Geographic Grid-Computing and HPC empowering Dynamical Visualisation for Geoscientific Information Systems

Claus-Peter Rückemann\textsuperscript{1,2,3}

\textsuperscript{1} Westfälische Wilhelms-Universität Münster (WWU), Münster, Germany
\textsuperscript{2} Regionales Rechenzentrum für Niedersachsen (RRZN), Leibniz Universität Hannover (LUH), Schloßwender Str. 5, 30159 Hannover, Germany
\textsuperscript{3} L3S Research Centre, Appelstr. 9a, 30167 Hannover, Germany
ruckema@uni-muenster.de

Abstract. This paper gives an overview of the potential of the current implementation of portable components for Geoscientific Information Systems (GIS) within the GISIG actmap-project. The computing problems addressed are multifold and for the first time presented here: With Active Source having extended the framework for conventional GIS, new features have been enabled like the use of Grid Computing and cluster resources, dynamical visualisation, and High Performance Computing (HPC) in order to be used for Geographic Grid Computing. Base of scientific content can for example be geophysical information like environmental or seismological data, geographical and spatial information using Geographic Data Infrastructures (GDI), as well as data from industrial, economic, cultural, and social sources. An integrated solution for monitoring, accounting, billing supporting the geo-information market can be incorporated into this context. An outlook is given for Geographic Grid Computing e.g. for the extended use of Web Services and GDI in the future.

Key words: Grid-Computing; High Performance Computing; HPC; Dynamical Visualisation; Geoscientific Information Systems; GIS; Geocognostic Views; GDI; Accounting; Billing; Cluster Computing

1 Introduction

1.1 Novelty in a Nutshell

Starting from standalone GIS applications on local hosts long over ten years ago the proof had to be done that dynamic visualisation of scientific information can be successfully realised on distributed computing and storage resources by using a fundamental scripting approach. Over the years a Grid-GIS framework with many features had to be implemented including several programming libraries providing a suitable API. The problem of dynamic cartography and geocognostic views with hundreds of thousands of data points having to be connected with live, quasi real time data being very computing intensive had to be solved. The
following sections sum up selected basics, case studies and evaluation regarding the developments of a portable, modular, scriptable, and scalable solution using HPC and Grid Computing resources at the user level for solving the problem of bringing distributed resources and scientific content together.

1.2 GIS, Grid, and HPC Working on the GISIG Implementation

Encouraging powerful computing resources for the use with spatial information and scientific visualisation in practice still does link with a bunch of obstacles, some important ones missing or being insufficient are:

- integrability of concepts,
- portability of implementations,
- interfaces for data and application interchange,
- framework for the use of computing resources,
- availability of sources,
- extendability of existing methods,
- frameworks for application of methods needed,
- reusability of existing solutions, and many more.

The implementation of portable components within the GISIG actmap-project [Rüc05] over the last years aims to extend the features and applicability of Geo-scientific Information Systems (GIS) for these purposes (e.g. [CPG99, Zer00, Sch01]).

Besides the named obstacles, inter-GIS-Computing targets are to

- enable the use of computing resources for GIS, spatial information systems, dynamical visualisation, dynamical cartography, virtual reality, and multimedia presentation,
- exploit Grid Computing for GIS,
- exploit High Performance Computing (HPC)/Supercomputing for GIS,
- exploit Cluster Computing for GIS.

Combined efforts [HET07, OGC07, OGF07] can upgrade the motivating forces for Geographic Grid Computing bringing the necessary disciplines together.

1.3 Disciplines Working on the Content

Base of scientific content can be any information that can be represented digitally. Favorable in this context is the ability for multimedia presentations, for example using geophysical information like environmental information or seismological data, geographical and spatial information using Geographic Data Infrastructures (GDI). Data for example from industrial, economic, cultural and social sources can be used in that way, too.

Any of the informations can be combined with user defined dynamical algorithms with or without spatial context to form new cognitive views.
2 Grid-GIS Framework Implemented

2.1 GISIG operations and extendability

Scripting does enable GISIG components to use distributed computing resources like HPC, Grid Computing and Cluster Computing resources with mechanisms from pseudo-interactive to batch use. Arbitrary services for a wide range of scientific fields can be built upon these mechanisms. Services and applications can act on top of the Grid middleware infrastructure like Globus Toolkit [Globus06] and SGAS for this purpose. In detail, at any state of the application operations can be done onto data, information, and configuration regarding nearly every piece of algorithm and implementation. Examples are regexp operations, substitution, item configuration, and remote control.

