, F. ; Siebenlist, F. ; et al. (2005): The Open Grid Services Architecture, Version 1.0 : Global Grid Forum (GGF).

Gentzsch, W. ; McGinnis, L. F. ; Wallom, D. C. : Grid Initiatives: Lessons Learned and Recommendations. Research And Industry, 1–31.

Licklider, J. C. (1960): Man-Computer Symbiosis. In: IRE Transactions on Human Factors in Electronics vol. HFE-1, Nr. 1, pp. 4–11.

Yamamoto, N.; Nakamura, R.; Yamamoto, H.; Tsuchida, S.; Kojima, I.; Tanaka, Y.; Sekiguchi, S. (2006): GEO Grid: Grid infrastructure for Integration of Huge Satellite Imagery and Geoscience data sets. The 6th IEEE International Conference on Computer and Information Technology, CIT '06. p 75.

GiN schreibt Geoinformatik-Nachwuchsförderpreis für das Jahr 2009 aus – Pressemitteilung 06/2009

Der GiN e.V. schreibt für das Jahr 2009 einen Förderpreis für junge Nachwuchswissenschaftler in Norddeutschland aus. Mit dem Preis werden herausragende **Diplom- und Masterarbeiten** gewürdigt, finanziell honoriert und einer größeren Öffentlichkeit bekannt gemacht. Der Förderpreis besteht aus drei Stufen, wobei jede Stufe mit einem Preisgeld und einer Urkunde ausgezeichnet wird. Über die Vergabe des Preises entscheidet eine unabhängige Jury. Die Preisgelder betragen je Platzierung 1000 Euro, 500 Euro sowie 250 Euro. Die Gewinner werden am 17. März. 2010 auf der GEOINFORMATIK 2010 in Kiel geehrt. Der Bewerbungsschluss ist der 01. Januar 2010. Alle interessierten Studenten aus Norddeutschland sind herzlich eingeladen, sich zu bewerben. Die Ausschreibungsunterlagen stehen im Internet unter http://www.gin-online.de/downloads/nachwuchspreis_2009/gin_nachwuchspreis_ausschreibung_2009.pdf zum Download bereit oder können über die Geschäftsstelle angefordert werden.

Nähere Informationen: Geschäftsstelle GiN e.V., E-Mail: info@gin-online.de, URL: www.gin-online.de

Open Grid Forum, The Aerospace Corporation, Open Geospatial Consortium

THE EVOLUTION OF GEOSPATIAL E-INFRASTRUCTURES

Dr. Craig A. Lee, George Percivall

Abstract: Linked to the constantly changing computer infrastructures and the analysis of spatial datasets, there currently exist networks which are capable of processing all types of information online across administrative domains and regulations. The target of such networks (also known as Grids or e-Infrastructures) is to combine complex datasets, which may come from all parts of the world, to process them using distributed computing and to allow an interdisciplinary interpretation of the results by scientists, engineers and administrations. The potential application areas for such spatial data networks ranges from e.g. agriculture, energy supply and medical care to themes such as disaster management.

Starting withfew examples for established national and international networks for the analysis of spatial information, this article presents work on the worldwide development and integration of geodata by easing the combination of the initiatives of the Open Grid Forum (OGF) and the Open Geospatial Consortium (OGC). With this, the need for a standardisation of web-services, transfer protocols and description languages will be shown. Finally, based on the example of Hurricane Katrina (2005) which caused immense damage, it is shown which steps will be necessary in the future in order to better predict such catastrophic weather events.

Keywords: Grid Computing, e-infrastructures, Open Geospatial Consortium (OGC), Open Grid Forum (OGF), OGC Web Service Standards

// DIE ENTWICKLUNG DER GEOSPATIAL E-INFRASTRUKTUR

// Zusammenfassung: Verbunden mit der sich stetig verändernden Computerinfrastruktur im Zusammenspiel mit der Auswertung räumlicher Datenbestände existieren heute weltweit Netzwerke, welche alle Arten von Informationen online über eine Vielzahl von administrativen Domains und Sicherheitsrichtlinien hinweg verarbeiten können. Der Hintergrund solcher Netzwerke (auch Grids bzw. e-Infrastructures genannt) besteht dabei darin komplexe Datenbestände, welche von verschiedenen Orten auf der Welt kommen können, miteinander zu vereinen bzw. weltweit verteilt zu berechnen, um die Ergebnisse dann von Wissenschaftlern, Ingenieuren sowie Regierungsstellen interdisziplinär auswerten zu können. Der potentielle Anwendungsbereich für derartige räumliche Datennetze erstreckt sich dabei z.B. auf die Landwirtschaft, die Energieversorgung, die medizinische Versorgung, aber auch auf Themen wie das Katastrophenmanagement.

Nach wenigen Beispielen für nationale bzw. international etablierte Netzwerke zur Auswertung räumlicher Informationen wird im Artikel auf Bestrebungen eingegangen, die weltweite Entwicklung und Vernetzung von Geodaten durch eine Verknüpfung der Initiativen der beiden Organisationen Open Grid Forum (OGF) und Open Geospatial Consortium (OGC) miteinander zu erleichtern. Damit verbunden wird auch die Notwendigkeit einer Standardisierung von Webdiensten, Übertragungsprotokollen und Beschreibungssprachen angesprochen. Abschließend wird am Beispiel des Orkans Katrina (2005), welcher immense Schäden verursacht hat, gezeigt, welche Schritte zukünftig notwendig sein werden, um derartige katastrophale Wetterereignisse verbessert vorherzusagen.

Schlüsselwörter: Grid Computing, e-infrastructures, Open Geospatial Consortium (OGC), Open Grid Forum (OGF), OGC Web Service Standards

Authors

Dr. Craig A. Lee
The Aerospace Corp. M1–102
2350 East El Segundo Blvd.
El Segundo, CA 90245 USA
E: craig.a.lee@aero.org

George Percivall
Open Geospatial Consortium
483B Carlisle Drive
Herndon, VA 20170 USA
E: gpercivall@opengeospatial.org

Computing infrastructures are constantly changing, and this includes those used for geospatial information systems. From mainframes, to minicomputers, to a connected world of networked machines and web-enabled devices, society has come to expect all manner of information and resources to be seamlessly online. While tremendous progress has been made with capabilities such as social computing networks that enable complex interactions, we are only beginning to see the development of distributed computing platforms that are flexible enough to be dynamically provisioned across multiple administrative domains, with strong security models – and self-sustaining business models. These platforms have been called webs, grids, service-oriented architectures – and more recently e-infrastructures. Application domains that will drive the development of such e-infrastructures are those that are inherently distributed, where data may come from multiple sources, and be processed and consumed in different locations.

Such is the case for many geospatial applications. Geospatial data has immense applicability across fields such as environmental monitoring, disaster management, civic planning, energy management, e-health, agriculture, and many more. It is clear that data for these application domains may come from many sources in the field, and from different archives maintained by different organizations. Likewise, the users in such geospatial domains could be distributed across academia, industry, governments, or other virtual organizations.

To cite just a few examples, the GeoDatenInfrastruktur Grid (GDI-Grid), is hosted on the German National D-Grid and supports applications such as flood simulation, disaster routing, and noise propagation. The Global Earth Observation Grid (GEOGrid), in Japan virtually integrates satellite imagery, geological data, and ground-sensed data for environmental monitoring and disaster management, such as evaluating landslides caused by earthquakes. Likewise, the Debris Flow Monitoring project (DebrisFlow) in Taiwan integrates data from multiple sensors in the field to predict and alert authorities to dangerous debris flows caused by torrential rains in mountainous areas that can threaten population centers at lower altitudes. The EU CYCLOPS project (CYCLOPS) to build a cyber-infrastructure for civil protection operative procedures demonstrated the integration of data from both Italian and French data archives to assess wild fire risk in border areas. Many other examples are possible.

Facilitating the development of both the geospatial applications and the supporting e-infrastructures has been the goal of the collaboration between the Open Geospatial Consortium (OGC) and the Open Grid Forum (OGF). While OGC represents many organizations with geospatial requirements, OGF represents many organizations with distributed computing requirements. Since both organizations endeavor to coordinate the specification of standards, a close collaboration with a formal memorandum of understanding was clearly needed to coordinate the development of geospatial e-infrastructures. This collaboration was recently reviewed in (Lee, Percivall 2008).

This formal collaboration resulted in OGF participating in the latest OGC Web Services testbed (OWS6). Here OGF standards, such as the Job Submission Description Language (JSDL) and the HPC Basic Profile (HPCBP), were integrated with different implementations of the OGC Web Processing Service (WPS). These were used in different tasks identified by major stakeholders to drive the integration and demonstration of end-to-end, distributed geospatial capabilities. One such scenario was an airport disaster where federal and local authorities could securely share information, and weather prediction and plume dispersion codes could be run dynamically to inform first responders in the field about potential toxic cloud hazards in affected buildings. An OGC-OGF Collaboration Group has also been set-up with a wiki and mailing list to facilitate interaction (OGCOGF). This will support groups such as the g-Lite OWS Working Group (G-OWS, (GOWS)). The goal of G-OWS is to implement many of the basic OGC Web Service standards for geospatial tools on g-Lite, the EGEE software stack. EGEE (EGEE) is Europe's flagship grid infrastructure and is generally recognized as the world's largest and most successful. Such a geospatial service layer will be essential to implement capabilities on national and regional infrastructures, as required by the European INSPIRE (Infrastructure for Spatial Information in Europe) initiative (INSPIRE). Finally we also note that both the European Geosciences Union (EGU) and the American Geophysical Union (AGU) have internal organizations for Earth and Space Science Informatics (ESSI) that are also promoting the development of geospatial e-infrastructures.

Concomitantly with the adoption of distributed computing platforms within the geospatial community, there is also a rapid evolution of infrastructures occurring within the distributed computing community. The concept of on-demand provisioning of abstracted or virtualized resources, i.e., cloud computing, is gaining enormous popularity. This has enabled clear business models whereby commercial operators of data centers, such as Amazon, Google, and Microsoft, can offer computing and storage services for a fee. These business models are also facilitated by the enormous economies of scale that large data centers can achieve. By using such dynamically provisioned resources, businesses can meet unexpected surge requirements without having to overprovision in-house resources, or they can outsource their entire IT requirements altogether. Clearly there are significant security and reliability issues in doing so, but the economics of the situation will attract many suitable applications. The development of further security and reliability mechanisms will only increase the scope of the public cloud marketplace.

On-demand resources, however, have great appeal even beyond the domains of e-commerce and IT as a service. The ability to acquire and release resources on-demand appeals to scientific and engineering users for many of the same reasons of flexibility and economy. Hence, there is strong interest in developing capabilities for HPC in the Cloud. EGEE has announced a collaboration with the RESERVOIR project to integrate some type of on-demand capabilities for their user base. Presumably such a capability would carry-over into the European Grid Infrastructure (EGI) that will be an interoperable association of numerous national grid infrastructures. Likewise in the USA, the NSF is planning a transition from the current TeraGrid projects to several eXtreme Digital (XD) projects that will be the cyberinfrastructure of the future. OGF is conducting a year-long effort to assist the XD teams in identifying key cyberinfrastructure requirements to be addressed, and support for on-demand resources is already on the short list. Specifically with regard to geospatial applications, a workshop on "Geospatial SOA and

Cloud Computing" was recently conducted by the US Federal Geographic Data Committee (FGDC) to identify the state-of-play of service architectures and cloud computing for geospatial applications. The workshop reviewed specific experiences in the use of SOA techniques and emerging cloud computing in the context of integrating online government Web services for geospatial applications. Topics included background on SOA in the federal context, plans for a Federal Service Oriented Infrastructure "cloud," and presentations by on best practices and lessons learned from designing and deploying Web services.

Such efforts are actually happening in a much larger context that will drive cloud – and geospatial – standards: the US, UK, and Japanese governments have all announced national cloud initiatives. The US initiative plans to open a Cloud Storefront with the ultimate goal of reducing the federal IT budget by hosting many governmental IT functions. The UK initiative is part of the Digital Britain plan and aims to rationalize across national computing, telecommunications, and data center requirements. The Japanese Kasumigaseki Cloud also intends to improve efficiency and exploit economies of scale across government agencies. These government clouds will definitely host a significant number of geospatial applications where portability and interoperability are mandated, at both the infrastructure and application levels.

Such national cloud initiatives will only add to the drive for green IT as it applies to data centers and clouds. Controlling energy consumption by enforcing energy policy across a distributed cloud infrastructure will not only reduce costs, but also determine where data centers can be located with respect to energy sources. We note that both the US and EU have efforts for managing energy policy. The US Environmental Protection Agency has a data collection initiative to develop an Energy Star rating for data centers. Likewise the EU Data Centre Code of Conduct will collect data to determine "best of breed" and feasible targets for data center efficiency. Such efforts will drive the need for standard sensor networks to monitor data centers and inform the workload management tools that can enforce energy policy. In many ways, green data centers could be considered an instance of green buildings and could leverage common, standard tooling.

All of the areas of activity cited in this short article will drive the co-development of geospatial grid and cloud standards. The goal of the OGC and OGF collaboration is to coordinate and promote this development. Likewise, in the cloud arena, OGF is a founding member of cloud-standards.org, a coordination venue for organizations working on cloud standards. Here, organizations such as OGF, the Distributed Management Task Force, the Storage Networking Industry Association, the Cloud Security Alliance and others are sharing information on related cloud standards work. As a case in point, the OGF Open Cloud Computing Interface (OCCI) could very well dovetail with the DMTF's Open VM Format (OVF) to enable portability and interoperable among public and private clouds.

In conclusion, we present an extraordinary case study that argues strongly for effective geospatial distributed computing. In 2005, Hurricane Katrina took over 1800 lives and caused over $81b in property damage. The available hurricane prediction models at the time were woefully unable to accurately predict the trajectory or severity of the storm (Bogden 2007). Without knowing where the hurricane was going, it was impossible to use precipitation models or flooding models to determine which geographical areas might be at risk. To effectively predict such events and mitigate such disasters, advancements are required in these areas:

1 fundamental atmospheric and oceanic science,
2 computational science to create the necessary models and codes,
3 operational infrastructures to enable the necessary resources to be allocated on-demand to produce results in time, and
4 easy-to-use geospatial information systems that can be used by decision makers and first responders in the field to mitigate risk and damage.

This is, in a sense, a geospatial computation "moon shot" that should inspire a generation of scientists, engineers, and political leaders for the common good. In much the same way that space exploration spurred the development of technology that greatly affected many other areas of industry, commerce, and society, developing the geospatial capabilities necessary to achieve hurricane prediction will enable many other areas of technical and societal benefit in ways that we cannot foresee at this time. ◀

References:

(GDI-Grid) http://www.gdi-grid.de

(GEOGrid) http://www.geogrid.org

(DebrisFlow) https://portal.opengeospatial.org/files/?artifact_id=34664

(CYCLOPS) http://www.cyclops-project.eu

(OGC) www.opengeospatial.org

(OGF) www.ogf.org

Lee, C., Percivall, G. (2008): Standards-based Computing Capabilities for Distributed Geospatial Applications, IEEE Computer, pp. 50–57, November, 2008.

(OWS6) http://www.opengeospatial.org/projects/initiatives/ows-6

(JSDL) http://www.ogf.org/documents/GFD.136.pdf

(HPCBP) http://www.ogf.org/documents/GFD.114.pdf

(WPS) http://www.ogcnetwork.net/node/392(OGCOGF) https://lists.opengeospatial.org/mailman/listinfo/ogf_ogc_coordination

(GOWS) http://sites.google.com/a/imaa.cnr.it/g-ows/Home

(EGEE) http://www.eu-egee.org

(INSPIRE) http://inspire.jrc.ec.europa.eu

(EGI) http://web.eu-egi.eu

(FGDC) http://www.fgdc.gov/library/workshops/soa-cloud-computing-workshop

(OCCI) http://www.occi-wg.org

(OVF) http://www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf

Bogden, P., et al. (2007): Architecture of a Community Infrastructure for Predicting and Analyzing Coastal Inundation, Marine Technology Society Journal, V41n1, pp. 53–61.

Leibniz Universität Hannover, Institute of Cartography and Geoinformatics, Technische Universität Berlin, Department for Geodesy and Geoinformation Science

PARALLELIZING GEOSPATIAL TASKS IN GRID COMPUTING

Stefan Werder, Andreas Krüger

Abstract: The available number and amount of geospatial data are increasing. So is the complexity of performed calculations on these datasets. This leads to the need for efficient parallelization of geospatial tasks. Another aspect that plays an important role in connection with parallelization is distributed computing.

In this paper six (research) questions are presented, that have to be answered in the context of parallelizing geospatial tasks in a grid computing environment. However, they are not restricted to the grid architecture. The questions include aspects of whether to parallelize the data or the tasks, data quality issues, system architecture decisions, and workflow orchestration.

Keywords: Grid Computing, Parallelization, Data Quality

// PARALLELISIERUNG VON GEOSPATIAL TASKS IM BEREICH GRID COMPUTING

// Zusammenfassung: Die verfügbare Menge und Anzahl von Geodaten nimmt permanent zu. Damit verbunden ist aber auch die Berechnung derartiger Datensätze immer komplexer geworden. Dies führt dazu, dass zukünftig effiziente Wege einer parallelen Datenberechung gefunden werden müssen. Ein weiterer wichtiger Aspekt, welcher in dieser Hinsicht bezogen auf eine Parallelisierung eine große Rolle spielt, ist das verteilte Rechnen.

In diesem Artikel werden sechs Fragen präsentiert, welche im Kontext einer simultanen Berechung von räumlichen Datensätzen in einem Grid beantwortet werden müssen. Damit verbunden werden hier auch Probleme mit der Datenqualität, der Systemarchitektur sowie der Arbeitsorganisation angesprochen.

Schlüsselwörter: Grid Computing, Parallelisierung, Datenqualität

Authors

Dipl.-Ing. Stefan Werder
Leibniz Universität Hannover
Institute of Cartography and Geoinformatics
Appelstraße 9a
D-30167 Hannover
E: stefan.werder@ikg.uni-hannover.de

Dipl.-Inf. Andreas Krüger
Technische Universität Berlin
Department for Geodesy and Geoinformation Science
Straße des 17. Juni 135
D-10623 Berlin
E: krueger@igg.tu-berlin.de

MOTIVATION

The demands that both spatial data infrastructures and GIS have to meet are rapidly increasing. Three main factors can be identified as the sources for these needs. Firstly, the available number and amount of spatial data sets are rapidly increasing. This is due to the availability of more detailed data acquisition techniques, such as airborne laser scanning, detailed models in 3D and also 4D, affordable geo-enabled sensors and such devices as mobile phones equipped with GPS and inertial navigation systems (INS), as well as data acquisition by the masses (crowdsourcing). Secondly, the complexity of applied operations or calculations on these data sets is increasing due to more complex models (e.g. optimization problems in generalization). The third factor is the consideration of security aspects, which are not only used for controlling who can access data to which extent and under which restrictions, but also as a basis for accounting and billing.

Demands increasing and so are the available infrastructures for data storage and computation in terms of performance and flexibility. One indicator for the increasing performance is the biannually released list of the top 500 supercomputers in the world, which shows an exponential growth (Meuer 2008). Another indicator is the dynamic development of the fields of grid computing and cloud computing. Grid computing has been discussed in the scientific community for almost a decade and is well defined by Foster's (2002) three-point checklist. In contrast, cloud computing is relatively new and not strictly defined (Vaquero et al. 2009). However, the two fields share the same vision of making high-performance computing available for a larger user base.

The different approaches for distributed computing have the need for efficient parallelization in common, which can be based on both data and computations. This paper focuses on the topic of parallelization of geospatial tasks in a grid computing environment, which raises two questions. The first one is how to divide a given task into several subtasks, which then can be executed concurrently. In most cases the solution to this problem depends only on the algorithms and processes of the task itself, and is therefore not specific or limited to grid computing. The second question is how to make use of the grid in order to accelerate processing, in which aspects such as available grid middleware and orchestration of web services have to be considered.

The remainder of this paper is structured as follows. In the next section the Spatial Data Infrastructure Grid (GDI-Grid), which is the project the authors are working in, is presented shortly. Thereafter six research questions that have to be answered in the context of parallelization in a grid computing environment are listed and explained in more detail in the consequent sections.

1. GDI GRID

In Germany the D-Grid initiative builds the foundation for a sustainable development of grid technology and eScience methods. The project is funded by the German Federal Ministry of Education and Research (BMBF) with 100 Million Euro and runs from 2005 to 2010. Part of D-Grid is the Spatial Data Infrastructure Grid (GDI-Grid), which aims at the efficient mining and processing of spatial data for simulation of noise dispersion and disaster management. The focus is on processing services, which are able to handle massive amount of geospatial data and also compute complex operations based on three real-world scenarios. The processing services are thereby both traditional Web Processing Services (WPS) defined by the Open Geospatial Consortium (OGC) and grid services according to the Web Service Resource Framework (WSRF) defined by the Open Grid Forum (OGF). One of the mentioned scenarios uses noise propagation simulations for the assessment and management of environmental noise. Based on a directive by the European Union (2002) all EU member states are obliged to inform the public every five years about environmental noise emitted by major transport routes (road, railway, air), and from sites of industrial activity. Therefore noise maps have to be produced, which also serve as a basis for further acoustic planning and may have significant economical impacts.

2. OVERVIEW OF RESEARCH QUESTIONS

Having introduced the relevant background as well as some basic concepts and terminology, we want to present six (research) questions that, from our perspective, have to be answered in the context of parallelization of geospatial tasks in a grid computing environment. The questions as well as the used terminology are more precisely outlined in the following sections.

1 Which existing geospatial tasks (potentially) benefit from parallelization?
2 Which of the two concepts of task or data parallelism yields the best results for a given problem, or is it a combination of both?
3 How does parallelization affect the quality of the task result?
4 Where do the actual parallel tasks run in the overall system architecture – on different cores of a CPU, in a GPU, in a cluster or on different worker nodes in a grid environment? And how are the data distributed in that architecture?
5 How to efficiently orchestrate several parallelized tasks in a complex work flow?
6 How complex and time-intensive is the adoption of existing code to gain advantage from parallelization?

The research questions two and six address the question how to divide a given task into several subtasks, whereas the research questions four and five address the question how to make use of the grid in order to accelerate processing.

2.1 GEOSPATIAL TASKS BENEFITING FROM PARALLELIZATION

The question which existing geospatial tasks benefit from parallelization includes two aspects. Firstly, a list of geospatial operators, tasks and complex processes which benefit from parallelization, e.g. by reducing runtime or increasing stability, should be created. Based on that list it would be easier for a GIS developer to decide which parts of a given program should be rewritten or exchanged by already parallelized code fractions or libraries.

Secondly, it also includes an in-depth performance analysis based on reproducible metrics. These metrics are not specific to geospatial tasks, but are suitable for comparing different approaches for parallelizing the same geospatial task. For analyzing performance three measures can be estimated. Speedup measures the factor of increased speed between parallel and sequential processed calculation. Efficiency measures the utilization of the available processors. Scalability measures the ability to increase performance while increasing the number of processors. For a quantification of speedup, efficiency, and scalability, several estimation metrics can be used. These are for instance Amdahl's Law, Gustafson-Barsis' Law, the Karp-Flatt Metric, or the Isoefficiency Metric (Quinn 2003).

2.2 TASK AND DATA PARALLELISM

Two high-level concepts exist for parallelizing a task (Culler et al. 1999). Firstly, a task can be divided into several independent subtasks operating on the same or different dataset (task parallelism). Secondly, it can be divided into several subtasks that each processes a part of the whole dataset (data parallelism). Data parallelism corresponds to tiling or partitioning of geospatial data. If the individual subtasks require no communication between each other in data parallelism, the problem is also called embarrassingly parallel or nicely parallel (Foster 1995).

For data parallelism explicit processing steps for both tiling and subsequent merging of the dataset are necessary. A generic approach for tiling is sufficient for our applications, and we suppose this applies also to other applications. The parameter set that satisfies our requirements are the width and height of a single tile, the coordinates of one corner of a start tile, the tile border size, and rules for objects located near or on tile borders (fig. 1). The tile border leads to the inclusion of additional data beyond the actual tile size.

For simulations of noise propagation within the GDI-Grid the tile border size is 3km for a tile size of $1km^2$ leading to a total of $49km^2$. This overhead is necessary, because all noise emitters in these areas contribute to the total noise level in the centre tile. Tiling rules specify how objects near or on tile borders are distributed between adjacent tiles, for instance they can be put into the tile where their centre point lies in, cloned for all tiles in which their bounding box is in, or simply being cut at the tile borders.

In contrast to the generic approach for a tiling service, the merging process incorporates application specific logic for most of our applications. The logic has to take the inter-dependencies at the tile borders into account in order to obtain a continuous result.

Parallelization can be achieved using different criteria – besides others by object type (e.g. buildings, streets), by geometry type (e.g. point, line, area), by operation type (e.g. calculate length, calculate angles), by spatial criteria (e.g. tiling), and by Level of Detail. Therein the geometry type is often tied to the object type or subtype. The following example shows a parallelization for checking the integrity of geospatial datasets (fig. 2). In the example buildings have to fulfil the constraints that their border consists of only orthogonal lines and that each building's area is greater than $100m^2$. For a detailed description of the constraints as well as their formalization in the context of the GDI-Grid see Werder (2009). The choice of the optimal parallelization criteria as well as their chaining can have significant influence on the performance. For checking data integrity, for instance the parallelization by object type is performed first, because integrity constraints differ in most cases for different object types.

The following example is concerned with map generalization (Müller 2008). There the concept of data parallelism is used in order to speed up the processing. Maps with land use information, which are traditionally acquired from satellite imagery, are derived from existing datasets of a national mapping agency offering more details as well as higher resolution. The overall task includes several processing steps, such as reclassification, adjustment of geometries, and aggregation. As a result, the runtime of the program is evaluated in relation to several tile sizes (fig. 3).

The runtime reaches the optimum for a tile size with a width and height of approximately 3000 m. For smaller tile sizes the data input and output operations slow down the overall task, while for larger tile sizes the

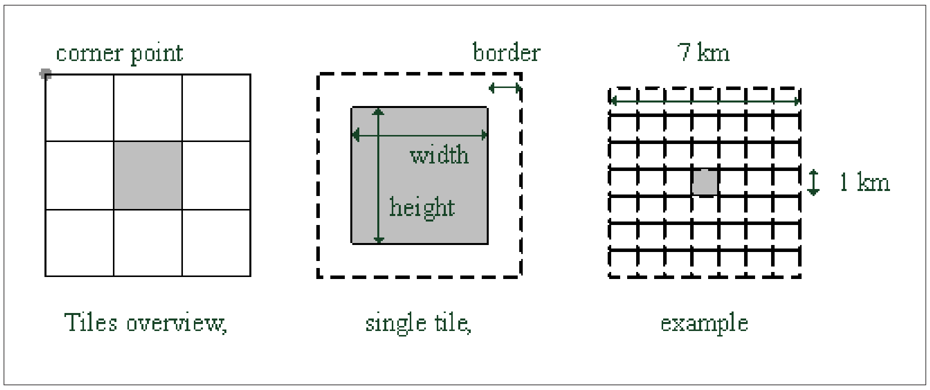


Figure 1: Parameters for a tiling service

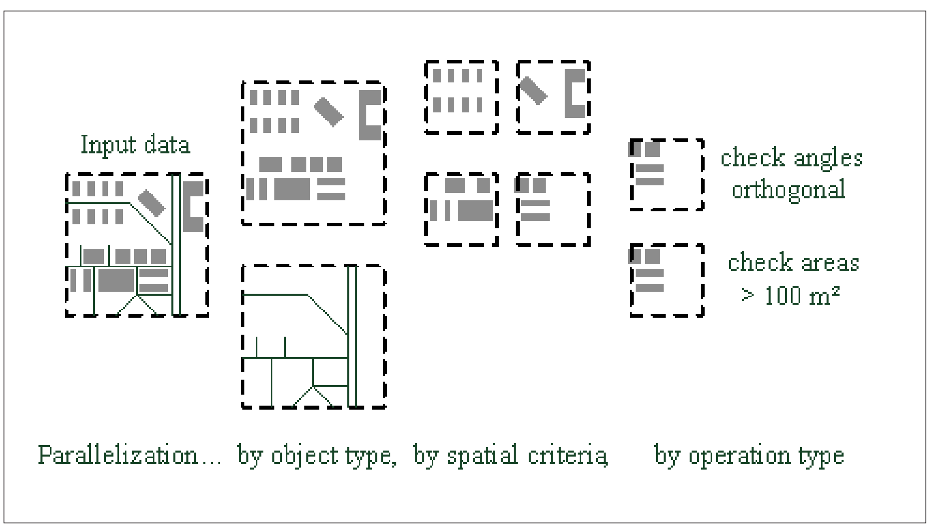


Figure 2: Parallelization of massive geodata for checking data integrity

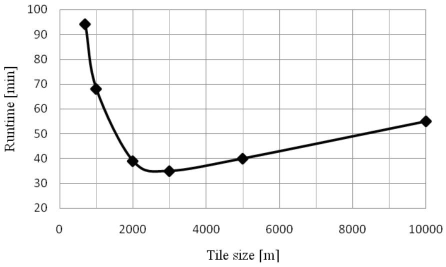


Figure 3: Runtime [min] in relation to tile size [m] (based on Müller (2008))

algorithm needs to compare more objects with each other. This leads to the conclusion that general recommendations concerning individual parallelization aspects, such as optimum tile size, are not reasonable, because they depend on the underlying algorithms, data, and system architecture.

2.3 PARALLELIZATION AND QUALITY

Parallelization can affect the quality of obtained results in a positive or negative way. An increase in quality can be obtained, for instance, by not only processing the standard model, but also calculating alternative models in parallel. After processing, the best model can be chosen or the results of several models can be combined. The quality of the result can also decrease, however, due to parallelization (Müller 2008). The result of the map generalization is compared to a reference dataset by matching identical features. The match rate has a range from 58.4% for the smallest tile size to 59.6% for the biggest tile size, with 59.3% for the tile size of 3000 m. The partitioning leads to a small quality loss, because the algorithm has no information for objects beyond the partition borders, which itself influences the quality of objects near the borders. Therefore the smallest tile size results in a quality loss of 1.2%, whereas the optimum tile size results only in a not significant quality loss of 0.3%.

Based on these considerations the (research) question is not only formulated in terms of whether the quality is affected but how it is affected. This incorporates analyzing whether the used parallelization affects the quality globally, that is the whole dataset, or whether the quality changes locally (e.g. near tile borders, for a specific type of object, etc).

2.4 OVERALL SYSTEM ARCHITECTURE

Tasks can be executed in parallel within several system architectures (e.g. on different cores of CPUs, on graphics processing units (GPUs), in clusters, or in grid environments). Choosing the appropriate system architecture depends on many factors, such as already available infrastructure, amount of money that can be invested in new hardware or in their rental, desired speed, types of used algorithms, data size and distribution, software licensing issues, and security considerations. Concerning spatial data infrastructures (SDI), grid computing is a reasonable choice, because it complements the SDI. The distributed nature of the numerous and massive geospatial datasets in an SDI harmonizes with their distributed computing in a grid environment.

In order to specify the overall system architecture of a grid system more precisely, the distribution of both the parallelized (sub-) tasks and of the corresponding data as well as their combination in a workflow have to be defined. Several individual services are orchestrated in a workflow. The actual execution of a workflow is then performed by a workflow engine. In a grid environment three different approaches can be distinguished (Krüger, Kolbe 2008).

In the first approach the workflow engine is not aware of the grid (fig. 4), i.e. it does not know of the existence of the grid infrastructure. The services in the grid are therefore encapsulated by traditional web services. Subtasks are executed sequentially and the input and result datasets of the subtasks are always moved to a central location outside the grid. With this solution, calculations of single services can be distributed on several worker nodes, but no parallel data and information flow between them is realized. Worker nodes are individual computers on which jobs actually run. For communication and data transfer between two services no grid technologies can be used.

In the second approach the workflow engine is aware of the grid, i.e. it knows of the existence of the grid infrastructure and is able to make use of it accordingly. The engine executes different subtasks sequentially, but it is able to create several parallel processes for the same subtask (fig. 5). This solution allows full use of grid technologies for communication and data transfer between two services. However, in the long term an efficient coupling of grid services should be achieved.

The third approach allows for the efficient coupling of grid services. In order to

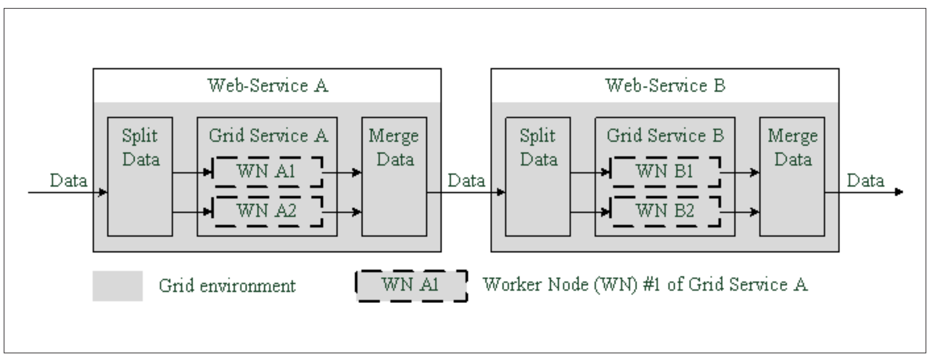


Figure 4: Workflow engine is not grid-aware

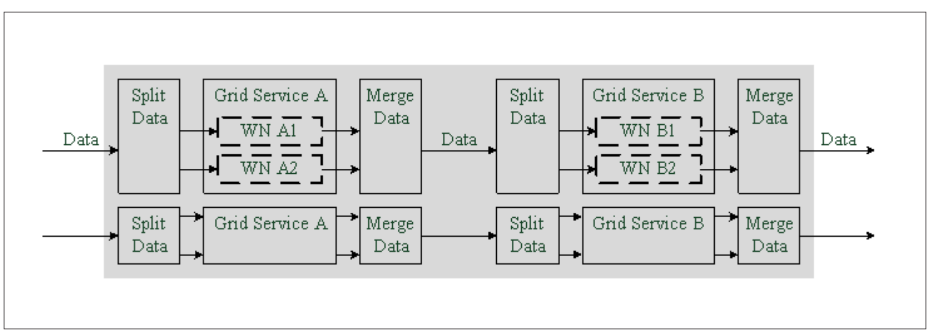


Figure 5: Workflow engine is grid-aware

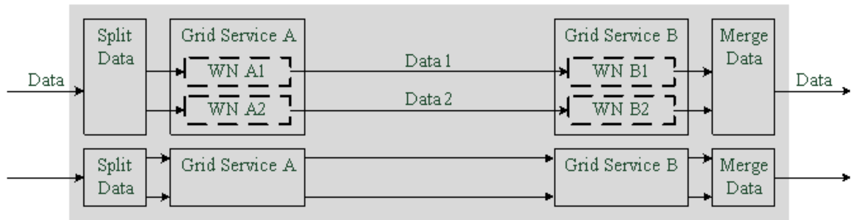


Figure 6: Transcendent gridification by compatible grid services

achieve this, grid services must be enabled to directly communicate with each other, especially concerning their accepted type of parallelization. Fig. 6 shows the two grid services A and B that are compatible concerning their parallelization and are therefore able to directly exchange the data between the respective worker nodes (WN). This way a time-intensive merge and subsequent split of the different datasets of the worker nodes is not necessary. Grid technologies can be utilized for parallel communication and data transfer directly between active worker nodes of subtasks. The concept of this parallelization approach has been named "transcendent gridification" by Krüger & Kolbe (2008).

Another aspect for an optimized use of the grid environment is the spatial distribution of files and algorithms within a grid environment. Choosing appropriate worker nodes within a grid environment depends, for instance, on the required memory size, the CPU speed and load, and the provided operating system. Requirements on the hardware must be fulfilled for high-performance workflow processing. The transfer of very large datasets can produce significant overhead, which has already been highlighted for the example of noise simulation within the GDI-Grid. Thus, services that process algorithms on very large datasets should be deployed near databases that store the datasets, and vice versa. In this context, "near" means not a real distance, but a connection providing preferably a high-speed data transfer. The actual selection of appropriate worker nodes for deploying a workflow within a grid environment is the work of a meta-scheduler. However, meta-schedulers are not incorporated in all grid environments and still require more research (Dong, Akl 2006).

2.5 ORCHESTRATION

The efficient orchestration of tasks in a complex workflow has two aspects. The first aspect covers the requirements that a workflow engine should satisfy in general. An important issue is thereby the support for basic control-flow patterns (e.g. split, merge, choice, loop), which are for instance defined by Russell et al. (2006).

The second aspect covers the technical issues that have to be solved in the context of grid computing for an SDI, which are highlighted in the remainder of this section. The most important aspect is that the types of (web) services used in a workflow differ between the GIS and the grid community. However, the respective standardization bodies, that are the Open Geospatial Consortium (OGC) for the GIS community and the Open Grid Forum (OGF) for the grid community, already collaborate in order to develop open standards that address the distributed computing needs of geospatial applications (Lee, Percivall 2008). The two approaches as well as the corresponding standards are presented in the following and an existing solution for the combination of both approaches is outlined.

In GIS the delivery and processing of spatial data are provided by special web services, which are standardized by the OGC. Such OGC compliant web services (OWS) are besides others:

- Web Processing Services (WPS) provide calculations on (very extensive) spatial data and GIS functionality, including access to calculations and/or computation models (OGC 2007a).
- Web Coverage Services (WCS) describe and deliver multidimensional coverage data (OGC 2008).
- Web Feature Services (WFS) provide data access functionality and operations on geographic features (OGC 2005).
- Catalogue Services (CS-W) handle the discovery and retrieval of spatial data, data stores and provide metadata of services (OGC 2007b).

In general, OWS are stateless and use proprietary security concepts. A web service description is stored in the corresponding capabilities document and is not (yet) realised by the Web Services Description Language (WSDL). The access to the services is performed over a "GetCapabilities" and a "describeX" operation, whereas "X" is a placeholder for the provided functionality. The service discovery is realised by CS-W. Messaging is realized by a combination of HTTP GET key value pairs using different data formats, like ASCII, binaries and OGC XML formats, e.g. the Geography Markup Language (GML) (Hobona et al. 2007).

Within a grid environment the Open Grid Services Architecture (OGSA) describes the architecture which is based on grid services. A grid service is a web service that provides a set of well-defined interfaces and that follows specific conventions. The interfaces address discovery, dynamic service creation, lifetime management, notification and manageability. Additionally grid services address authorisation and concurrency control (Foster et al. 2002). OGSA and SDI differ in some technologies. OGSA requires stateful web services and the service description is realised by WSDL. Most grid environments provide the Simple Object Access Protocol (SOAP) as messaging format. For service discovery the Monitoring and Discovery Service (MDS) and Universal Description Discovery and Integration (UDDI) are used. In contrast to the OWS, the security is based on the Grid Security Infrastructure (GSI), which uses asymmetric encryption per credentials (certificates which contains a public key) and a private key for decryption. (Foster, Kesselman 2004; Hobona et al. 2007).

In the GDI-Grid the approaches from the GIS and grid community are combined in a comprehensive workflow engine (Fleuren, Müller 2008). The engine is based on the Business Process Execution Language (BPEL). The aim of the GDI-Grid project is to be able to combine four different service types. These are standard web services described by WSDL, WSRF-based grid services, a special grid service that allows submitting and monitoring of simple executables as jobs in the grid, and traditional OWS which do not (yet) support WSDL. In order to grid-enable existing OWS providing computation-intensive functionality, they have to be ported to grid services. However, it is important that the workflow engine can also handle traditional OWS, for which a conversion is either impossible or not reasonable, because they are being provided by third parties or offer only simple functionalities.

2.6 ADOPTION OF EXISTING CODE

The last research question is a short one. The time investment for parallelizing an already existing task ranges from setting a compiler switch to a thorough code analysis and reconstruction process. In finance the measure of return on investment exists, which is calculated as the ratio of profit earned in relation to the amount of money invested (Feibel 2003). This concept can be transferred to parallelization either incorporating money or time.

3. CONCLUSIONS

Parallelization in a grid computing environment raises two questions: (1) how to parallelize tasks, which is independent from the grid, and (2) how to efficiently execute these tasks in the grid. We have identified six (research) questions that have to be answered by actual work in the context of parallelization in a grid computing environment. These questions have been explained in more detail by covering efficiency metrics, task and data parallelism, quality issues, system architectures, and workflows.

The question is not whether to parallelize, because we face massive geospatial data and more complex algorithms, but how to parallelize efficiently. But thereby we have to keep in mind that the absolute improvement is hard to measure exactly, because it depends on many factors.

4. ACKNOWLEDGEMENTS

The research described 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

Culler, D.E.; Singh, J.P.; Gupta, A. (1999): Parallel Computer Architecture – A Hardware/Software Approach. Morgan Kaufmann Publishers.

Dong, F.; Akl, S.G. (2006): Scheduling algorithms for grid computing: State of the art and open problems. http://research.cs.queensu.ca/TechReports/Reports/2006–504.pdf, accessed 08/09.

European Union (Ed.) (2002): Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise. Official Journal of the European Communities, L189.

Feibel, B.J. (2003): Investment Performance Measurement.Wiley.

Fleuren, T.; Müller, P. (2008): BPEL Workflows Combining Standard OGCWeb Services and Grid-enabled OGC Web Services. In: Proceedings of the 34th Euromicro Conference on Software Engineering and Advanced Applications, Sep 1–5, Parma, Italy.