Multimedia objects like source animations, videos, sound features and many more can be integrated into the data base on base of canvas embedding and event binding, for example in order to support complex geocognostic cartographic views. It is possible to create runtime functions in real time, to do replication, to clone parts of applications, use user defined servers and clients even inside the application, to do user or application defined history management, or even to use data consisting of GISIG Object Sources (GOS) [R"uc01b]. Flexible event databases are integrated and can e.g. be used interactive and in batch mode via scripting. Internationalisation is possible at database level as well as on application and data level. Security levels can be defined and configured as well as sandbox models and trusted computing. Components containing all the parts needed, including bytecode and data, can be compiled into self-contained executables plus separate optional runtime-time containers. For extended use even own kernel modules are possible. User applications can be configured for use with workstations to PDAs, while as the basic framework application is highly portable. Testing has so far been done with scripting, dynamical visualisation and cartography respectively mapping using Tcl/Tk [Tcl06], VTK, PV-Wave, C, Fortran, Perl, and Shell.

The following examples for using the ActiveSource framework as being part of GISIG will show a tiny part of the multitude of possible applications. The features shown give an impression of the connections available now between the GIS and the Grid world and its application background in Grid Computing, reaching from GISIG Object Sources to remote control, IPC, and the use of cluster resources.

2.2 Selected Insights to Active Source Framework

The concept has been described in detail in [R"uc01b] Parts of the Implementation base are available on the Internet [R"uc05, R"uc01f, R"uc01e, R"uc01c, R"uc04, R"uc01d, R"uc01a, R"uc01g] In the following passages some small feature snippets from the implementation are presented.
2.3 Active Source

Listing 1.1 shows a simple code fragment of a data set for an Active Map layer based on GISIG Active Source. Active Source can be pure source code or byte-code. The fragment shown is in parts a native data language representation. Ellipses (...) are shown for those parts of the real data set missing here for compactness.

Listing 1.1. GISIG Active Source code fragment.

2.4 Object Graphics

An example fragment of an object graphics data set with completely native data language is given in listing 1.2.

Listing 1.2. Object graphics code fragment.

2.5 Remote Control

Listing 1.3 shows a code fragment for remote control of objects in active instances of two components (actmap and actsea).

Listing 1.3. Remote control code fragment.
2.6 Inter-Process Communication

The example in listing 1.4 shows handling for child process and fileevent using a channel.

```
proc was (arg) {
    global jobFinished
    puts "Still at $arg"
    if { ![eof $arg] } {
        gets $arg data
        if ![eof $arg] {
            set jobFinished 1
            catch {close $arg}
            puts "EOF reached"
            return
        }
    }
}

set f [open "|calc|" r]
fileconfigure $f -buffering none -blocking no
fileevent $f readable "was $f"
wait jobFinished
exit
```

**Listing 1.4.** Child process and fileevent (channel).

Inter-Process Communication (IPC) holds very powerful functionalities for application communication. The basic ability is to execute a script containing an algorithm when a channel gets readable or writable. This way file event handler between a channel and a script or event can be created. For example GISIG IPC via Tool Command Language (TCL) provides a flexible fileevent and send (e.g. X send) and goes far beyond the features of other modern shells.

2.7 Computing Resources

With the scripting features various resources can be used via Grid, Cluster, and High Performance Computing at this level. Examples for batch systems used are Portable Batch System (PBS) [OPBS06], Cluster Computing software like Condor [Lew05], and LoadLeveler which have been successfully used. Tests with Sun Grid Engine [SGE06] are under way.

The example in listing 1.5 shows a Condor job using distributed resources on a cluster.

```
universe = standard
executable = /home/cpr/grid/job.exe
should_transfer_files = YES
transfer_input_files = job.exe, job.input
input = job.input
output = job.output
error = job.error
log = job.log
notify_user = ruckena@uni-muenster.de
requirements = (Memory >= 50)
requirements = ( (OpSys=="Linux")|((OpSys=="AIX")&& (Memory >= 500) )
queue
```

**Listing 1.5.** Condor cluster job.
The following listing (listing 1.6) shows a collection of commands for handling Condor jobs. Any of these commands can be integrated into the GISIG event databases, datasets, and applications.

```bash
condor_compile g77 -g77libs job.f
condor_store_cred delete
condor_store_cred add
condor_submit job.sub
condor_status
condor_q -analyze
condor_q -run
condor_userprio -all -allusers
condor_rm Job-ID
condor_rm -all
condor_hold Job-ID
condor_release Job-ID
```

**Listing 1.6.** Condor handling.

3 Selected Case Studies

The following case studies show various GIS applications, for example geocog-nostic views and dynamic cartography, using data, information and events from distributed storage resources, using distributed computing resources for live plotting and raytracing as well as different hardware resources.

3.1 Spatial Data and Active Source

GISIG Active Maps can consist of vector and raster layers as well as of multimedia parts and events. The example shows a dynamical event-driven city map containing environmental and infrastructure data that is delivered from distributed sources (figure 1).

![Fig. 1. Active Map with vector layers, raster layers, and events.](image)
3.2 Geocognostic Views

Most flexible geocognostic views can be developed using the local and background computing resources.

The example shows cartography combined with aerial data, and vector data all bound together by events (figure 2). The selected part shown is a highly zoomed area of the previously presented map, here in different thematical geocognostic context.