Foster, I. (1995): Designing and Building Parallel Programs. Addison Wesley.

Foster, I. (2002): What is the Grid? A three point checklist. GridToday 1(6).

Foster, I.; Kesselman, C. (2004): The Grid 2: Blueprint for a new Computing Infrastructure. Elsevier.

Hobona, G.; Fairbairn, D.; James, P. (2007): Workflow Enactment of Grid-Enabled Geospatial Web Services. In: Proceedings of the 2007 UK e-Science All Hands Meeting, Sep 10–13, Nottingham, UK.

Krüger, A.; Kolbe, T.H. (2008): Mapping Spatial Data Infrastructures to a Grid Environment for Optimised Processing of Large Amounts of Data. In: Proceedings of the ISPRS Congress Beijing 2008, Jul 3–11, Beijing, China.

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

Meuer, H.W. (2008): The TOP500 Project: Looking Back Over 15 Years of Supercomputing Experience. Informatik-Spektrum 31(3), Springer.

Müller, P.G. (2008): Ableitung von Corine Land Cover Daten aus dem ATKIS-Basis-DLM. Master Thesis, Institute of Cartography and Geoinformatics, Leibniz Universität Hannover.

OGC (2005): Web Feature Service Implementation Specification: Version 1.1.0. OGC Document 04–094.

OGC (2007a): OpenGIS Web Processing Service: Version 1.0.0. OGC Document 05–007r7.

OGC (2007b): OpenGIS Catalogue Services Specification: Version 2.0.2. OGC Document 07–006r1.

OGC (2008): Web Coverage Service (WCS) Implementation Specification: Version 1.1.2. OGC Document 07–067r5.

Quinn, M.J. (2003): Parallel Programming in C with MPI and OpenMP. McGraw-Hill Professional.

Russell, N.; ter Hofstede, A.H.M.; van der Aalst, W.M.P.; Mulyar, N. (2006): Workflow Control-Flow Patterns: A Revised View. http://www.workflowpatterns.com/documentation/documents/BPM-06–22.pdf, accessed 08/09.

Vaquero, L.M.; Rodero-Merino, L.; Caceres, J.; Lindner, M. (2009): A Break in the Clouds: Towards a Cloud Definition. ACM SIGCOMM Computer Communication Review 39(1).

Werder, S. (2009): Formalization of Spatial Constraints. In: Proceedings of the 12th AGILE International Conference on Geographic Information Science, Jun 2–5, Hannover, Germany.

University of Bonn, Department of Geography

GRIDIFICATION OF OGC WEB SERVICES: CHALLENGES AND POTENTIAL

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 G-OWS Working Group comprised of the projects CYCLOPS, GENESI 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 Fritzsch and Hiller (Fritzsch, 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.

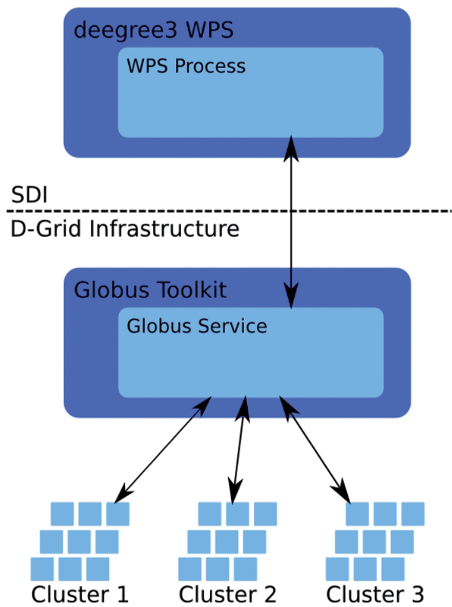


Figure 1: Model for a grid enabled WPS

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 GapSLC 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.

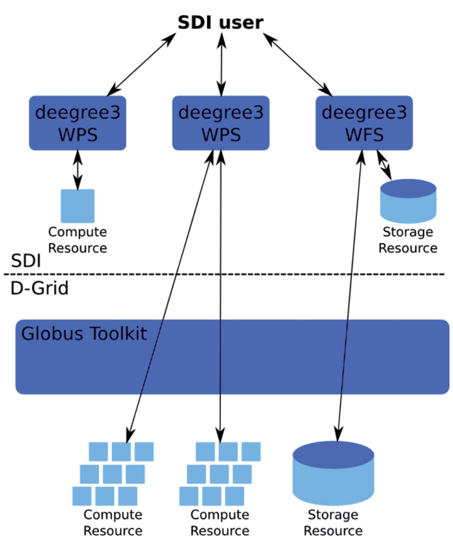


Figure 2: Architecture of an SDI including grid-enabled services

5. CONCLUSION

The aforementioned method for grid-enabling OGC Web Services is a straight-forward 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

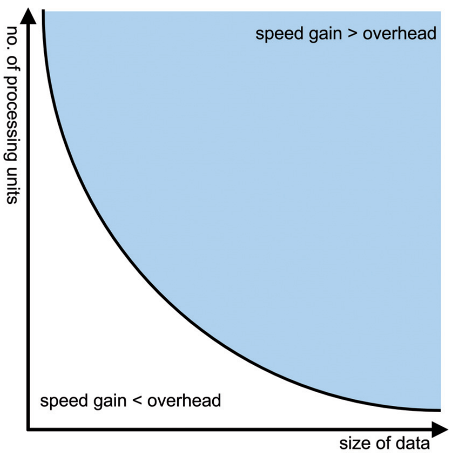


Figure 3: Estimation of a break-even function

and by Level of Detail (Chaudry and 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.

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, 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.

STFC e-Science Centre, Rutherford Appleton Laboratory

AN APPROACH TO ENCAPSULATION OF GRID PROCESSING WITHIN AN OGC WEB PROCESSING SERVICE

Dr. Andrew Woolf, Dr. Arif Shaon

Abstract: Increasingly complex environmental problems and the growth in accessible geospatial data in interoperable formats is leading to the investigation of Grid processing as a potential tool for advanced geo-processing within a standards-based environment. The OGC Web Processing Service (WPS) is an open standardised interface for geo-processing, compatible with technologies being adopted by spatial data infrastructures. We investigate the integration of Grid computing capability within a WPS. Our approach is based on the use of the Job Submission Description Language (JSDL) being developed by the Grid community as a standardised description of computational jobs to be executed on heterogeneous Grid platforms. We develop a mechanism to integrate JSDL resource requirement descriptors within a standard WPS request, and show how process execution and status querying may be proxied to a Grid environment through the WPS interface. A proof-of-concept implementation is developed using an atmospheric particle tracing application and the UK National Grid Service.

Keywords: Grid, OGC, Web Processing Service, WPS, Job Submission Description language, JSDL, spatial data infrastructure

// EINE METHODE ZUR EINBETTUNG DER GRID-TECHNOLOGIE IN EINEN OGC WEB PROCESSING SERVICE

// Zusammenfassung: Die zunehmenden Umweltprobleme und das stetige Wachstum an zugänglichen raumbezogenen Daten in interoperablen Formaten führte zur Entwicklung der Grid-Technologie als potentielle Möglichkeit für eine fortgeschrittene Geodatenverarbeitung innerhalb einer standardisierten Umgebung. Der OGC Webverarbeitungsdienst (WPS) stellt dabei eine offene standardisierte Schnittstelle für die Geodatenverarbeitung dar, welche mit weiteren Technologien bezogen auf raumbezogene Daten kompatibel ist. Die Untersuchung bezog sich auf die Integration von Grid Computing mit WPS. Der Ansatz beruhte auf der Verwendung der Job Submission Description Language (JSDL) durch die Grid-Gemeinschaft als eine standardisierte Beschreibung von rechenintensiven Jobs auf heterogenen Grid-Plattformen. Dabei wurde ein Mechanismus entwickelt, um JSDL mit WPS zu verbinden.

Schlüsselwörter: Grid, OGC, Web Processing Service, WPS, Job Submission Description language, JSDL, spatial data infrastructure

Authors

Dr. Andrew Woolf
Environmental Informatics lead
STFC e-Science Centre
Rutherford Appleton Laboratory
Chilton, Oxon., OX11 0QX
UNITED KINGDOM
E: andrew.woolf@stfc.ac.uk

Dr. Arif Shaon
Software Research Scientist
STFC e-Science Centre
Rutherford Appleton Laboratory
Chilton, Oxon., OX11 0QX
UNITED KINGDOM
E: arif.shaon@stfc.ac.uk

1. INTRODUCTION

The Geospatial community requires an ability to analyse and process geographic datasets using a combination of spatial software and analytical methods. Increasingly complex geo-processing operations are driving a demand for greater computing capability – it is no longer always possible to rely on a desktop GIS to perform spatial analysis of interest, especially where very large datasets are concerned, or the results of complex simulation models need to be integrated.

There is also a need to be able to share processing and analytical capability across the community in an interoperable fashion. A research group having developed a new analysis technique may wish to make it available to the broader research community, or the owner of a high-performance computing resource may wish to allow use by collaborators for a defined purpose.

Within the context of emerging developments in spatial data infrastructures (SDIs, for instance INSPIRE in Europe), there is a significant potential for Grid computing to play a major role (Padberg, Kiehle 2009). The three key elements of SDIs are metadata, interoperable data, and network services. ISO 19119 (ISO 2006) provides a wide taxonomy of relevant services, including model/information management services, workflow/task management services, and processing services (for metadata, spatial, thematic and temporal processing). Within all of these service categories, Grid computing could play a role. For instance, the Grid community is very active in the area of workflow and service chaining, and dynamic information services that expose the current state of the Grid 'fabric' are especially relevant for real-time geospatial information, e.g. from environmental sensors. For standard geospatial data services (e.g. WFS, WCS), the suite of Grid data management middleware offers possibilities to facilitate integration, for example by combining data from multiple heterogeneous databases, and transforming to GML representations. Conversely, the use of geospatial data and services within Grid infrastructures can be useful for 'e-science' and other advanced computing applications (SEE-GEO 2008).

2. THE PROBLEM

The recent development of web services provides a powerful technical solution to simplify the sharing of computational resources and algorithms. Software may be exposed for use through simple web-based protocols, and chains of such services may be orchestrated in value-adding workflows. These 'service-oriented architectures' provide a new paradigm for enabling collaboration.

To date, most web-based geo-processing services take a traditional 'RESTful' 'stateless' view of resources: data are not distinguished from their access service; asynchronous interaction sequences are poorly supported; there is no notion of resource types (e.g. computational) other than data and service instances; there are no efficient means for handling resource-intensive processes. For example, an application for predicting future global climate change may involve analysing large volumes of past weather data collected over a number of years, and running compute-intensive forecast models. Such a complex application will require a large amount of computational resource, such as disk space, memory and CPU power. Executing this application through a standard web service allows no scheduling capability, and could utilise all available computational resources, resulting in significant delays for other processes.

In addition, typical geo-processing web services do not take a sophisticated approach to security-related issues associated with the underlying processes and datasets. In fact, service providers often rely on ad-hoc access control at the client level to ensure security of the resources exposed. This leads to non-interoperability in workflows that require interaction between multiple services, each with different security protocols.

The aforementioned limitations are precisely the niche of Grid computing (Foster, Kesselman 2004). While Grid architectures and technologies vary, they share in common an attempt to abstract models of stateful resource (data, storage, compute, etc.) within a standardised framework (Foster et. al. 2002) in order to simplify the construction on demand of complex workflows. For instance, using Grid technology, execution of a large-scale parallel process may be accelerated by load-balancing across different Grid nodes (Thain et. al. 2003). In addition, Grid middleware enables allocation of specific amounts of computational resource, such as disk space, to a particular process (Huedo et. al. 2004). In terms of access control, many Grid architectures employ a common security framework (e.g. the Grid Security Infrastructure (GSI) (Foster et. al. 1998)) to provide secure, robust and interoperable communications. Authentication in the GSI architecture is based on a public-key infrastructure (PKI) with x.509 certificates containing information for identifying users or services.

From this perspective, geoprocessing applications and services should benefit from integration with Grid computing resources and technologies to enable applications to scale out. This goal is reflected in a Memorandum of Understanding (MoU) on collaboration signed in late 2007 (OGC 2007) by the Open Geospatial Consortium (OGC (OGC 2009)) and the Open Grid Forum (OGF (OGF 2009)). The initial targets of the collaboration are: integration of the OGC Web Processing Service (WPS) with Grid processing and workflow tools, and integration of geospatial catalogues with Grid data movement tools. From a wider perspective, it is intended to promote the use of Grid-based resources within the Geospatial community.

3. THE PRINCIPLES OF GRID-ENABLING WPS

The OGC standard Web Processing Service (WPS (Schut 2007)) interface is designed to facilitate publishing, discovery and use of geospatial computational processes in a standardised and interoperable manner. The WPS interface is implemented as a Web Service, with operations for: describing process functionality in terms of inputs and outputs, triggering its execution, monitoring its status and finally retrieving its output.

An analysis of WPS and Grid processing paradigms indicate that both provide a remote interface for invoking computational processes. WPS functionality includes remote data inputs/outputs, progress monitoring, and asynchronous delivery. Beyond this, Grid job submission mechanisms typically add the ability to stage data, and specify required computational resource requirements (Woolf 2006).

The conceptual overlaps and differences between WPS and Grid processing paradigms can also be demonstrated by a comparison between the WPS specification and the Job Submission Description Language (JSDL, (Anjomshoaa et. al. 2005)) (Figure 1), an OGF specification that provides a standardised description of a computational job and the resources (data and computational) it requires. JSDL provides a normative XML schema for describing a computational job, including job description and identification, the software application to be run, resource requirements (filesystem parameters, disk space, operating system, CPU, etc.), and data staging instructions

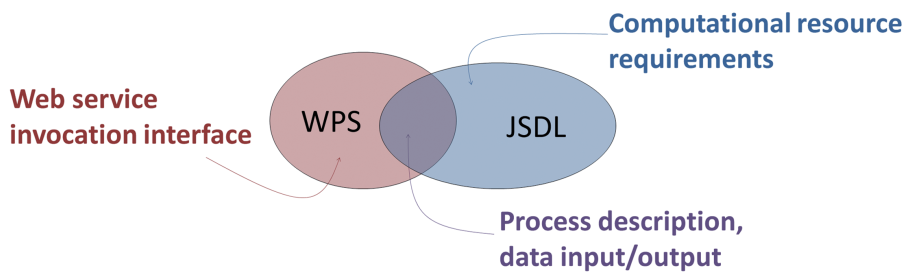


Figure 1: Conceptual overlaps and differences between WPS and JSDL

for input and output. It also includes standardised POSIX extensions for executable filename, command-line arguments, etc. The purpose of JSDL is to enable a standardised specification of job submission to Grid infrastructures irrespective of the back-end schedulers or resource managers used. JSDL job descriptions may be submitted to consuming services like GridSAM (GridSAM 2009).

At the specification level, both WPS and JSDL include process description information (WPS: Identifier; JSDL: JobIdentification/JobName), and process outputs and inputs (WPS: DataInputs; JSDL: POSIXApplication/{Input,Argument,Output}, DataStaging/Source). These analogous parameters are sufficient to produce a valid JSDL document from a standard WPS request; this is fundamental to the approach presented in this paper. Thus, standard WPS data input and output parameters may be used to generate equivalent JSDL DataStaging or POSIXApplication/Input elements. Similarly a WPS process identifier may be mapped to a JobName and POSIXApplication in JSDL. What WPS lacks are any standard input parameters for specifying required computational resources. Vice versa, while JSDL enables the specification of a computational job and its resources, it lacks a standardised web service interface for enabling the submission of processing requests. Our solution combines the unique aspects of WPS and JSDL, using the overlaps to maximise the integration of WPS in a Grid context.

4. EXISTING APPROACHES

The most common approach to grid-enabling WPS (e.g. (Di et. al. 2002), (Nativi et. al. 2009), (Baranski et. al. 2008)) involves encapsulating Grid "characteristics" within a standard WPS instance without compromising its compliance with the OGC specification WPS (Schut 2007). This approach is often referred to as "profiling" of WPS. The nature of the encapsulated Grid "characteristics" varies between implementations. However, only a very few instances of the existing WPS-Grid profiling approaches incorporate the ability to specify computational Grid resources (disk space, CPU power, memory) needed to run an application. For example, this has been used in (Baranski 2008) to enable executing geospatial processes on a Grid backend through a standard WPS server. However, the ability to specify computational resource for a process is restricted to a few predefined JSDL specific parameters (e.g. DiskSpace, TotalCPUCount). In addition, it is not possible to query the states associated with a process except its execution status. Future WSRF-based approaches may enable other aspects of Grid 'state' to be encapsulated.

5. A WPS GRID PROFILE

We have developed a WPS-Grid "profiling" solution that combines the simple web service interface of WPS with the ability of JSDL to specify resource requirements for a large computational process. The WPS-Grid profile enables Grid computing resources to be encapsulated behind a WPS: the WPS acts as an interface to the backend computing resources on the Grid.

In contrast to the existing approaches, our WPS-Grid profile is achieved by enabling encoding of JSDL-related information directly as part of the WPS Execute request defined in version 1.0 of the WPS specification (Schut 2007). The JSDL-related information is then used by the WPS instance for constructing a JSDL document, and submitting the process for execution to a Grid backend through a JSDL-enabled Grid client, such as GridSAM (Figure 2). The rationale of this approach is to incorporate JSDL handling ability into a WPS while ensuring its conformance to the WPS specification. This allows Grid computing capability to be combined with the simplicity of the WPS interface. Thus, this approach improves the mechanism adopted in previous work, by providing users with the flexibility of specifying the computational resource requirements for a process.

5.1 WPS EXECUTE REQUEST

Two broad options are available for including JSDL-related information within the WPS Execute request:

- where JSDL-related parameters are regarded as conceptually distinct from other WPS input parameters: through a specific 'JSDL' parameter as part of the WPS DataInputs
- where JSDL-related parameters are regarded simply as additional WPS input parameters: through individual WPS DataInput parameters

We outline in the next two sections (5.1.1 and 5.1.2) these alternative approaches to specifying JSDL input parameters in a WPS Execute request.

5.1.1 SPECIFIC 'JSDL' INPUT PARAMETER

Typically, JSDL-related WPS input parameters will be concerned with specifying resource requirements for executing the process on a computing Grid backend, e.g. required disk space, CPU time, etc. Conceptually, such parameters are not process-related; they only determine whether, and perhaps how quickly, a result is computed. The output results themselves are independent of these

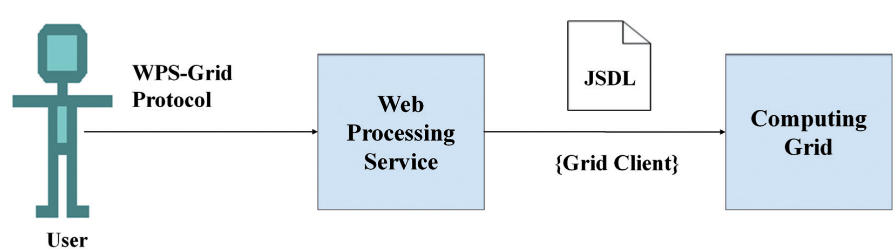


Figure 2: WPS-Grid generates JSDL document for submission to Grid

parameters. There is a strong case, therefore, to regard the JSDL-related input parameters as different in nature to other process-related WPS input parameters. In this case, a specific 'JSDL' WPS input parameter may be used to specify resource requirements. The use of this special parameter 'JSDL' enables a WPS instance to distinguish between JSDL parameters and the other (process-related) parameters.

JSDL input parameters may be supplied in a conformant XML format, or in a proposed microformat, as described in the next two sections respectively.

5.1.1.1 FULL JSDL DOCUMENT OR VALID SNIPPET

In this case, a URL may be provided referencing a complete pre-created JSDL document, or valid snippet (e.g. specifying just the JSDL 'Resource' XML elements), Listing 1.

Advantages of this approach are that the JSDL input may be validated against the JSDL schema. Alternatively, the JSDL document or snippet may be URL-encoded and included directly within the WPS request, Listing 2. Note that such a JSDL document or document fragment may also be supplied very naturally through a HTTP POST request (Listing 3), facilitating the service invocation in XML-based workflows.

5.1.1.2 JSDL ELEMENTS MICROFORMAT

As an alternative to providing a full valid JSDL snippet, JSDL-related parameters may be represented more simply as key-value pairs, with the keyword deriving directly from the relevant JSDL element name. An Xpath-like syntax can be used (replacing '/' with '#') to specify the JSDL element name.

These key-value-pair JSDL parameters are specified in a microformat as the value of the complex WPS DataInput parameter, 'JSDL' (Listing 4). They are enclosed within square brackets [], with the WPS 'Format' attribute set to the special value 'text/kvp'. (Note that the entire URL must be URL-encoded as per IETF RFC 1738, although for clarity the examples in the listings below are left unencoded.) As an even simpler alternative to specifying full Xpath JSDL element names, simplified keywords may be designated for predefined common JSDL elements (e.g. 'Source_URI' for the JSDL 'DataStaging/Source/URI' XML element).

Multiple values for a repeatable JSDL parameter are separated using "," (Listing 5).

```
<wps:Execute service="WPS" version="1.0.0" xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
..\wpsExecute_request.xsd">
    <ows:Identifier>WeatherGenerator</ows:Identifier>
    <wps:DataInputs>
        <wps:Input>....</wps:Input>
        <wps:Input>
            <ows:Identifier>JSDL</ows:Identifier>
            <wps:Data>
                <wps:ComplexData encoding="" schema="http://server/jsdl.xsd">
                    <jsdl:JobDefinition
                    xmlns:jsdl="http://schemas.ggf.org/jsdl/2005/11/jsdl">
                        <jsdl:JobDescription>
                            <jsdl:Resources>
                                <jsdl:TotalDiskSpace>
                                    <jsdl:Exact>5</jsdl:Exact>
                                </jsdl:TotalDiskSpace>
                                < jsdl:TotalCPUCount>
                                    < jsdl:Exact>1</ jsdl:Exact>
                                </ jsdl:TotalCPUCount>
                            </ jsdl:Resources>
                        <jsdl:JobDescription/>
                    </jsdl:JobDefinition>
                </wps:ComplexData>
            </wps:Data>
        </wps:Input>
    </wps:DataInputs>
    <wps:ResponseForm>
        <wps:ResponseDocument storeExecuteResponse="true" status="true"/>
    </wps:ResponseForm>
</wps:Execute>
```

Listing 3: WPS Execution request with HTTP-POST JSDL snippet

```
http://foo.bar.1/wps?version=1.0.0&request=Execute&service=WPS&Identifier=WeatherGenerator&DataInput=other_
inputs=xxx;JSDL=http://www.foo.com/myfoo.jsdl@Format=text/xml&storeExecuteResponse=true&status=true
```

Listing 1: WPS Execution request with URL to JSDL document

```
http://foo.bar.1/wps?version=1.0.0&request=Execute&service=WPS&Identifier=WeatherGenerator&DataInput=other_
inputs=xxx;JSDL=%3CJobDefinition%3E%3CJobDescription%3E%3CResources%3E%3CTotalDiskSpace%3E%3CExact%3E5
%3C%2FExact%3E%3C%2FTotalDiskSpace%3E%3CTotalCPUCount%3E%3CExact%3E1%3C%2FExact%3E%3C%2FTotalCP
UCount%3E%3C%2FResources%3E%3C%2FJobDescription%3E%3C%2FJobDefinition%3E@Format=text/xml@Schema=h
ttp://server/jsdl.xsd&storeExecuteResponse=true&status=true
```

Listing 2: WPS Execution request with URL-encoded JSDL snippet

```
http://foo.bar.1/wps?version=1.0.0&request=Execute&service=WPS&Identifier=WeatherGenerator&DataInput=other_
inputs=xxx;JSDL=[TotalDiskSpace=5;TotalCPUCount=1]@Format=text/kvp&storeExecuteResponse=true&status=true
```

Listing 4: WPS Execution request with microformat for JSDL input parameters

```
http://foo.bar.1/wps?version=1.0.0&request=Execute&service=WPS&Identifier=WeatherGenerator&DataInput=other_
inputs=xxx;JSDL=[DataStaging#Target#URI=http://www.foo.com/myfoo.xml,http://www.foo.com/myfoo2.xml]@Forma
t=text/kvp&storeExecuteResponse=true&status=true
```

Listing 5: WPS Execution request with multiple JSDL element values

```
http://foo.bar.1/wps?version=1.0.0&request=Execute&service=WPS&Identifier=WeatherGenerator&DataIn
put=other_inputs=xxx;TotalDiskSpace=5;TotalCPUCount=1&storeExecuteResponse=true&status=true
```

Listing 6: WPS Execution request with individual JSDL input parameters

```
http://foo.bar.1/wps?version=1.0.0&request=Execute&service=WPS&Identifier=WeatherGenerator&DataInput=otheri
nput@datatype=string;JSDL=[TotalDiskSpace=5]@Format=text/kvp;MyProxy=[username=xxxx;password=xxxx;host=xxx
;port=xxxx]@Format=text/kvp&storeExecuteRepose=true&status=true
```

Listing 7: WPS Execution request with MyProxy input parameters

5.1.2 INDIVIDUAL JSDL INPUT PARAMETERS

Rather than regarding JSDL parameters as special, they may be regarded as simply additional WPS literal input parameters individually. In this case, they may be included as key-value pairs in a manner similar to that described in section 5.1.1.2 above, except as individual WPS parameters instead of within an aggregate microformat 'JSDL' parameter (Listing 6).

5.2 PROGRESS MONITORING

A JSDL-enabled WPS is also able to monitor the status of the process on the Grid, for instance using the same Grid client used for the job submission (e.g. GridSAM). The response to a WPS Execute request contains an element indicating the status of the request (process accepted, started, paused, succeeded, or failed). In addition, for longer-running asynchronous requests, the response may include a URL reference that can be polled for status updates. The status returned by monitoring job progress with the Grid middleware may be mapped to the WPS status indicator. For instance, GridSAM is a Grid middleware component that accepts JSDL job submission requests for execution on a Grid. It may also be used to monitor job progress, returning a status of: pending, staging-in, staged-in, active, executed, staging-out, staged-out, done, failed, and undefined. These may be mapped to the abovementioned WPS status codes to enable progress monitoring in the standard WPS manner.

5.3 GENERATION OF JSDL DOCUMENT

The generation by the WPS server of a conformant and appropriate JSDL document for submission to a Grid backend is implementation-defined. Typically, an internal configuration would assign a default JSDL to a specific WPS process, with user-supplied JSDL parameters over-riding the defaults. As mentioned earlier, in fact a conventional WPS request contains sufficient information to generate a minimal compliant JSDL document, even without the Grid-specific WPS extensions proposed above. Such a job description, of course, would include only job identification information and data inputs; it would not include parameters specifying computational resource requirements. An XSLT approach may be used to generate a JSDL document from a template based on WPS request parameters (Baranski 2009).

5.4 SECURITY

Secure access to OGC web services is the subject of considerable ongoing work. Rather than develop a divergent solution, a very light-weight 'placeholder' approach has been taken to security within the WPS-Grid profile. We allow a user to embed MyProxy (Basney et. al. 2005) parameters (host, port, username, password) within the WPS request using the microformat encoding mechanism described earlier (5.1.1.2), as the value of a special DataInput parameter, 'MyProxy' (Listing 7). These are processed by the WPS server and used at job submission for authentication.

Including sensitive information, such as a user's MyProxy credentials in a WPS request does pose security risks. Therefore, this approach assumes that an implementation would incorporate an appropriate transport-layer security protocol (e.g. SSL) to ensure the security of MyProxy information in the WPS request.

6. IMPLEMENTATION

A Grid-enabled WPS service has been implemented within the OGC's OWS-6 activity (Baranski 2009) as a proof-of-concept using an atmospheric particle-tracing 'trajectory service' (BADC 2009) and deployment on the UK National Grid Service (UK NGS (NGS 2009)), which is explicitly mentioned as a target Grid infrastructure in the OGC-OGF MoU. The implementation of this Grid-enabled WPS is fully compliant with the OGC WPS specification 1.0.0, and uses Python as the underlying programming language and Pylons (Pylons 2009) as the integrated web development framework. At an architectural level, it depends on a number of other services

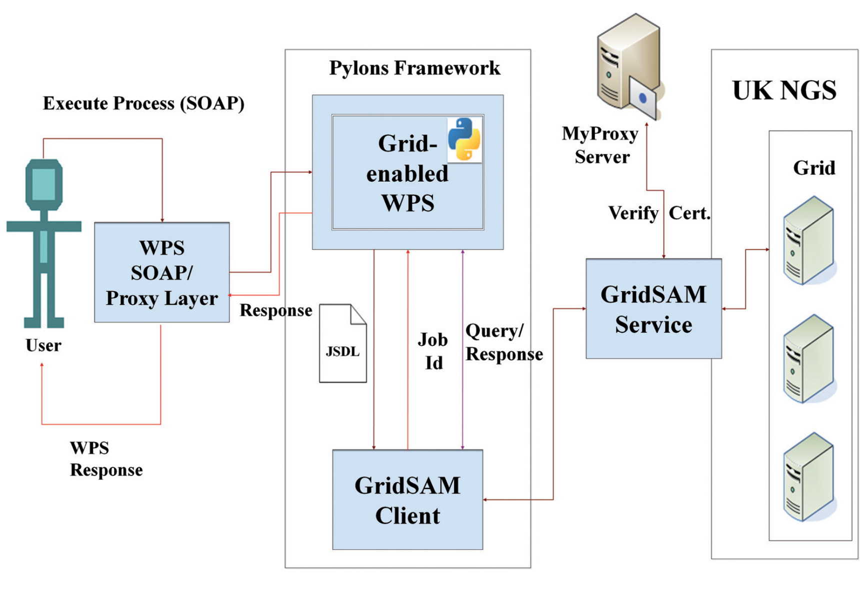


Figure 3: An Architectural View of the Grid-enabled WPS

and components (Figure 3) to enable invocation and controlling of Grid-based processes through standard WPS requests. We outline below some of the key components.

6.1 GRID-ENABLED WPS

The implemented WPS conforms to the Grid profile described in this paper, using a specific 'JSDL' input parameter (5.1.1) for JSDL-related information, and using the simplified microformat (5.1.1.2) key-value-pair syntax. A HTTP GET binding was implemented for WPS requests, although a SOAP wrapper was also developed (see 6.3 below).

The deployed application ('trajectory service') makes use of very large four-dimensional (space and time) gridded fields of analysed atmospheric parameters (windspeed and direction, pressure, humidity, etc.) to trace the trajectory of a hypothetical particle released at a given location and time. While such a service is not suitable for full dispersion modelling it was used in a demonstration scenario to simulate the dispersal of a plume of toxic gas released in an emergency incident (Baranski 2009). The overall scenario integrated the trajectory service with other services (including a plume service for simulating the gas plume from a calculated trajectory) using a workflow engine.

6.2 GRIDSAM SERVICE

The aforementioned GridSAM client (5.2) submits the JSDL job description to the UK NGS GridSAM service (installed in the prototype on the Oxford node of the UK NGS), which then executes the requested job on the NGS. This service also receives queries (for example job status check, etc.) from the GridSAM client and responds to them by liaising with the Grid with which it is associated. The GridSAM service is also responsible for retrieving and verifying users' Grid certificates using the MyProxy credentials received along with the JSDL job description from the GridSAM client.

6.3 WPS SOAP / PROXY LAYER

We also implemented a Java-based WPS SOAP/Proxy layer that provides a SOAP wrapper outside the network firewall to proxy SOAP requests through HTTP GET requests to the Grid-enabled WPS. The SOAP interface provided by the SOAP wrapper conforms to the WPS specification 1.0.0. In addition, this layer is also used for forwarding other HTTP GET requests to the Grid-enabled WPS, such as a request for downloading process output and status check requests for a process.

7. CONCLUSIONS & FUTURE WORK

Grid computing provides an efficient means of executing resource-intensive processes, and thus should be beneficial to the Geospatial community that has an increasing need for performing highly complex geo-processing operations involving large geographical datasets. The WPS-Grid profiling approach presented in this paper demonstrates the feasibility of integrating Grid capability within standard geo-processing web services, such as the Web Processing Service. However, there is considerable scope for further work in this area. For example, the suite of OGC web services could be refactored around a resource-oriented view of data (Woolf, Woodcock 2005) within the Grid infrastructure using technologies such as the Web Services Resource Framework (WSRF (Banks 2006)). WSRF standardises the representation of stateful resources within web services. Applied to the Web Processing Service, for instance, computational process instances could be regarded as resources, with associated state corresponding to current job status, computational resource utilisation, etc.

It may also be useful to look into replacing GridSAM as the grid middleware for consuming JSDL documents and job management on the Grid, with any middleware conforming to the 'HPC Basic Profile' (Dillaway et. al. 2007), subject to associated restrictions on the allowable scope of JSDL. The HPC Basic Profile defines a base level of interoperability for high-performance computing systems within a Grid environment, by specifying a restricted use of several OGF specifications, notably JSDL for job description, and the 'OGSA Basic Execution Service' (Foster et. al. 2008) for job scheduling and management. A number of core JSDL elements must be supported; these include job description elements (JobIdentification, JobName, JobProject) and computational resource elements (CandidateHosts, ExclusiveExecution, OperatingSystem, CPUArchitecture, TotalCPUCount). These core elements are obvious candidates to standardise as simplified keywords within the WPS-Grid profile, as described earlier (5.1.1.2), avoiding the JSDL XPath syntax.

Finally, future evolution of the WPS-Grid profile will need to align with best practice for adding security to other OGC service interfaces, possibly by harmonising the Grid Security Infrastructure (GSI) and the current OGC approaches to security. ◀

(References please read next page)

IMPRESSUM // PUBLICATION INFORMATION:

GIS.SCIENCE – Die Zeitschrift für Geoinformatik ISSN 1430–3663 // Herausgeber: Dirk Schmidbauer // Redaktion: Dipl.-Ing. Lutz Kreßner, lutz.kressner@abcverlag.de, Tel. +49(0)0381498-3211, Fax +49(0)0381498-3202 // Hauptschriftleiter: Prof. Dr.-Ing. Ralf Bill, ralf.bill@uni-rostock.de, Tel. +49(0)0381498-3200, Fax +49(0)0381498-3202 // Editorial-Board: Prof. Dr. Lars Bernard, TU Dresden; Dr. Andreas Donaubauer, ETH Zürich; Prof. Dr. Max Egenhofer, University of Maine Orono; Prof. Dr. Manfred Ehlers, Universität Osnabrück; Prof. Dr. Klaus Greve, Universität Bonn; Dr. Stefan Lang, Universität Salzburg, Prof. Dr. Stephan Nebiker, Fachhochschule Nordwestschweiz, Prof. Dr. Matthäus Schilcher, TU München, Prof. Dr. Josef Strobl, Universität Salzburg // Kontakt Redaktion Rostock: +49(0)0381498-3211, lutz.kressner@abcverlag.de | Konzeption und Layout: Dipl. Des. Birgit Speiser // Anzeigen: Matthias Knecht, matthias.knecht@abcverlag.de, +49(0)06221/75704–105 // Leserservice: Ingrid Gimbel, ingrid.gimbel@abcverlag.de, +49(0)06221/75704–100 // GIS.SCIENCE erscheint im abcverlag GmbH, Waldhofer Str. 19, 69123 Heidelberg, Tel. +49(0)6221/75704–100, Fax +49(0)6221/75704–109, info@abcverlag.de // Geschäftsführung: Dirk Schmidbauer, HRB 337388, Ust-ID: DE 227 235 728 // Druck: abcdruck, Heidelberg // Erscheinungsweise: 12 x jährlich, davon 4 Ausgaben GIS.SCIENCE plus 2 Sonderthemenhefte GIS.TRENDS+MARKETS // Jahresabonnement (12 Hefte): Inland 157,25 EUR inkl. Versandkosten, Ausland 166 EUR inkl. Versandkosten, Studenten/Auszubildende 89,- EUR inkl. Versandkosten, Mitglieder des Deutschen Dachverbandes für Geoinformation e.V. (DDGI) erhalten das Abo im Rahmen ihrer Mitgliedschaft // Bezugszeitraum: Das Abonnement läuft zunächst für 12 Monate. Zum Ablauf des ersten Bezugsjahres kann das Abonnement zum Ende des Kalenderjahres mit einer Kündigungsfrist von 3 Monaten gekündigt werden. Bei Nichterscheinen aus technischen Gründen oder höherer Gewalt entsteht kein Anspruch auf Ersatz.

References

Anjomshoaa, A.; Brisard, F.; Drescher, M.; Fellows, D.; Ly, A.; McGough, S.; Pulsipher, D.; Savva, A. (2005): "Job Submission Description Language (JSDL) Specification, Version 1.0", OGF document GFD-R.056. Available at: http://www.gridforum.org/documents/GFD.56.pdf, Accessed 08/09.

BADC (2009): "BADC Trajectories – Main Page". http://badc.nerc.ac.uk/community/trajectory, Accessed 08/09.

Banks, T. ed. (2006, ed.): "Web Services Resource Framework (WSRF) – Primer v1.2", OASIS document wsrf-primer-1.2-primer-cd-02. Available at: http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf, Accessed 08/09.

Baranski, B. (2008): "Grid Computing Enabled Web Processing Service". In: Pebesma, E., Bishr, M. & T. Bartoschek (Eds.): GI-Days 2008 – Proceedings of the 6th Geographic Information Days. IfGIprints Nr. 32, Institut für Geoinformatik, Münster.

Baranski, B. ed. (2009, ed.): "OGC® OWS-6 WPS Grid Processing Profile Engineering Report", OGC document 09–041r2.

Basney, J.; Humphrey, M.; Welch, V. (2005): "The MyProxy online credential repository", In: Software: Practice and Experience, 35(9), 2005, 801–816. See also http://grid.ncsa.uiuc.edu/myproxy, Accessed 08/09.

Di, L.; Chen, A.; Yang, W.; Zhao, P. (2002): "The Integration of Grid Technology with OGC Web Services (OWS) in NWGISS for NASA EOS Data". Available at: http://laits.gmu.edu/~achen/download/Liping_Aijun.pdf, Accessed 08/09

Dillaway, B.; Humphrey, M.; Smith, C. (ed.); Theimer, M.; Wasson, G. (2007, ed.): "HPC Basic Profile, Version 1.0 ", OGF document GFD-R-P.114. Available at: http://www.ogf.org/documents/GFD.114.pdf, Accessed 08/09.

Foster, I.; Kesselman, C.; Tsudik, G.; Tuecke, S. (1998): "A Security Architecture for Computational Grids", 5th ACM Conference on Computer and Communications Security, 1998, 83–91.

Foster, I.; Kesselman, C.; Nick, J.; Tuecke, S. (2002): "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Globus Project, 2002. Available at: http://www.globus.org/research/papers/ogsa.pdf, Accessed 08/09.

Foster, I.; Kesselman, C. (2004, eds.): The Grid 2: Blueprint for a New Computing Infrastructure, Morgan-Kaufmann, 2004.

Foster, I.; Grimshaw, A.; Lane, P.; Lee, W.; Morgan, M.; Newhouse, S.; Pickles, S.; Pulsipher, S.; Smith, C.; Theimer, M. (2008): "OGSA Basic Execution Service Version 1.0", OGF document GFD-R.108. Available at: http://www.ogf.org/documents/GFD.108.pdf, Accessed 08/09.

GridSAM (2009): "GridSAM – GridSAM – Grid Job Submission and Monitoring Web Service". http://gridsam.sourceforge.net, Accessed 08/09.

Huedo, E.; Montero, R.S.; Llorente, I.M. (2004): "A Framework for Adaptive Execution on Grids", In: Software – Practice and Experience, 34(7), 2004, 631–651.

ISO (2006): Geographic information – Services, ISO 19119, International Organisation for Standardisation, 2006.

Nativi, S.; Verlato, M.; Pacini, F.; Pugliese, G.; Mazzetti, P.; Angelini, V. (2009):" G-OWS: the Grid-enabled Open-Geospatial Web Services", In: Geophysical Research Abstracts, Vol. 11, EGU2009–3424, 2009. Proceedings of the EGU General Assembly 2009, Vienna, Austria. Available at: http://meetingorganizer.copernicus.org/EGU2009/EGU2009–3424.pdf,Accessed 08/09

NGS (2009): "National Grid Service". http://www.ngs.ac.uk, Accessed 08/09.

OGC (2007): "Open Grid Forum and OGC Sign Memorandum of Understanding". http://www.opengeospatial.org/pressroom/newsletters/200801#C5, Accessed 08/09.

OGC (2009): "Welcome to the OGC Website | OGC®". http://www.opengeospatial.org/, Accessed 08/09. OGF (2009): "Open Grid Forum". http://www.ogf.org/, Accessed 08/09.

Padberg, A.; Kiehle, C. (2009): "Towards a grid-enabled SDI: Matching the paradigms of OGC Web Services and Grid Computing", GSDI 11 World Conference, Rotterdam, June, 2009. Available at: http://www.gsdi.org/gsdi11/papers/pdf/174.pdf, Accessed 08/09.

Pylons (2009): "PylonsHQ – Home". http://pylonshq.com/, Accessed 08/09.

Schut, P. ed. (2007, ed.): "OpenGIS® Web Processing Service", OGC document 05–007r7. Available at: http://portal.opengeospatial.org/files/?artifact_id=24151, Accessed 08/09. See also http://www.opengeospatial.org/standards/wps, Accessed 08/09.

SEE-GEO (2008): "See Geo Overview". http://edina.ac.uk/projects/seesaw/seegeo, Accessed 08/09.

Thain, D., T. Tannenbaum, and M. Livny, M. (2003): "Condor and the Grid". In: Berman, F.; Hey, A.J.G.; Fox G. (Eds.): Grid Computing: Making the Global Infrastructure a Reality, Wiley, 2003.

Woolf, A. (2006): "Wrappers, portlets, resource-orientation and OGC in Earth-System Science Grids", OGC Technical Committee meeting, Edinburgh. Available at: http://portal.opengeospatial.org/files/?artifact_id=15966, Accessed 08/09.

Woolf, A.; Woodcock, R. (2005): "Grid-enabling OGC: profiling against WSRF", Workshop on Grid Middleware and Geospatial Standards for Earth System Science Data, National e-Science Centre, Edinburgh, UK. Available at: http://epubs.cclrc.ac.uk/bitstream/1523/OGCWSRF.ppt, Accessed 08/09.

Hamburg University of Technology, Department of River and Coastal Engineering, University of Bonn, Department of Geography, Chair of Cartography

BENEFITS OF GRID COMPUTING FOR FLOOD MODELING IN SERVICE-ORIENTED SPATIAL DATA INFRASTRUCTURES

Stefan Kurzbach, Prof. Dr. Erik Pasche, Sandra Lanig, Prof. Dr. Alexander Zipf

Abstract: In 2007, the European Commission has passed the "Flood Directive" (2007/60/EG) dealing with the identification of inundated areas and the creation of flood risk maps. The basis for flood modeling is provided by computationally and storage-intensive flow simulations. Digital terrain data is the starting point for generating two-dimensional flow models. When discretizing the computational mesh for hydraulic simulation, digital terrain models with a resolution of one meter or less have to be subjected to several elaborate preprocessing steps. A number of simulation and modeling tools for this purpose have already been developed as "classical" SDI applications. However, building flood and risk models for a study area covering many square kilometers is not possible using common desktop GIS. The German GDI-Grid project (SDI-Grid, www.gdi-grid.de) extends regular OGC-based SDI services with grid computing capabilities. It focuses on WPS tools and an architecture for geoprocessing in the grid. At the example of flood modeling, this article shows which benefits can be generated in spatial data infrastructures by using grid technology.

Keywords: Grid computing, flood modeling, web service, workflow, spatial data infrastructure

// VORTEILE VON GRID COMPUTING FÜR DIE ÜBERSCHWEMMUNGS-MODELLIERUNG IN EINER SERVICEORIENTIERTEN GEODATENINFRASTRUKTUR

// Zusammenfassung: Die Europäische Kommission verabschiedete 2007 die Hochwasserschutzrichtlinie (2007/60/EG). Diese umfasst sowohl die Ausweisung von Überschwemmungsflächen als auch die Kartierung von Risikogebieten. Die Grundlage hierfür liefern rechen- und speicherintensive Strömungssimulationen. Digitale Geländedaten bilden den Ausgangspunkt für die Erstellung zweidimensionaler Strömungsmodelle. Bei der Diskretisierung des Berechnungsnetzes müssen Geländemodelle mit einer Auflösung von einem Meter oder weniger in mehreren Schritten aufwändig vorprozessiert werden. Zahlreiche Werkzeuge für Simulation und Modellierung wurden bereits als „klassische" GDI-Applikationen entwickelt. Für die Hochwasser- und Risikomodellierung eines viele Quadratkilometer umfassenden Untersuchungsgebiets reichen herkömmliche Desktop-GIS jedoch nicht aus. Im Rahmen des Projektes GDI-Grid (www.gdi-grid.de) erfolgt die Adaption herkömmlicher OGC-basierter GDI-Dienste für den Einsatz von Grid-Computing. Den Schwerpunkt bilden Werkzeuge und eine Architektur zur Geoprozessierung mit WPS im Grid. In diesem Beitrag wird anhand des Szenarios Hochwassermodellierung dargestellt, welcher Mehrwert in Geodateninfrastrukturen durch die Verwendung von Grid-Technologie generiert werden kann.

Schlüsselwörter: Grid Computing, Hochwassermodellierung, Webservice, Workflow, Geodateninfrastruktur

Authors

Dipl.-Inf. Stefan Kurzbach
Prof. Dr.-Ing. Erik Pasche
Hamburg University of Technology
Department of River and Coastal Engineering
Denickestraße 22
D-21073 Hamburg
E: {stefan.kurzbach, pasche}@tu-harburg.de

Dipl.-Geoinf. Sandra Lanig
Prof. Dr. Alexander Zipf
University of Bonn
Department of Geography, Institute of Cartography
Meckenheimer Allee 172
D-53115 Bonn
E: {lanig, zipf}@geographie.uni-bonn.de

1. MOTIVATION

Recent history in Europe has been shadowed by numerous flood disasters and their devastating consequences for the environment, economy, and citizens. Climatologists anticipate even more frequent and extreme precipitation events leading to extreme floods (Barredo 2007). In 2007 the European Commission acted on this issue and passed the "Flood Directive". Its scope is the evaluation and management of flood risk in all European countries. National actions are required in three steps (European Commission 2007):

1 preliminary flood risk assessment (until 2011)
2 generation of flood hazard and flood risk maps for all flood-prone river and coastal zones (until 2013)
3 preparation of flood risk management plans (until 2015).

According to the Flood Directive, flood hazard maps must display inundated areas, water depths, and flow velocities for statistical flood events of medium probability, meaning a water level or discharge that is expected to occur about every 100 years on average, as well as extreme floods, and events with lower recurrence periods. A combination of numerical simulation models and GIS has to be applied to fulfill these requirements, but the number of models to be created puts enormous pressure on the national authorities.

Simple inundation maps can be created, for instance, by extrapolation of a critical water level onto coastal areas and forelands. This can merely give a static view of the flooded areas, however, and could only be useful in the preliminary assessment step. Flood hazard maps, on the other hand, including varying water depths and flow velocities, are typically based on multi-dimensional, time-dependent flow models (also called hydraulic or hydrodynamic models) that take into account the various parameters affecting the flow situation, such as surface topography and roughness. In practice mostly one-dimensional and depth-averaged two-dimensional models are used because fully three-dimensional models have high computational requirements, and because the vertical flow component only plays a minor part in river flow (Pasche 2007).

Digital elevation models (DEM) are the main data source for flow model topography. DEMs are now readily available with a resolution of 1 meter or less. Model creation, in particular two-dimensional discretization of the flow network, is a time- and storage-consuming process and is usually carried out by consulting engineers on behalf of the national authorities (Rath 2007). A typical desktop computer is not capable of handling the data of more than a few square kilometers at a time, and it takes hours to complete a discretization process. In particular the use of high-resolution topographic data across large areas and the evaluation of the detailed simulation results creates a need for sophisticated processing techniques and storage management.

Grid computing is a technology that allows many distributed computers to collaboratively solve a single problem (Foster and Kesselman 1999). Foster has proposed a three-point checklist defining the properties of a grid. According to this list a grid "coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of services" (Foster 2002). A grid may provide the required computational power and storage capacities for flood simulations at low cost and on demand. In this article we focus on the German D-Grid infrastructure. The application of flood modeling is investigated in the research project GDI-Grid to implement geoprocessing services within the D-Grid infrastructure using Globus Toolkit 4 and standards of the Open Geospatial Consortium. Most of the users of computing grids come from academic institutions or are associated industry partners in research projects. Not only is this due to the fact that only participants of approved projects can get access to the national grid infrastructures, but also that the access hurdle of using grid technology is very high. Solely academic institutions have the required technical know-how to overcome this barrier and to profit from the power of grid computing. Private users and small companies like consulting engineers cannot easily gain this benefit.

To overcome the problem of access to the grid and to provide the available computing resources to flood modelers we suggest the following actions:

- Supply standards-based geoprocessing services for tasks related to flood modeling,
- enable these services to utilize the power of a grid as a back-end computing and storage environment while hiding the grid's complexity, and
- integrate these flood modeling services into a service-oriented spatial data infrastructure (SDI).

2. STATE OF THE ART

We present existing practices related to service-oriented SDI and geoprocessing in grid computing environments. Integrating domain-specific services into a SDI and grid-enabling geospatial services is not limited to the field of flood modeling.

2.1 SPATIAL DATA INFRASTRUCTURES

A SDI provides access to globally distributed spatial data through standard, interoperable services in a service-oriented architecture (SOA). As a leading organization for voluntary consensus standardization the Open Geospatial Consortium (OGC) has published a number of open standards suitable for building SDIs in collaboration with the ISO/TC 211 (ISO Technical Committee 211 – Geographic Information / Geomatics). Previous efforts of the OGC have primarily been based on discovery, access, and visualization of geospatial data. However, according to Nebert (2004), a fundamental element of future SDI will be the integration of geoprocessing services, that is, processing functions that work on spatially related data. Geoprocessing services have only recently been considered in the OGC by issuing the Web Processing Service (WPS) standard. The WPS offers any processing functionality through a web-based interface via three mandatory operations. These service operations are getCapabilities for a brief service description, describeProcess returning a detailed description of selected processes, and execute for running a process (Schut 2007).

Existing geoprocessing routines, such as standard spatial algorithms (e. g. buffer, intersection, and generalization operations), can easily be wrapped as web services. Since OGC's publication of the WPS standard many reference implementations and case studies have been done. Kiehle, Greve and Heier (2007) discuss the potential of extending SDIs with geoprocessing services and state that a "generic web service architecture for providing common geoprocessing capabilities" must be established using OGC and well-known web standards.

WPS geoprocessing tasks have been implemented in several other spatial research domains e. g. for precision farming (Nash et al. 2008), simplification (Foerster and Schäffer 2007), hydrological applications (Diaz et al. 2008), biogeography (Graul and Zipf 2008), forest fire (Friis-Christensen et al. 2007), housing marketing analysis and disaster management (Stollberg and Zipf 2007, 2008), urban waste land determination and land management (Lanig et al. 2009), and terrain processing (Lanig et al. 2008, 2009). Some basic calculations like buffering are described in (Heier and Kiehle 2005). A range of processes have been implemented by the cartography research group of the University of Bonn, Germany, and have been made available at http://www.opengeoprocessing.org.

2.2 GEOPROCESSING IN GRID COMPUTING ENVIRONMENTS

A grid "coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces, and delivering nontrivial qualities of service" (Foster 2002). Grid computing infrastructures use grid middlewares for accessing and managing distributed computing and data storage resources, and to provide security mechanisms. There exist several grid middlewares. The currently most utilized and adopted middlewares are Globus Toolkit (Foster 2005), UNICORE (Uniform Interface to Computing Resources) (Streit 2009), LCG/gLite (http://www.glite.org) and dCache (Fuhrmann 2004).

In 2008, the OGC and the Open Grid Forum (OGF), an organization dedicated to the development of standards for the management of distributed computing resources as required for grid computing, have agreed to work together on harmonizing standards for geoprocessing in the grid. They have signed a memorandum of understanding concerning future collaboration (Lee and Percivall 2008). Grid-enabling geospatial processes has already been evaluated in several fields of study. Research in earth sciences strives for providing services that process sensor observations for wildfire applications as part of the GEOSS architecture (Mandl et al. 2007, Lee 2008). The CrossGrid project investigated the use of grid computing for flood forecasting (Hluchý et al. 2004).

Geoprocessing workflows and a grid processing profile for WPS are part of the OGC Web Services (OWS) Interoperability Testbed, phase 6 (OWS-6). Within the OWS-6 Baranski et al. (2009) and Schäffer and Schade (2009) deal with the chaining of geospatial processes and give guidelines for developing WPS with access to a grid computing environment. Liping Di et al. (2003, 2008), Baranski (2008), and Padberg and Kiehle (2009) give general ideas about linking grid technology and OWS. One important aspect is to overcome differences in service communication between OWS and generally SOAP- and WSDL-based grid services. Hobona, Fairbairn et al. (2007, 2009) have developed a workflow management system (Semantically-Aware Workflow Engines for Geospatial Web Service Orchestration, SAW-GEO) supporting the orchestration of grid-enabled geospatial web services.

Some research has been done on providing simulation models as geoprocessing services in a SOA. Floros and Cotronis (2006) have developed the "Service-Oriented Simulations" framework (ServOSims) aiming at composing and executing scientific simulations as stateful web services. In their model, service orchestration is based on data-centric notifications between service instances, but OGC-compliant services are not considered. Gregersen, Gijsbers and Westen (2007) designed the "Open Modeling Interface" (OpenMI) for easy definition and linking of processes in the hydrological domain. However, this approach is not based on standard web service technology, so it does not strictly fit into the SOA paradigm. The GEOSS (Global Earth Observation System of Systems) Architecture Implementation Pilot (AIP, http://www.ogcnetwork.net/AIpilot), which is part of the OGC Network, develops and deploys new process and infrastructure components for the GEOSS Common Infrastructure (GCI) and the broader GEOSS architecture based on OGC specifications.

2.3 RELATED WORK

Within the GDI-Grid project a number of WPS for the processing of digital elevation models have been developed based on the deegree framework (Fitzke et al. 2004) and have been extended for geoprocessing in a grid computing infrastructure based on the grid middleware Globus Toolkit 4 (Padberg and Kiehle 2009). Lanig et al. (2008) have shown how massive terrain data can be processed in grid computing environments based on OGC standards. We have developed a 3D Terrain Discretization Grid Service (Gaja3D) and evaluated the efficiency of this new technology at the creation of a large-scale two-dimensional flow model for the estuary of the river Elbe (Germany) from several million measured elevation points (Kurzbach and Pasche 2009). The technology presented below is based on the results and experiences of this work. We apply the WPS standard and Globus Toolkit to implement a flood modeling architecture suited for integration in a SDI that is using the German D-Grid infrastructure.

3. FLOOD MODELING BY HYDRODYNAMIC SIMULATION

A flow model represents the motion of water, e. g. in pipe networks, rivers, open channels or oceans. The common basis for all flow models is the numerical solution of the Navier-Stokes equations, a set of equations that describe the motion of fluids (Malcherek 2001). For free surface flow, as it occurs in such moving water bodies as rivers, estuaries, and oceans, two-dimensional depth-averaged models are preferred over fully three-dimensional models. This simplification results in a set of equations called shallow water equations needing to be solved by numerical methods. In some situations the flow process can be further reduced to a one-dimensional model. In order to save computation time, a combination of one- and two-dimensional (coupled) models can be applied (Schrage et al. 2009). The output of a numerical model includes time-series of variables like water depth, flow velocity, temperature, salinity, and bed load.

The numerical solution of a hydrodynamic model is based on a discretization of the surface topography and other properties affecting the flow situation, like surface roughness, vegetation, hydraulic structures (e. g. dikes, weirs, and bridges), as well as wind and waves. A two-dimensional discretization consists of a network of nodes and elements and is either a structured, regular grid or an unstructured mesh. It is usually created based on a digital elevation model of the topography and the bathymetry of a study area and has to incorporate characteristics of the terrain that are vital to the simulation (Rath 2007).

High-resolution topographic data for flood plains is nowadays gained using remote sensing methods (e. g. LiDAR). Initially, the measured points contain measurement errors, vegetation, and man-made structures. These have to be filtered prior to use. After filtering, the points are triangulated to form a continuous surface model (Triangulated Irregular Network, TIN). This TIN can, however, not directly serve as input for a hydrodynamic simulation because the number of points is much to high. In order to make high quality DEMs manageable for hydrodynamic simulation it is necessary to generalize and to simplify the underlying terrain model while preserving critical terrain features (Rath 2007).

Several algorithms are available for generating multi resolution DEMs at different levels of detail (LODs). Lanig et al. (2008, 2009) have implemented algorithms based on the research work by Garland and Heckbert (1997) as a geoprocessing service. This 3D Terrain Generalization WPS processes multi-scale DEMs in predefined LODs. The surface geometry is stored as a TIN, and the algorithm is based on an iterative generalization of edge aggregation by vertex pair contraction. The error approximation for simplification of each vertex is the sum of squared distances to the planes. This algorithm cannot be applied, however, for flow model simplification, but is rather suited for display purposes (e. g. reduction of the number of triangles for different levels of detail depending on viewer distance).

Flow models have to fulfill a number of criteria for hydrodynamic simulation. Most importantly, structural features of the terrain have to be enforced as edges in the discretization network. Other requirements may restrict the element sizes and internal angles. Structural features (e. g. breaklines or contour lines) can be derived from the DEM or can originate from external data sources. Detection of structural features is often based on a regular, rasterized version of the DEM using image processing methods (Rath 2007). This raster DEM can be interpolated from the terrain in TIN format with a resolution appropriate for the detection process.

Applying line generalization methods to the detected structural lines reduces the number of points in the resulting flow model. Based on the Douglas-Peucker algorithm Lanig et al. (2008) have implemented a 3D Line Simplification WPS. When enough structural information and a model boundary have been gathered, a constrained Delaunay triangulation is performed on the lines. Elevations in the resulting TIN are interpolated from the original DEM or from the simplified contour lines. Simulations have shown that this strategy is well-suited for flow model creation.

The geoprocessing workflow for flood modeling is depicted in Figure 1. It focuses on flow model creation. Starting with a DEM, all necessary steps for flow model creation can principally be performed automatically. Tiling the input DEM makes it possible to execute the raster creation, breakline detection and generalization tasks in parallel for different subsets of the data (denoted by three parallel arrows).

Succeeding the model creation process is the calibration of the hydrodynamic model. This means performing a possibly very large number of simulations with varying flow parameters so that the model can correctly represent one or more previously observed flow situations. Only if the calibration process has been finished successfully, the model can be used to predict the consequences of a flood event. Simulations provide the water level and flow velocity results for creation of inundation maps. The inundated areas are derived by intersection of the water levels with the original DEM. A subsequent flood risk analysis integrates vulnerability information for the flooded areas to derive a flood hazard map.

4. GRID-ENABLING SIMULATION SERVICES

Services for flow simulation and flood modeling require and produce a large volume of data. As shown above they are also based on multiple resource-intensive processing steps, which are nowadays often executed on a single computer limiting the size of flow models, blocking the computer for the time of a simulation, and cluttering the local hard drives with heaps of simulation results. Grids deliver computational power and storage capacities on demand and without the administrative effort of local computing systems. Making use of grid computing for geoprocessing and simulation tasks is thus a logical consequence. However, for adoption in an SDI the geoprocessing services should conform to the WPS standard. Many differences between OGC and grid standards concerning service discovery, description, messaging, and security methods lead to interoperability problems between OWS and grid services. Grid services based on the WSRF are described by the Web Service Description Language (WSDL) and communicate by means of Simple Object Access Protocol (SOAP), both standards of the W3C. OGC web services, on the other hand, are following a restful service style that is in conflict with message style services like WSRF services.

In contrast to widely available and simple spatial algorithms, the majority of today's simulation models are trusted, well-tested legacy applications written in a classical scientific programming language like Fortran. Examples for two-dimensional hydraulic models include Resource Management Associates' RMA2 and RMA10 (http://www.rmanet.com), free RMA•KALYPSO developed at Hamburg University of Technology, Department of River and Coastal Engineering (http://www.tuhh.de/wb), Mike 21 by DHI (http://www.dhigroup.com), Delft3D by Deltares (http://www.deltares.nl), and others. Inputs and outputs are usually file-based, and processing is monolithic, which makes the models hard to be integrated with new technologies or to be coupled with other simulations.

Simulation services have to be grid-enabled in order to be used in the grid. Grid-enabling a part of software has become known under the term "gridification" (Lee and Percivall 2008). Aspects of gridification are making use of grid computing standards like the Web Services Resource Framework (WSRF) to develop stateful grid services, and to submit computationally intensive tasks into a computing cluster, e. g. by means of a Globus WS-Gram job submission service. Only recently there have been efforts to provide SOAP/WSDL interfaces for OWS as parts of the standards. As described in the previous section, simulation services shall be implemented as OWS, so gridification includes harmonizing or adapting the interface to the WSRF. The WPS specification has some potential to be extended with a WSRF interface thereby gaining additional capabilities. Dorka (2009) deals with the advantages of using WSRF for WPS. In the OWS-6 Grid Processing Profile engineering report (Baranski et al. 2009) we have presented our results concerning gridification of the WPS by means of the WSRF. For example, a stateful service controls and manages the submitted job and stores references to the results, which the user can later retrieve. Current developments around the WPS show that there is a need for maintaining the state of a geospatial process (Schäffer 2008). WSRF grid services provide similar functionality and the concepts to implement a stateful WPS using the WS-Resource and WS-ResourceProperties standards. Adhering to the WSRF has the additional effect that WPS developers get security "for free".

Security is a major requirement in many grid computing environments. Grid Security Infrastructure (GSI) is a specification for ensuring privacy, integrity, and delegation of privileges for communication between grid services and the user. It is used in grid middlewares like Globus Toolkit, LCG/gLite, and UNICORE. Gridification of OWS has to solve the security problem as many grid services rely on GSI. A problem is that no OWS standards support security methods like authorization and authentication in the grid. A number of possible solutions have been discussed, for instance, retrieving a stored GSI proxy certificate from a MyProxy server based on username and password credentials for a client (Padberg and Kiehle 2009, Liping Di et al. 2008). In 2007 the OGC Geo Rights Management Working Group (GeoRM, formerly GeoDRM) has issued an abstract model for rights and access management of geospatial resources (Vowles 2006). This model lacks a technical integration with W3C standards like the WS-Security specification, but a new OGC initiative strives to develop standard ways of performing web service authentication using these existing mechanisms while, at the same time, conforming to OWS standards (press release of August 4, 2009). Former security-related activities in the local German SDI North Rhine-Westphalia (GDI NRW) have resulted in the specification of the Web Authentication Service (WAS) and Web Security Service (WSS) in 2003. WAS and WSS are currently only applied in this context and have not yet been approved by the OGC.

Another aspect of gridification is that standards-based asynchronous notification mechanisms are yet missing in WPS. When a user has submitted a flow model to a flood simulation service or, likewise, started a long-running geoprocessing workflow, it is not feasible for him to wait for the results blocking his computer. For extremely large models the simulation may run for many hours if not days. The simulation service must be able to execute asynchronously and to deliver results

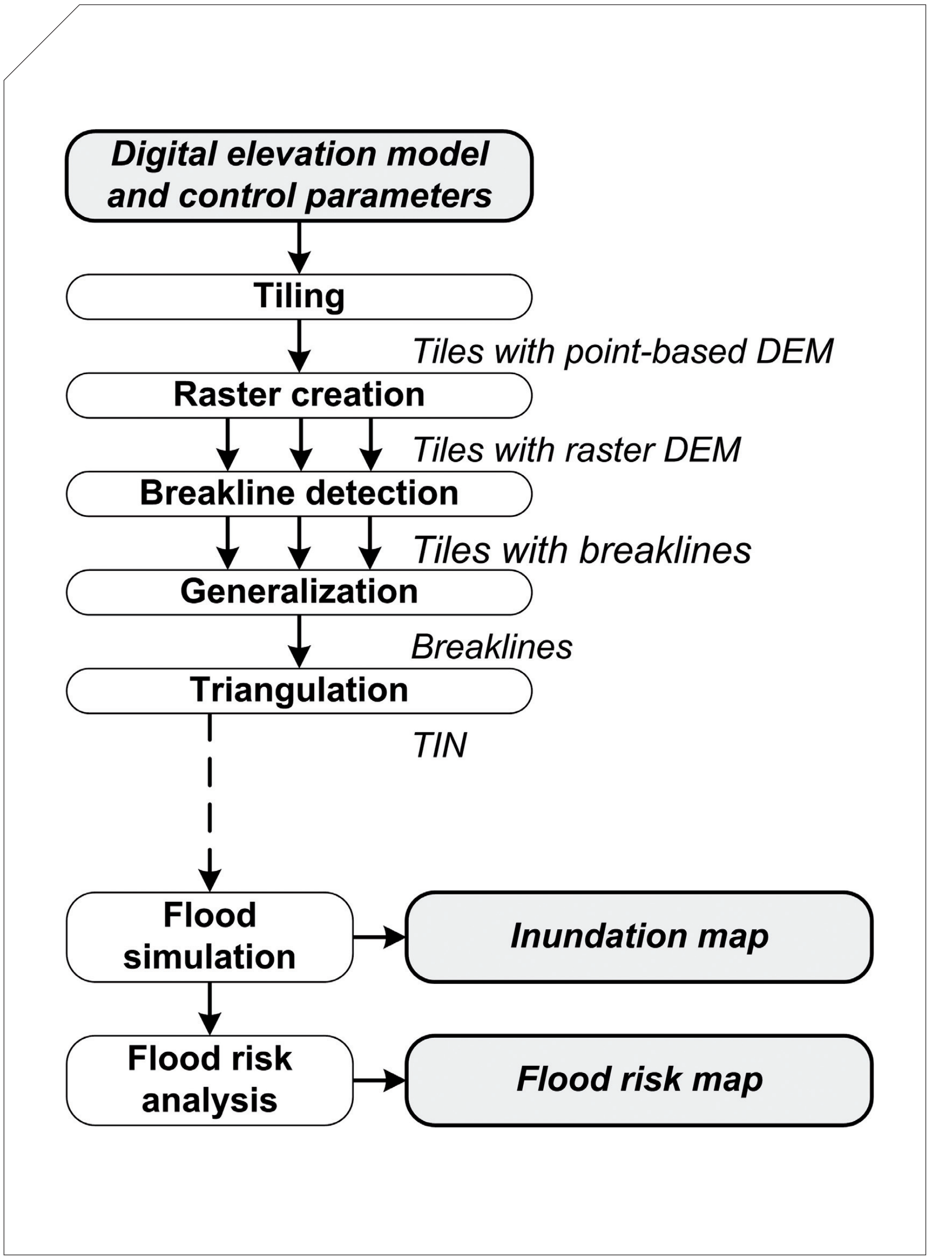


Figure 1: Geoprocessing workflow for flood modeling (focus on flow model creation)

when requested. The WPS standard supports this feature, but the client needs to poll for results. Much more convenient, and "clean" from a programmer's perspective, is an asynchronous notification using the WS-Notification standard, which is also part of the WSRF specification. For this purpose the OGC has issued the Web Notification Service standard, which is similar to WS-Notification, but uses other protocols. This makes it necessary to work on harmonizing the two standards.

5. 5 BENEFITS OF GRID COMPUTING FOR FLOOD MODELING IN A SDI

Flood simulation models are an interesting candidate for geoprocessing in an SDI. The current need for many large-scale flood simulations in Europe could be fulfilled by national flood modeling services. As the models for national rivers and potentially flooded areas are mostly non-existent, there is a need for services helping flow model creation. These could be used by engineering companies for building up the necessary models. A predefined geoprocessing workflow for flood modeling as shown in Figure 1 would further simplify the process significantly. A natural precondition is the availability of digital elevation models and other terrain data in the SDI.

By creation of geoprocessing services for legacy simulation models the functionality can be made available to a larger audience. The integration into a SDI and the specification of a standard service interface enables developers to realize an added value. There are many benefits in using grid technology for flood simulation. The most important ones from a user's point-of-view are listed below. They provide the starting point to set the requirements for our flood simulation service architecture:

- Processing on a remote machine leaving the user's computer free for other tasks,
- creation of larger flood models,
- parallel simulation of flow models,
- processing of massive terrain data,
- result management in the grid,
- keeping data confidential, securing it from unprivileged access, and
- automated execution of complete geoprocessing workflows for flood modeling.

Users as well as service developers benefit from grid technology. The existence of grid standards and their implementations makes it easier to write and to maintain better software. The existing grid middleware Globus Toolkit 4 (GT4) presents a reference implementation of the WSRF including GSI. This forms a solid base for developing standards-conforming grid services for geoprocessing and simulation as well as submission of jobs into a computing cluster. Our architecture has been designed to fulfill the mentioned requirements, but a complete practical evaluation is yet open. Nevertheless, we have quantified the efficiency of a grid service for terrain processing in (Kurzbach and Pasche 2009). The results show that a terrain dis-cretization process of the river Elbe estuary, which would take hours on a regular desktop computer, can be performed in less than 20 minutes when executed on a computing cluster. The input DEM has been partitioned into 67 tiles and separate grid jobs were run for each processing step. Input data, intermediate and final results have summed up to about 3 GB. A complete model of the river Elbe from the city of Hamburg to the estuary would be 7–8 times this size.

The grid consists of a multitude of processing and data resources that are either part of a computing cluster or single computers. A flood simulation service based on a parallel calculation core can make use of several resources for a single simulation by application of memory-parallel (OpenMP) or message-passing (MPI) communication mechanisms. Many flow models are already capable of parallel execution. However, they all lack the possibility to scale in a WSRF-based grid. Our efforts are to parallelize a flow model while respecting grid standards. We are currently developing a Flood Simulation Service that can be executed on an arbitrary number of grid nodes using standardized grid service communication and a WPS front-end for the user. Domain decomposition techniques are applied to exchange inner boundary conditions of connected model parts. Boundaries are iteratively improved to converge to a global solution. The Flood Simulation Service will be evaluated at a partitioned Elbe model that is created using the existing terrain discretization methods.

6. FLOOD SIMULATION SERVICE ARCHITECTURE

As part of our work in the GDI-Grid project, we have implemented different terrain processing services based on the WPS specification for different surface generalization functionalities. In a second step, we have extended the WPS interface using the GT4 middleware so the processes can be seamlessly integrated into the grid. The services are implemented as grid services either with GSI through MyProxy credentials or, in case of the Flood Simulation Service, as a WPS with a WSRF-conforming interface. Additional GT4 services include the 3D Line Simplification and Terrain Generalization WPS.

The geoprocessing grid services are then orchestrated using a formal workflow description (Business Process Execution Language, BPEL) and a workflow engine capable of automatically executing the workflow in the grid (Fleuren and Müller 2008). The workflow engine contacts the Flood Simulation Service via the WSRF interface using SOAP messaging. Each grid service execution results in a job being submitted to a GT4 WS-Gram service. This enables us to control an arbitrary number of remote jobs on grid-based computing resources.

A major feature of our implementation is that no data transfers go through the workflow engine, but instead third-party transfers are initiated, and references to the results are handed over to the control of the workflow engine (see Figure 2). Data transfers are performed by a Data Access and Integration Service (OGSA-DAI) that efficiently gathers data from a large number of different file and data base sources for processing in the grid. For fast file transfers in the grid the GridFTP standard is used. WPS is the front-end interface to the GDI-Grid infrastructure and uses the grid as a back-end computing environment.

A major problem is staying OGC-compliant while, at the same time, supporting GSI and including legacy OWS into the workflow. Web Feature and Coverage Services (WFS, WCS) can now easily serve as data sources in the workflow. The OGSA-DAI server requests data from WFS and WCS, which in turn may retrieve data from a spatial data base outside the grid, and delivers the results directly to a location inside the grid. This ensures that subsequent access

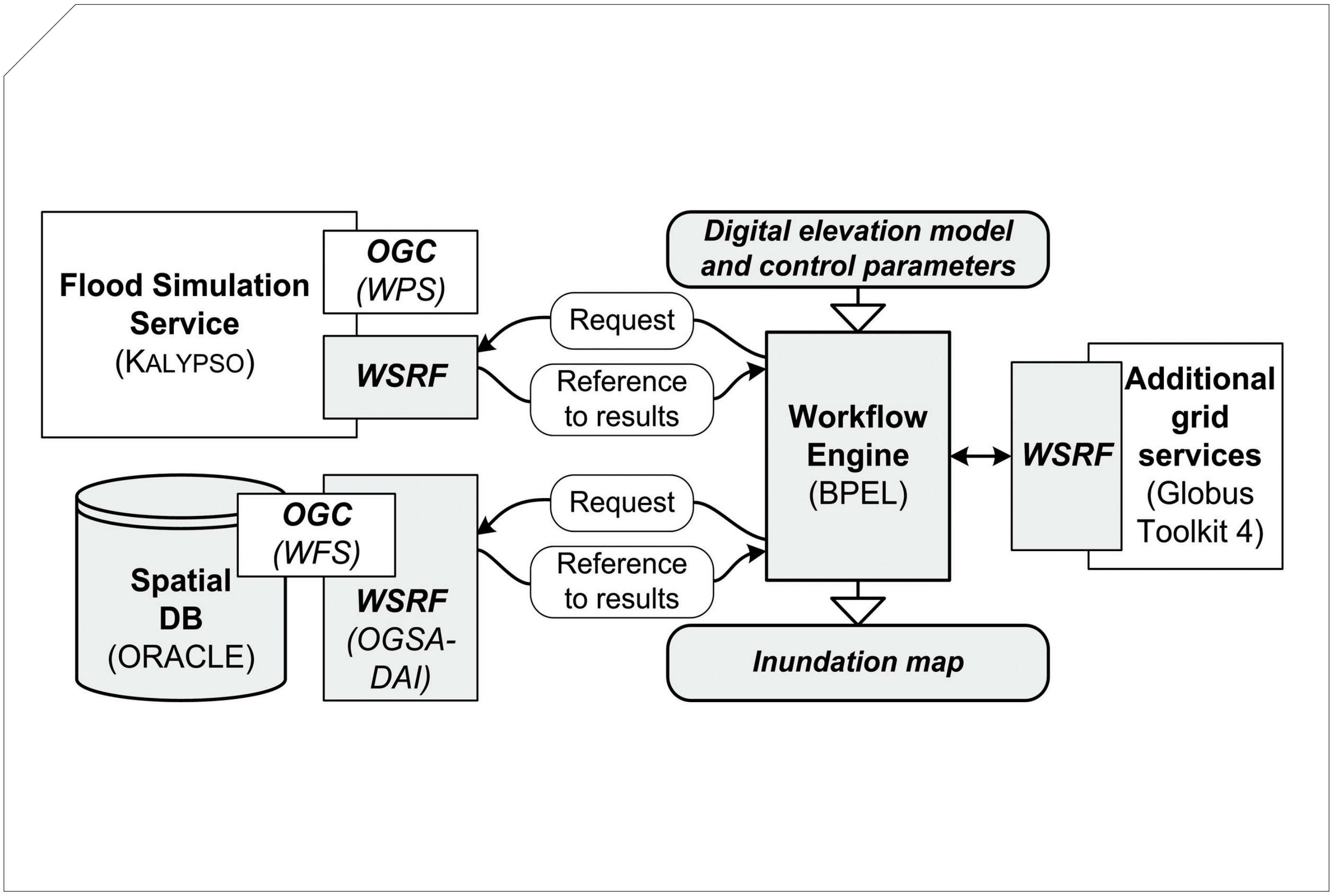


Figure 2: Distributed architecture for flood modeling service orchestration

to the data can be done efficiently based on GridFTP. Regular WS-Security mechanisms and delegation of proxy certificates to the OGSA-DAI WSRF-based service ensure that the data is kept confidential.

We have implemented a prototypical geoprocessing workflow for flow model creation in BPEL based on the workflow from Figure 1. It is executed on a Mage BPEL4Grid workflow engine with extensions for WSRF-based web services. The workflow includes retrieving data from an external WFS, processing a DEM with breakline detection and generalization as well as final TIN creation by Delaunay triangulation. Grid services have been developed with only open source software using GT4, the deegree framework, and the KALYPSO simulation platform (http://kalypso.sourceforge.net).

7. CONCLUSIONS AND FUTURE WORK

The need for computing power and storage capacity is steadily rising within the geo-community. In particular LiDAR data is being used to create high-resolution digital elevation models for flood modeling, but processing this terrain data means to work with millions of raw data points, and to run computationally intensive algorithms. In this article, we presented the possibility to enhance the processing of massive digital elevation data for flood modeling using standardized WPS and grid computing.

We also displayed how this technology can aid the creation of flow models in times of high need. The integration of grid-based geoprocessing services into a spatial data infrastructure is a logical next step. National SDIs could provide flood modeling services that help in realizing the Flood Directive, more precisely services for flow model creation, and generation of inundation and flood hazard maps. Modelers could then save time and money by using an existing grid infrastructure instead of buying expensive hardware to run their simulations. We have presented an architecture that uses WSRF-based grid services with a WPS front-end.

In future research, the management and provisioning of flood models in an SDI should be investigated. SDI and grid computing together with the appropriate tools can allow for collaborative modeling and flow model sharing. Model interfaces like the OpenMI would create the possibility to connect different flood models. If the interface is extended to create stateful WSRF-based OpenMI grid services, the shared models could then be run in the grid in a coupled fashion. This could be the future of flood modeling.

8. ACKNOWLEDGEMENTS

This work is part of the Spatial Data Infrastructure Grid project (GDI-Grid, http://www.gdi-grid.de) funded by the German Federal Ministry of Education and Research (BMBF). The main goal of the project is the efficient mining and processing of spatial data for the simulation of noise dispersion and disaster management. ◀

References

Baranski, B. (2008): Grid Computing Enabled Web Processing Service. GI-Days 2008, Münster, Germany.

Baranski, B.; Shaon, A.; Woolf, A.; Kurzbach, S. (2009): OWS-6 WPS Grid Processing Profile Engineering Report. OGC 09–041r1.

Barredo, J. (2007): Major flood disasters in Europe: 1950–2005. In: Natural Hazards, Vol. 1, pp. 125–148.

Di, L.; Chen, A.; Yang, W.; Liu, Y.; Wei, Y.; Mehrotra, P.; Hu, C.; Williams, D. (2008): The development of a geospatial data Grid by integrating OGC Web services with Globus-based Grid technology. In: Concurrency and Computation: Practice & Experience, 14, pp. 1617–1635.

Di, L.; Chen, A.; Yang, W.; Zhao, P. (2003): The Integration of Grid Technology with OGC Web Services (OWS): NWGISS for NASA EOS Data. GGF8 & HPDC12 2003, pp. 24-27. Science Press.

Díaz, L.; Granell, C.; Gould, M. (2008): Case Study: Geospatial Processing Services for Web-based Hydrological Applications. In: Sample, J.; Shaw, K.; Tu, S.; Abdelguerfi, M. (Edts.): Geospatial Services and Applications for the Internet, pp. 31–47. Springer Science+Business Media, New York, USA.

European Commission (2007): Directive 2007/60/EC of the European Parliament and of the Council of 23 October 2007 on the assessment and management of flood risks: Official Journal of the European Union, L 288/27.

Fleuren, T.; Müller, P. (2008): BPEL Workflows Combining Standard OGC Web Services and Grid-enabled OGC Web Services: 34th Euromicro Conference on Software Engineering and Advanced Applications. SEAA '08, Sept. 3–5, Parma, Italy, pp. 337–344. The Printing House.

Floros, E.; Cotronis, Y. (2006): ServOSims: A Service Oriented Framework for Composing and Executing Multidisciplinary Simulations. In: Kellenberger, P. (Edt.): e-Science 2006. Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, 4–6 December 2006, Amsterdam, Netherlands, pp. 66–66. Wiley-IEEE, Washington DC, USA.

Foerster, T.; Schäffer, B. (2007): A Client for Distributed Geo-processing on the Web: Web and Wireless Geographical Information Systems. Proceedings of the 7th International Symposium, W2GIS 2007, Cardiff, UK, November 28–29, 2007. Lecture Notes in Computer Science, Vol. 4857, pp. 252–263. Springer, Berlin, Heidelberg.

Foster, I. (2002): What is the Grid? – a three point checklist. In: GRIDtoday, 6.

Foster, I. (2005): Globus Toolkit Version 4: Software for Service-Oriented systems. In: Jin, H.; Reed, D.; Jiang, W. (Edts.): Network and Parallel Computing. Proceedings of the IFIP International Conference, NPC 2005, Beijing, China, November 30-December 3. Lecture Notes in Computer Science, 3779/2005, pp. 2–13. Springer, Berlin, Heidelberg.

Foster, I.; Kesselman, C. (Edts.) (1999): The Grid: Blueprint for a New Computing Infrastructure.

Friis-Christensen, A.; Ostländer, N.; Lutz, M.; Bernard, L. (2007): Designing service architectures for distributed geoprocessing: challenges and future directions. In: Transactions in GIS, 6, pp. 799–818.

Fuhrmann (2004): dCache, the Overview. White paper. http://www.dcache.org/manuals/dcache-whitepaper-light.pdf, accessed August 2009.

Garland, M.; Heckbert, P. (1997): Surface simplification using quadric error metrics: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 209–216.

Graul, C.; Zipf, A. (2008): Putting biogeographic research on the spatial web: Towards interoperable analysis tools for global change based on the web processing service (WPS). In: Huadong, G. (Edt.): Digital Earth Summit on Geoinformatics: Tools for Global Change Research, Vol. 1. Taylor & Francis.

Gregersen, J.; Gijsbers, P.; Westen, S. (2007): OpenMI: Open modelling interface. In: Journal of Hydroinformatics, 3, pp. 175–191.

Heier, C.; Kiehle, C. (2005): Geodatenverarbeitung im Internet-der OGC Web Processing Service. In: GIS, pp. 39–43.

Hluchý, L.; Tran, V.; Habala, O.; Šimo, B.; Gatial, E.; Astaloš, J.; Dobrucký, M. (2004): Flood Forecasting in CrossGrid Project. In: Dikaiakos, M. (Edt.): Grid Computing. Second European AcrossGrids Conference, AxGrids 2004, Nicosia, Cyprus, January 28–30, 2004. Revised Papers. Lecture Notes in Computer Science, Vol. 3165, pp. 51–60. Springer, Berlin, Heidelberg.

Hobona, G.; Fairbairn, D.; Hide, H.; James, P. (2009 [Im Druck]): Orchestration of Grid-Enabled Geospatial Web Services in Geoscientific Workflows. In: IEEE Transactions on Automation Science and Engineering, 99.

Hobona, G.; Fairbairn, D.; James, P. (2007): Workflow Enactment of Grid-Enabled Geospatial Web Services. Proceedings of the UK e-Science All Hands Meeting.

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

Kurzbach, S.; Braune, S.; Grimm, C.; Pasche, E. (2009): Hochwasser-Modellierung im Geodateninfrastruktur-Grid. In: Strobl, J.; Blaschke, T.; Griesebner, G. (Edts.): Angewandte Geoinformatik 2009: Beiträge zum 21. AGIT-Symposium Salzburg. July 7–10, Salzburg, Austria. Wichmann Verlag.

Kurzbach, S.; Pasche, E. (2009): A 3d Terrain Discretization Grid Service for Hydrodynamic Modeling: Proceedings of the 8th International Conference on Hydroinformatics (HEIC), Concepción, Chile.

Lanig, S.; Klärle, M.; Zipf, A. (2009): Offenes, nachhaltiges Landmanagement mittels standardisierter Web Processing Services (WPS). In: Strobl, J.; Blaschke, T.; Griesebner, G. (Edts.): Angewandte Geoinformatik 2009: Beiträge zum 21. AGIT-Symposium Salzburg. July 7–10, Salzburg, Austria. Wichmann Verlag.

Lanig, S.; Schilling, A.; Stollberg, B.; Zipf, A. (2008): Towards Standards-Based Processing of Digital Elevation Models for Grid Computing through Web Processing Service (WPS). Work ▷

shop on Information Systems and Information Technologies (ISIT 2008). In: Gervasi, O.; Laganà, A.; Murgante, B.; Taniar, D.; Mun, Y.; Gavrilova, M. (Edts.): Proceedings of the International Conference on Computational Science and its Applications. Perugia, Italy, June 30 – July 03. Lecture Notes in Computer Science, Vol. 5073, pp. 191–203. Springer, Berlin, Heidelberg.

Lanig, S.; Zipf, A. (2009): Towards generalization processes of LiDAR data based on GRID and OGC Web Processing Services (WPS): Geoinformatik 2009, Osnabrück, Germany.

Lee, C. (2008): Grid Technologies for GEOSS: The User and the GEOSS Architecture XXV. Perspectives on GEOSS Architecture: Principles and Implementation. Workshop on OGC TC Plenary, December 3–4th 2008, Palacio de Congresos, Valencia, Spain.

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

Malcherek, A. (2001): Hydromechanik der Fließgewässer. Bericht Nr. 61. Habilitation, Hannover.

Mandl, D.; Sohlberg, R.; Justice, C.; Ungar, S.; Ames, T.; Frye, S.; Chien, S.; Cappelaere, P.; Tran, D. (2007): Experiments with user centric GEOSS architectures. In: Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International, pp. 297–300.

Nash, E.; Korduan, P.; Bill, R. (2007): Optimising data flows in precision agriculture using open geospatial web services. In: Precision Agriculture, pp. 753–759.

Nebert (2004): Developing Spatial Data Infrastructures: The SDI Cookbook, Version 2.0.

Padberg, A.; Kiehle, C. (2009): Geodateninfrastrukturen & Grid-Computing: Aktuelle Herausforderungen und Anwendungsbeispiele. In: Reinhardt, W.; Krüger, A.; Ehlers, M. (Edts.): Geoinformatik 2009, pp. 129–138.

Pasche, E. (2007): Flood Modelling in Urban Rivers – the State-of-the-Art and Where to Go. In: Ashley, R.; Garvin, S.; Pasche, E.; Vassilopoulos, A.; Zevenbergen, C. (Edts.): Advances in Urban Flood Management, pp. 59–89. Taylor & Francis, Leiden, The Netherlands.

Rath, S. (2007): Model discretization in 2D hydroinformatics based on high resolution remote sensing data and the feasibility of automated model parameterisation. Dissertation. 1st ed. Hamburger Wasserbau-Schriften. Hamburg University of Technology, Hamburg, Germany.

Schäffer, B.; Schade, S. (2009): OWS-6 Geoprocessing Workflow Architecture Engineering Report. 07–143r1.

Schrage, N.; Antanaskovic, D.; Jung, T.; Pasche, E. (2009): KALYPSO – An Open Source Software Tool for Flood Studies in Rivers: Proceedings of the 8th International Conference on Hydroinformatics (HEIC), Concepción, Chile.

Schut, P.; Whiteside, A. (2007): OGC Web Processing Service (WPS) Specification. OGC 05–007r7.

Stollberg, B.; Zipf, A. (2007): OGC Web Processing Service Interface for Web Service Orchestration – Aggregating Geo-processing Services in a Bomb Threat Scenario: Web and Wireless Geographical Information Systems. Proceedings of the 7th International Symposium, W2GIS 2007, Cardiff, UK, November 28–29, 2007. Lecture Notes in Computer Science, Vol. 4857, pp. 239–251. Springer, Berlin, Heidelberg.

Stollberg, B.; Zipf, A. (2008): Geoprocessing Services for Spatial Decision Support in the Domain of Housing Market Analyses-Experiences from Applying the OGC Web Processing Service Interface in Practice. In: Friis-Christensen, A.; Pundt, H.; Compte, I. (Edts.): Proceedings of 11th AGILE International Conference on Geographic Information Science. Taking Geoinformation Science One Step Further. Girona, Spain, May 5–8.

Streit, A. (2009): UNICORE: Getting to the heart of Grid technologies. In: eStrategies – Projects, 9th editition, March 2009, pp. 8–9.

Vowles, G.; Herring, J.; Goodwin, C.; Scardina, M.; Matheus, A.; Opincaru, C.; Debeugny, W. (2006): OGC Geospatial Digital Rights Management Reference Model (GeoDRM RM). OGC 06–004r3.

International Institute for Geoinformation Science and Earth Observation (ITC)

ENHANCING CARTOGRAPHIC GENERALIZATION PROCESSING WITH GRID COMPUTING POWER AND BEYOND

Theodor Foerster, Dr. Jantien Stoter, Dr. Javier Morales

Abstract: Automated cartographic generalization requires a lot of processing power, due to its intrinsic complexity. Grid Computing is considered to be beneficial in this context. Grid Computing has specific requirements regarding the characteristics of deployed processes. This article validates the requirements of Grid Computing for the case of agent-based generalization. In particular, it describes how agent-based generalization can benefit from Grid Computing and describes a conceptual architecture. The architecture is demonstrated by the application of on-demand base maps for physical plans. Finally, an outlook is given which goes beyond the application of cartographic generalization and describes how automated generalization can support Web Service architectures in the future.

Keywords: Cartographic generalization, Grid Computing, agent-based generalization

// KARTOGRAPHISCHE GENERALISIERUNG UNTER NUTZUNG VON GRID COMPUTING

// Zusammenfassung: Automatische kartographische Generalisierung benötigt auf Grund ihrer hohen Komplexität viel Rechenleistung. Grid Computing wird in diesem Kontext als hilfreich angesehen, hat aber spezielle Anforderungen in Bezug auf den Generalisierungsprozess. Dieser Artikel validiert die Anforderungen von Grid Computing für den Fall von Agenten-basierter Generalisierung. Er beschreibt, wie Agenten-basierte Generalisierung von Grid Computing profitieren kann und beschreibt eine konzeptionelle Architektur. Die Architektur wird für den Fall der Erzeugung von on-demand Basiskarten für Raumpläne veranschaulicht. Abschließend wird ein Ausblick gegeben, der über die Anwendung von kartographischer Generalisierung hinausgeht, indem beschrieben wird, wie automatische Generalisierung zukünftig in Web Service Architekturen eingesetzt werden kann.

Schlüsselwörter: Kartographische Generalisierung, Grid Computing, Agenten-basierte Generalisierung

Authors

Theodor Foerster
Dr. Jantien Stoter
Dr. Javier Morales
International Institute for Geoinformation Science and Earth Observation (ITC)
Postbus 6
7500AA Enschede, the Netherlands
E: {foerster, stoter, morales}@itc.nl

1. INTRODUCTION

The generation of readable maps at a specific scale by means of automated generalization is a long standing challenge (Mackaness et al. 2007; Weibel, Dutton 1999). This so-called cartographic generalization is considered to be an optimization problem (Sester 2005) and requires a lot of processing power depending on the specific map requirements and on the specific configuration of objects available on the map. One of the optimization approaches towards automated generalization is the agent-based approach, which has yielded promising results (Lamy et al. 1999; Regnauld, Revell 2007). However, performance is still an unsolved issue, which becomes more urgent facing the challenge of producing readable maps for the web by the means of automated generalization.

With increasing network capabilities and processing power, distributed processing of data by means of Grid Computing has matured in the last years and is thereby promising to enhance performance of automated generalization processing. This was the starting point for designing an approach for integrating cartographic generalization processing and Grid Computing.

The paper claims that integrating the agent-based approach for automated generalization and Grid Computing is highly applicable, as the agent-based approach provides a valid means to split the processing of complete datasets into small tasks. Additionally, those tasks can run in parallel. Both aspects are crucial for integrating processes in Grid Computing (Jacob et al. 2005). The proposed integration is conceptually applied to a Web Service architecture for generating on-demand base maps for physical planning on the web (Foerster et al. 2008). A final proof of the proposed integration is still considered to be future work.

Section 2 will examine the characteristics of the agent-based approach for generalization. Section 3 will demonstrate how the agent-based approach matches the requirements of Grid Computing and will give an architecture overview. Section 4 applies the architecture to the application of on-demand base maps for physical plans. Section 5 presents applications beyond mapping, which also require generalization processing enhanced with Grid Computing power. Finally, the paper will discuss the approach and draw conclusions.

2. AGENT-BASED APPROACH FOR AUTOMATED GENERALIZATION

In an agent-based generalization process, an agent is attached to a single or group of objects on the map and is able to configure and perform generalization algorithms autonomously to satisfy its state according specific requirements. Those requirements (also known as generalization constraints) describe the conditions of the final map (e.g. the distance between two buildings should always be larger than 5 map units). The agent-based approach for automated generalization defines a hierarchy of three types of agents, to address the different types of requirements (Ruas 2000):

- Micro agent representing a single object
- Meso agent representing a group of objects
- Macro agent representing all objects available on the map display.

The hierarchy allows the generalization process to divide the map space into small partitions (e.g. urban block), which are then attached to agents. According to the hierarchy the agents on the upper level can influence the behavior of the agents on the lower level (Figure 1).

The agents perform the generalization according to a specific plan using a specific set of algorithms. The specific algorithms are configured and executed during the generalization process by each agent. According to the agent cycle the agent evaluates the result of the applied algorithm and reprocesses the algorithm with alternative parameter configurations until its state is satisfied.

3. ARCHITECTURE FOR A GRIDIFIED AGENT-BASED SYSTEM FOR AUTO MATED GENERALIZATION

The following aspects make an integration of Grid Computing and agent-based generalization highly applicable.

1 The agent model divides the generalization problem into small sub-problems by partitioning the map space. An agent is attached to each of the partitions. The agents configure generalization tasks, which have a small memory footprint.
2 Besides dividing a problem, it is also important to merge the result of each sub-problem again to one result. This is possible in the case of agent technology based on the agent's identity and the location each agent knows.
3 The generalization tasks can run in parallel, as they are configured as atomic and do not interfere with other tasks. From a Grid Computing perspective, both aspects are considered to be crucial to use the grid infrastructure efficiently.
4 As the generalization system runs multiple iterations to find the most applicable solutions, Grid Computing is highly beneficial for agent-based generalization processing.

The architecture for a gridified agent-based system for automated generalization is shown in Figure 2. Each agent creates a specific generalization task and submits it as a process job to the grid infrastructure. The created task consisting of process (executable code) and data (the parameters) is then handled by a grid node and the result is returned to the generalization system. According to the agent cycle the

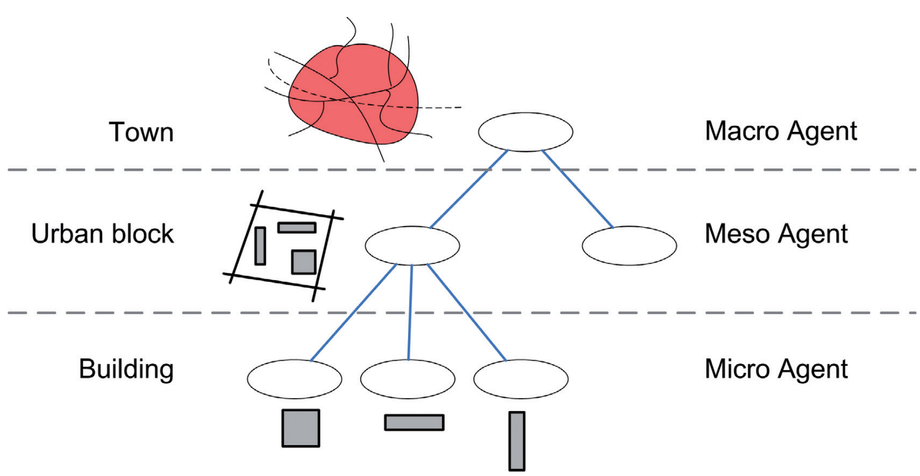


Figure 1: Hierarchy of agents (Ruas, 2000).

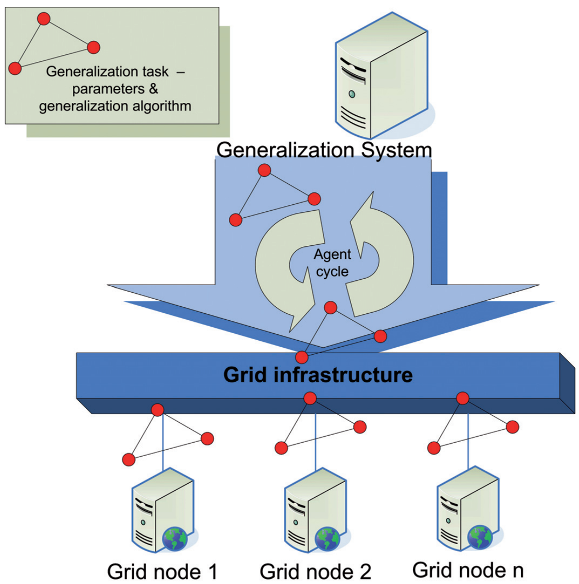


Figure 2: Architecture enabling Grid Computing access for agent-based generalization.

grid infrastructure is configured by many tasks at the same time and used iteratively until all the agents have reached the most satisfying state. During the execution of the generalization task on the grid, the agents are not able to communicate with each other. The configuration of the generalization tasks, the evaluation of the generalization result and the communication between the agents is implemented inside the generalization system.

4. AN ARCHITECTURE FOR WEB-BASED DISSEMINATION OF PHYSICAL PLANNING MAPS

To show its feasibility, the possible advances of an architecture for gridified agent-based generalization system are presented for a case study. The case study aims at disseminating physical plans on the Web. The plans are projected on base maps, which support the communication of the planning information to a user (Poppe, Foerster 2006). As shown in the map example (Figure 3), the granularity of objects in the base map does not match the scale of the overlaying physical plan nor does it match specific requirements of the user. In the future, the base maps will be generated from a single-scale topographic database according to specific user requirements by the means of automated generalization.

Foerster et al. (2008) describe a web-based architecture, which serves these on-demand base maps. The core of the architecture is the so-called generalization-enabled Web Map Service (WMS), which is enhanced with the agent-based generalization system. The WMS forwards the requests to the generalization system, which generates the base maps accordingly.

Because of different user requirements and the resulting processing effort to generate the base maps for each request accordingly, the application of Grid Computing is promising for the generalization-enabled WMS. The physical planning objects provide a partition of the base map. The physical planning objects define formal boundaries and are thereby applicable partitions for the base map. These partitions are used as topological constraints for the base map. In fact the generalization process has to preserve the topological relationship between the base map object and the physical plan. These partitions of the base map are used to set up the agent hierarchy consisting of meso agents (defined by the extent of the physical planning object) and the micro agents (defined by the base map objects). In this context a generalization task consists of meso agents representing the physical plan and the underlying base map objects represented by micro agents. This task is then submitted to the Grid infrastructure (as explained in Section 3).

First practical experiments regarding the generalization of the base maps have underlined the demand for such integration. Processing of hundred complex building stemming from a large-scale database (scale 1:1000) takes 20–30 seconds (2 CPUs @ 2.13 GHz and 2 GB of RAM). The process involved aggregation and simplification of the base map objects, regarding the available map space and the topological relations of the buildings with the overlaying physical plan object. Currently, during the iterations of the agent cycle two algorithms are applied (aggregation and simplification). It is important to note that these experiments have not been carried out on the Grid yet, but stress the demand to improve the performance also for more complex scenarios (involving more data and more generalization algorithms).

5. BEYOND CARTOGRAPHIC GENERALIZATION

Despite applying cartographic generalization for mapping at various scales, generalization becomes relevant for Web Service architectures in the future. Characteristics of generalization, which become important for such Web Service architectures requiring real-time and on-demand data access, are the ability to reduce the amount of data for network transfer and to change the granularity of data regarding a specific data model. Changing the granularity of data by automated generalization for data harmonization purposes is especially relevant for applications related to Spatial Data Infrastructures (SDIs). Data harmonization and data reduction are crucial aspects and require massive processing capabilities (Williamson et al. 2003). This need for massive processing power is met by Grid Computing.

In particular, data harmonization is applied to integrate data from different countries and captured at different scales into a common data model. This is achieved by attribute names renaming, coordinate transformation, but also by adjusting the granularity of complete datasets. The first two aspects are carried out by so-called schema transformation (Lehto 2007). The latter aspect is addressed by generalization. Recently, Foerster et al. (2009) investigated the combination of these two types of processes for a data harmonization use case in the context of the Finish SDI. Due to the complexity of the two types processes and the large amount of data to be processed, the application of Grid Computing can improve the performance of the overall process significantly.

Generalization also plays an important role for data reduction, as generalization limits the amount of data by preserving the relevant information. The data is available in SDIs but needs to be sent through networks with limited bandwidth, especially in case of mobile applications. Therefore data reducti-

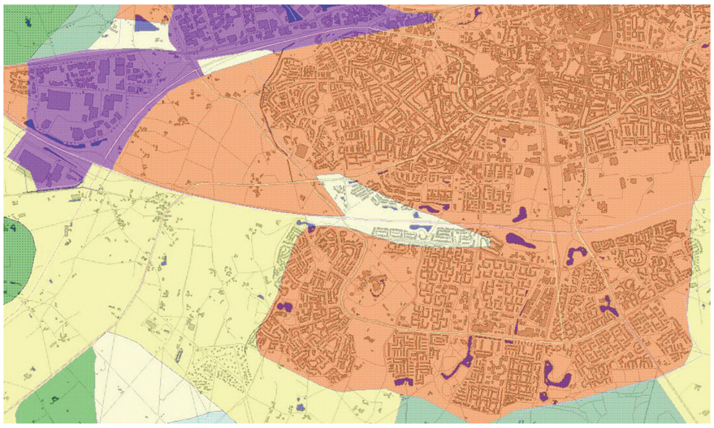


Figure 3: Example provincial plan with a non-generalized base map (original scale 1:10K) at a scale of 1:25K.

on as achieved by generalization can realize such a mobile application. This mobile application has constraints regarding the network, but also regarding the processing capabilities and battery life time of the mobile device. Reduced data limits the processing effort of the mobile device and thereby improves the performance and reduce the power consumption. Also related to the issue of reduced data for limited bandwidth, is the application of progressive transfer (van Oosterom, 2005). This approach also uses generalization to generate a data structure, which can be sent progressively over the network. The client first gets the most important details of the data, on which he/she can already start working, while the application retrieves the rest of the data concurrently. This reduces the latency of the application and improves its performance and usability significantly.

The presented examples show, that generalization processing is not only relevant for mapping, but also for database-centered and mobile applications, which benefit from generalization processing. However, to enhance the performance of these generalization processes and to meet the application requirements, Grid Computing is promising.

6. CONCLUSION

The integration of Grid Computing and agent-based generalization is highly promising to serve on-demand maps on the web. This article shows how agent-based generalization can be integrated into Grid Computing. For this study a scenario of generating on-demand base maps for physical plans is presented. Initial studies on the performance of this complex scenario have stressed the demand for integrating Grid Computing into agent-based generalization processing.

Besides the application of mapping, generalization is also required for SDI-related applications. To meet the performance requirements of these applications, generalization needs to be enhanced with Grid Computing power. ◀

References

Foerster, T.; Lehto, L.; Sarajakoski, L. T.; et al. (2009): „Map Generalization and Schema Transformation of Geospatial Data Combined in a Web Service". In: Computers, Environment and Urban Systems. accepted , doi:10.1016/j.compenvursys.2009.06.003.

Foerster, T.; Stoter, J. E.; Lemmens, R.; Ruas, A.; Gold, C. (eds.) (2008): „An interoperable Web Service Architecture to Provide Base Maps Empowered by Automated Generalization". In: Headway in Spatial Data Handling. Montpellier, France: Springer (Lecture Notes in Geoinformation and Cartography), pp. 255–76. — ISBN: 978–3–540–68565–4.

Jacob, B.; Brown, M.; Fukui, K.; et al. (2005): Introduction to Grid Computing. IBM (IBM Redbooks). — ISBN: 0738494003.

Lamy, S.; Ruas, A.; Demazeu, Y.; et al. (1999): „The Application of Agents in Automated Map Generalization". In: 19th International Cartographic Conference. Ottawa, Canada.

Lehto, L.; Wachowicz, M.; Bodum, L. (eds.) (2007): „Schema Translations in a Web Service Based SDI". In: 10th Agile International Conference on Geographic Information Science 2007. Aalborg University, Denmark.

Mackaness, W. A.; Ruas, A.; Sarjakoski, L. T. (eds.) (2007): Generalisation of geographic information: cartographic modelling and applications. Elsevier.

van Oosterom, P. (2005): „Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest". In: Cartography and Geographic Information Science. 32 (4), pp. 331–346.

Poppe, E.; Foerster, T. (2006): „Automated application-driven generalization of base maps for DURP". In: Proceedings geo-innovatie 2006 : 4e symposium. Ede, the Netherlands: Ruimte voor Geo – Informatie (RGI) pp. 85–87.

Regnauld, N.; Revell, P. (2007): „Automatic Amalgamation of Buildings for Producing Ordnance Survey 1:50 000 Scale Maps". In: The Cartographic Journal. 44 (3), pp. 239–250, doi: 10.1179/000870407X241782.

Ruas, A. (2000): „The Roles of Meso Objects for Generalization". In: Proceedings of the 9th International Symposium on Spatial Data Handling. Beijing, China pp. 3b50–3b63.

Sester, M. (2005): „Optimization approaches for generalization and data abstraction". In: International Journal Of Geographical Information Science. 19 (8–9), pp. 871–897.

Weibel, R.; Dutton, G.; Longley, P.; Goodchild, M.; Maguire, D.; et al. (eds.) (1999): „Generalising spatial data and dealing with multiple representations". In: Geographic Information Systems – Principles and Technical Issues. 2. ed. John Wiley & Sons pp. 125–155.

Williamson, I.; Rajabifard, A.; Feeney, M.-E. F. (2003): „Future Directions in SDI Development". In: Developing Spatial Data Infrastructures: From concept to reality. Taylor & Francis pp. 301–312. — ISBN: 0–415–30265-X.

AUSSCHREIBUNG

FÖRDERPREIS GEOINFORMATIK DES RUNDER TISCH GIS E.V.

BESTIMMUNGEN ÜBER DIE VERLEIHUNG DES FÖRDERPREISES GEOINFORMATIK DES RUNDER TISCH GIS E.V.

Der Runder Tisch Geoinformationssysteme e.V. an der Technischen Universität München verleiht im Rahmen des jährlich stattfindenden Münchner Fortbildungsseminars seinen Förderpreis. Mit dem Förderpreis werden herausragende Abschlussarbeiten (Bachelorarbeit, Master Thesis, Diplomarbeit, Dissertation) ausgezeichnet. Um eine ausgewogene Vergleichbarkeit zu gewährleisten, werden i.d.R. ein Preis für die beste Bachelor-, Master- oder Diplomarbeit und ein Preis für die beste Dissertation verliehen. Die Förderpreisträger werden mit einer Urkunde und einem Preisgeld von insgesamt 4.000,- € geehrt. Ferner besteht die Möglichkeit, die wesentlichen Ergebnisse der Arbeit in der Zeitschrift GIS.Science zu publizieren. Die Siegerarbeiten werden vom abc-Verlag als Buch herausgegeben.

Für die Auszeichnung kommen Arbeiten in Betracht, die im Umfeld der Geoinformatik angesiedelt sind. Die Arbeiten müssen folgende Bedingungen erfüllen:

- Der Schwerpunkt der Arbeit muss im Bereich der Geoinformatik angesiedelt sein. Es können Arbeiten eingereicht werden, die sich mit GIS-Technologie und/oder GIS-Anwendungen beschäftigen.
- Die Arbeit muss auf eigenen Erkenntnissen beruhen und der Erweiterung der Kenntnisse im Bereich der Geoinformatik dienen. Die Arbeit muss grundlegend neue Entwicklungen aus Forschung und Anwendung der GIS-Technologie enthalten oder andere wichtige Beiträge für die Entwicklung des Geoinformationsmarktes liefern.

TEILNAHMEBEDINGUNGEN:

- Teilnehmen können alle Studenten resp. Absolventen aus Präsenz- oder Fernstudiengängen und Mitarbeiter von Technischen Hochschulen, Universitäten und Fachhochschulen im deutschsprachigen Raum.
- Die Arbeit muss mit der Note „sehr gut" bewertet sein.
- Eine Arbeit darf nur einmal eingereicht werden.

EINZUREICHEN SIND:

- Eine formlose Bewerbung um den Förderpreis des Runder Tisch GIS e.V. mit Angabe des Themas der Arbeit
- Bewertung durch die Betreuerin bzw. den Betreuer
- eine digitale Fassung der Arbeit
- eine digitale Zusammenfassung der Arbeit in einer allgemein verständlichen, öffentlichkeitswirksamen Darstellung im Umfang von einer Seite
- Es besteht die Möglichkeit, die eingereichte Arbeit im Rahmen einer Posterpräsentation auf dem Münchner Fortbildungsseminar vorzustellen. Die Teilnahme am Fortbildungsseminar ist für Förderpreisteilnehmer kostenlos.

GUTACHTERTEAM:

Die Jury zur Vergabe des Förderpreises Geoinformatik besteht aus vier unabhängigen Gutachtern, welche die Arbeiten nach einem vorgegebenen Bewertungskatalog beurteilen. Das Gutachterteam setzt sich aus Wissenschaftlern aus dem gesamten deutschsprachigen Raum zusammen.

ALLGEMEINES:

- Die Verleihung des Förderpreises an eine Arbeit, die von mehreren Personen verfasst wurde, ist nicht ausgeschlossen.
- Die Arbeiten sind in der Geschäftsstelle des Runder Tisch GIS e.V. einzureichen.
- Einsendeschluss ist jeweils der 31. Oktober des laufenden Kalenderjahres.
- Der Abschluss der Arbeit darf bei der Einreichung nicht länger als 12 Monate zurückliegen.
- Der Rechtsweg ist ausgeschlossen.

Weitere Informationen erteilt die Geschäftsstelle.

Vorstand:
Univ.-Prof. Dr. M. Schilcher (Vorsitzender)
Dr. Klement Aringer
Dipl.-Ing. U. Voerkelius (2. Vorstand)
Dipl.-Ing. H. Gotthardt
Dr. Ulrich Huber
Univ.-Prof. Dr. A. Koch

D-22297 Hamburg
E: jochen.schiewe@hcu-hamburg.de

Geschäftsstelle:
Technische Universität München
Fachgebiet Geoinformationssysteme
Dr. Gabriele Aumann
Arcisstr. 21, 80333 München
T: +49 89 289–22857
F: +49 89 289–22878
E: runder-tisch@bv.tum.de
I: www.runder-tisch-gis.de
Bearbeiter: Dr. Gabriele Aumann

RUNDER TISCH GIS E.V.

EVENT REPORT // VERANSTALTUNGSBERICHT

CONFERENCE REPORT: 12TH AGILE CONFERENCE, HANNOVER, 2009

Jochen Schiewe, Hamburg

From June 2nd to 5th, 2009, the 12th AGILE International Conference on Geographic Information Science was held at Leibniz University Hannover, Germany. Together with the ISPRS Workshop "High-Resolution Earth Imaging for Geospatial Information" this event brought together about 300 experts, 200 of them officially registered for the AGILE meeting.

This high number was preceded by about 130 submissions for the conference, 71 of them as full paper contributions which went through a double blind review process. Out of that, 22 full papers, 38 short papers and 33 posters could be presented during the conference in three parallel sessions. With that, an interesting and qualitatively outstanding selection was made which covered the whole range of data modeling, analysis and visualization topics. This setting was supplemented by the ISPRS Workshop sessions with an emphasis on data acquisition and processing based on imagery and LIDAR.

In addition, common keynote sessions of both meetings brought together all participants. Here, Gerd Binnig (Germany), Nobel Prize Winner in Physics in 1986 and founder of Definiens gave a review on "Principles of Human Cognition Utilized for Automated Image Analysis", covering not only topographical but also medical imagery. Barbara Koch (Albert-Ludwigs-University of Freiburg, Gemany) showed examples from her work in landscape modeling based on multi-sensor data in a GIS environment. In a very inspiring speech, Hanan Samet, Professor at the Department of Computer Science at University of Maryland (USA) reflected on the developments of spatial data structures. Finally, Frederik Jung-Rothenhäusler, Head of Product Development at RapidEye AG (Germany), gave an insight into products and services of his company which is a geospatial solutions provider that also operates a constellation of Earth observation satellites.

The meeting was supplemented by seven pre-conference Workshops dealing with very up-to-date topics ranging from Grid Technologies over Early Warning and spatial communication issues to education in Geoinformatics. About 100 participants made these workshops to a substantial, complementary part to the main conference.

The conferences were accompanied by an interesting joint social program, including an evening run and the highlight of a reception at the Lord Mayor of the City of Hannover and a following Dinner in the New Town Hall.

In conclusion, the local organizer, Monika Sester (Leibniz University Hannover), together with her ISPRS counterpart, Christian Heipke, accomplished the balancing act between large quantity (in terms of a huge number of participants) and high quality (with respect to the technical program). The applied reviewing procedure, a suitable time schedule including long breaks and a good location made this event to a success.

Proceedings of the AGILE conference are split into two parts, a Springer book containing the selected full papers and a CD-ROM with all other contributions:

Sester, M, Bernard, L. & Paelke, V. (2009, Eds.): Advances in GIScience. Proceedings of the 12th AGILE International Conference on Geographic Information Science, Springer-Verlag, Berlin.

Haunert, J.-H., Kieler, B. & Milde, J. (2009, Eds.): Proceedings of the 12th AGILE International Conference on Geographic Information Science, CD-ROM.

Author

Prof.-Dr. Jochen Schiewe
HafenCity University Hamburg
Lab for Geoinformatics and Geovisualization
Hebebrandstr. 1
D-22297 Hamburg
E: jochen.schiewe@hcu-hamburg.de

EVENT REPORT // VERANSTALTUNGSBERICHT

GRID TECHNOLOGIES FOR GEOSPATIAL APPLICATIONS AGILE 2009 WORKSHOP

Patrick Maué, Dr. Christian Kiehle

Presentation by P. Maué

Bringing the benefits of Grid technologies into spatial data infrastructures (SDI) has certainly a promising sound to it. Massive parallelization of complex calculations, instantaneous access to an abundance of geospatial data, no need for cost-intensive hardware resources, the list of apparent advantages of Grids is long. On the other hand, challenging problems make it difficult to integrate Grids into SDIs, and the list is long.

Understanding how we can integrate these two solutions for large-scale infrastructures is one of the major goals of the German GDI-Grid project. The underlying technologies are technically quite similar. On first sight it appears to be simply a matter of implementing new interfaces for existing components to enable exchange of data within both networks. In fact we realized early in the project that our vision of seamless integrated Grids within SDIs will mainly be driven by standardization. Web Services with standard-compliant interfaces interact with other Web Services according to standardized rules for security and the encoding of data and meta-data. The software engineers among us were confident: just glue two Application Programming Interfaces together and the SDI is grid-enabled.

Eventually, after various meetings discussing about security constraints or about the quest for defining OGF-compliant interfaces (needed within Grids) for OGC-compliant interfaces, we realized that we've lost the underlying research principles on the way.

We were caught in discussions about minor standardization issues, and we realized that, even though interesting for some, we have to also discuss how the geospatial community can actually benefit from Grid technologies. The long list of benefits mentioned earlier: do they really exist, do contemporary SDIs lack these properties, do we really need these technologies? Are there any novel research studies, algorithms, or applications, which could not yet be used within SDIs due to resource constraints (and which will be here in the future once Grids are incorporated into SDIs)?

These and many other questions were raised and discussed in the Workshop "Grid Technologies for Geospatial Applications", organized by the editors of this special issue in conjunction with the 12th AGILE International Conference on Geographic Information Science 2009 in Hannover, Germany. The workshop was mainly targeting researchers from the GIScience community, and the thematic range of the submissions matched this expectation. We asked for short position statements which discuss open GIScience research issues which can be better addressed through Grid technologies. We wanted to know about potential implications of Grids for geospatial applications, what kinds of geospatial algorithms can benefit of parallelization, or how traditionally popular topics within GIScience such as Data Quality or Data Semantics relate to Grids. The submissions were reviewed by an international program committee with experts from GIScience, Geosciences (in particular environmental modeling) and computer science communities. Authors of the accepted papers were invited to give a short presentation on their research, the program which includes links to the submissions is available online at http://purl.net/ifgi/agile2009.

The workshop was from its early beginnings planned as discussion platform. Even though we had submissions and presentations during the workshop, we were mainly interested in having a large number of experts (we had around 20 participants in the workshop) to discuss urgent research issues and may be come up with a better idea what science between Grid and GIS should care about in the immediate future. We asked the audience to discuss where we currently stand, what are the next (and following) steps, and what are the major roadblocks which slow us down (Readers used to agile software development may recognize these questions from Scrum methodology.)? Hence, we organized four working groups each assigned with one topic and asked the participants to prepare a presentation wrapping up their findings at the end of the workshop. The topics were Grids and Algorithms, Security, Standards and Markets. Not one participant was willing to join the last working group about economic issues. The rising popularity of cloud computing and its commercial success raised the public awareness of Grid technologies, we expected to have at least few participants who may have been interested in the scientific implications of this phenomenon. But the other three topics were apparently more interesting, the finding of the individual groups are discussed in more detail in the remainder.

GRID & ALGORITHMS

The current trend of pushing more and more processing units already in single desktop computers leverages the development of dividable algorithms. Many simulations could benefit massively from parallelization, but the underlying algorithms have been often developed many years ago, usually not with parallelization in mind. In the case of spatial algorithms, one might argue that the spatial division (splitting up the data into tiles) should be sufficient. The algorithms are then simply applied to small tiles, and consequently much more efficient. In the end, the results coming from many different nodes are then simply merged (stitched together).

Although this sounds like a convenient solution and hence like a perfect answer to the question how to parallelize existing algorithms, we have to acknowledge that most algorithms working on whole data sets behave differently on smaller tiles. Environmental models, e.g. for the computation of noise dispersion, have to take (as the name implies) the surrounding environment into account. Geospatial phenomena like weather, floods, forest fires, and others, can not simply be divided into tiles. Each phenomenon has dependencies to its spatial and temporal surroundings. An algorithm computing a local weather forecast has to be aware of the recent weather conditions as well as the current (and projected) situation in the neighboring regions. Accordingly, the development of algorithms which can be distributed to Grids is not simply a matter of splitting up the data into isolated junks, with each representing a small fraction decoupled from its neighbors in space and time. Dividing means to find a scalable and generic solution to split up data without loosing topological relations, allowing for instant access to surrounding content, and formally identifying the parallelization constraints for the individual algorithms.

In the scientific community, the need for parallelization (and hence for Grids) has long been acknowledged, and research projects depending on either massive storage or resource-intensive computations (with the Large Hadron Collider at the European CERN research institute certainly the most famous example) are already depend on Grids. But widespread success of Grid technology – ranging from research, the public sector (including military), commercial applications, and even the personal use – requires more incentives. The development of new algorithms which demonstrate the promised benefits may leverage the acceptance. Hence, as for most novel technologies, the Grid is in need for killer applications.

GRID & STANDARDS

SDI development is mainly driven by standards from the Open Geospatial Consortium (OGC). On the other hand, standardization within the Grid computing community is driven by the Open Grid Forum (OGF). Both standardization models differ significantly and have been discussed in both standardization organizations for a long time. Different approaches to service description, modeling stateful Web Services, security mechanisms and interfaces have been discussed. The following questions were raised, and consequently discussed in the working group:

1 How should a standardization process be structured?
2 Should OGC focus on domain-specific tasks rather then standardizing network protocols?
3 Who is the target audience for standards?
4 What are the metrics for measuring weather a standard is a success?

These four questions could have been raised in any OGC-related discussion, having them asked and discussed (quite intensely) here did surprise us. Our current issue of merging the OGC standards with the standards coming from OGF are mostly driven by one problem: OGC members have specified how to access OGC compliant Web Services. Although a reasonable step in the mid nineties (where no adequate alternative existed), it can be rather considered as roadblock today. Consequently, the second question implies that, in a perfect world, OGC would focus on standardizing domain-specific tasks. OGC standards should, for example, address the semantics of the Web Service operations or the encoding of geospatial data, but not what protocols to use to access a Web Service, or how to implement security constraints. Otherwise, we end up struggling with conflicting standards (which is the current situation in between the OGF (and W3C) and the OGC).

The fourth question also triggered an interesting debate on how to identify when it is worth to invest into an emerging standard (or when to wait whether it either disappears or gets commonly accepted). Within the OGC (but also the W3C or OGC), several standards are recommended every year, and only few remain over the years. It was argued, that the answer for the first question does actually indicate whether a standard is prone to failure. We identified the building of a community which implements and tests the recommended standards as one crucial step, standards which lack the community obviously also lack a future. How to identify if a standard has a thriving and expanding community is another question, though.

5 Who will be grid-enabled?

This last question refers to the interesting question, if the deployment of Grid technologies should actually have an impact on the exposed standardized interfaces. If only the underlying process is distributed to the Grid, why should the interface to the Web, e.g. the WPS, be affected? The answer is obviously complicated and application-dependent. In most cases, the end user using some Desktop GIS as client to the Web Service should not be aware of neither the Standards used for the Interface nor the implementation-specific details like a distribution to a Grid. But issues like trust (maybe the user doesn't like to have her data send to remote nodes) or security (not every user can simply load data on every node) have to be addressed and communicated on the client-side. Hence, even though it sounds convenient to encapsulate Grid technology within the algorithms, it depends on the application how much of this should be known to the end user.

GRID & SECURITY

Most of the AGILE workshop participants were involved in the Geoscientific community. Although, largely having a background in Computer Sciences and Software Engineering, the application of security mechanisms generally is not within the portfolio of most people dealing with GIS and SDI technologies. Grid computing on the other hand relies heavily on security mechanisms, since Grids are always subject to intrusion. Security mechanisms are also only fairly covered by OGC standards and therefore provide a high barrier for Geoscientists approaching Grid technology for the first time.

During the workshop research questions regarding Grid and security were discussed. An emphasis was put on the coupling of the security concepts of Grid computing and the geospatial community. Different approaches used in SDIs (GeoDRM, Webservice-Security, etc.) were discussed in contrast to more Grid-centric solutions based on the Grid Security Infrastructure (GSI). Lessons can be learned from the U.K. eScience initiative which relies on Shibboleth for about 8 Mio. users; parts of this infrastructure have also been coupled with spatial web-services.

The discussion lead to the conclusion that currently there is a lack of guidance on which security approach could be used for which scenario. In general, the security infrastructure chosen inside Geo-Grid projects is problematical since it oftentimes breaks existing client solutions. Furthermore, sophisticated security mechanisms like provided by GSI are based on a Public-Key-Infrastructure (PKI), complicating the interaction with a Grid-based SDI. Further effort should be put in providing best practice studies on security integration into existing SDI components. This seems to be a field of activity where SDI technologies can gain massive input from Grid infrastructures.

CONCLUSION

The interest in the workshop demonstrated that within the GIScience community Grid computing is an issue. With SDI technology becoming more and more mature, large scale problems can be targeted with distributed computing platforms. The concept of Grid computing sounds promising but leads to anther iteration: relevant application fields will have to be discovered, existing algorithms and libraries will have to be optimized (where applicable) to run parallel on distributed computers, standardized interfaces will have to be adapted, security issues will have to be integrated.

It became evident that geospatial sciences will have to adapt paradigms and techniques from the grid computing community in order to achieve a successful integration of grid technologies into geospatial applications. And there seems to be a lot of research to be done on various frontiers.

We hope that the selected papers offer a good overview of the emerging research questions and that they stimulate a vital scientific discussion. ◀

Presentation by A. Shaon

Authors

Dr. Christian Kiehle
lat/lon GmbH
Aennchenstrasse 19
D-53177 Bonn
E: kiehle@lat-lon.de

Dipl.-Geoinf. Patrick Maué
University of Muenster
Institute for Geoinformatics
Weseler Strasse 253
D-48151 Muenster
E: pajoma@uni-muenster.de