![Fig. 2. Active Map combined geocognostic view with map data, aerial data, and vector data.](image)

3.3 Configuration for Hardware

The configuration of GISIG components is very flexible and adaptable to the hardware medium. Figure 3 shows the same application used for the previous to examples configured for PDA-like hardware.
Data, event mapping and so on are identical, only appearance of the application differs depending on hardware and configuration.

### 3.4 Cartographic Mapping

The number of objects handled in object source is only limited by the system and hardware used. Figure 4 shows a worldmap consisting of several hundred thousands of vector points in source. Any part may be delivered from computing and storage units on distributed resources, e.g. via HTTP or HTTPS.
3.5 Synthetic Data and Raytracing

Figure 5 shows an interactive dynamical presentation with data samples for a synthetic stone texture palette raytraced with POV-Ray on distributed computing resources.

Samples can be closely linked with GISIG components as well as loosely linked with any other applications. An example for a tiny closely linked component is the data window, including scrolling and dimming, shown in the upper right corner.

![Image of e-Science application with raytraced synthetic texture data.](image)

Fig. 5. e-Science application with raytraced synthetic texture data.

Live plotting with GNUplot plotting into an instance of the canvas or 3D visualisation are possible, too. More examples of that kind are provided on the Internet.

3.6 Dynamical Cartography and Visualisation with GISIG actmap

Figure 6 shows two examples for event-driven, dynamical cartography hat can be used standalone as well as in combination and using event links to distant resources.
Fig. 6. Dynamical cartography, event-driven.

Any part of this concept can be used by event steering for highly dynamical interactive applications.

4 Evaluation and Lessons Learned

Although the implementations presented here are already available, this is still work in progress as new fields for application are currently under development. An evaluation for the current state of development is given for the current state.

The implementation based on GISIG actmap is portable, can be used to integrate various concepts, delivers flexible interfaces, and enables the use of the computing resources needed, like Grid Computing, HPC and Cluster Computing.

For data and components, sources can be made available to any extend wished. It is extensible by a wide range of means and can integrate a lot of existing frameworks while parts still being most reusable.

Therefore various distributed resources –computing and storage resources– can be used with this concept for many scientific problems as were shown by the examples.

In the past years the current GISIG actmap implementation has been successfully used for applications in the fields of geoinformatics, geophysics, geology, environmental sciences, remote sensing, mathematics, physics, chemistry, and social sciences for the purpose of event driven, dynamical, and cognitive cartography, dynamical GIS, spatial event handling, cognitive visualisation integrating animation and virtual aspects, visualisation extended presentation.
Some of the features that have been used are kernel modules, internal servers and clients, scripting, trusted computing, event databases, Grid Computing, Cluster Computing, HPC.

To some extend testing and application has been done for commercial purpose. Many other examples have been made available on the Internet [Rüc05].

The development should make use of extensive collaboration between developers working for different disciplines in combination with the defined use of standards in order to reduce compatibility problems.

5 Future Work

As the features presented here are already implemented, this is still work in progress as new fields of application for Geographic Grid Computing are currently under development.

Planning has already begun for using Web Services via the Web Services Resource Framework (WSRF) and Geographic Data Infrastructures (GDI), supporting the Open Grid Service Architecture (OGSA) and Open Grid Services Infrastructure (OGSI) with the framework in the future. Interdisciplinary work [OGC07, OGF07] should be encouraged.

For both Grid and HPC monitoring, accounting, and billing will become just more pragmatic when differing models [RMv06, EGM+03, SGE+04, GEJ+06, BCM05, EGE05] can be overcome.

A lot of basic work [RMR+05, Rüc06] has been done within the D-Grid project [D-Grid07] with the result of a monitoring/accounting/billing concept on which the current prototype for the integrated, solution has been set up.

In order to support a working geo-information market, an integrated, “holistic”, modular solution [RGB07] for monitoring, accounting, and billing is needed.

Figure 7 shows the Grid-GIS framework, the “Grid-GIS house” as it can be used with GISIG. The framework still has to be upgraded regarding inter-level connectivity and still has to be extended using Web Services and common standards.

Basic fundamentals are Grid and HPC resources namely computing and storage resources. Based on this layer Grid middleware and Grid services are installed. Special services can be created for nearly any application needed at this level. Future joint efforts like HET [HET07] can help to build the necessary meta-organisation background for HPC and Grid Computing.

Main issues for enlivening the “Grid-GIS house” under aspects of the geo-information market are Grid accounting as well as trusted computing and security at the service level.
Grid-GIS services and Web Services interfacing with GISIG namely the actmap component are sitting on top of that layer providing the interface for the geo-information market while providing usage of Grid services for data collection and automation, data processing, and data transfer.

6 Summary and Concluding Remarks

This article gave insight into the employment of scripting languages and source code based persistent object data for enabling the use of computing resources for geoscientific information systems. The basic concept of object graphics based on Active Source has been developed within the GISIG actmap project.

Dynamical cartography and visualisation using event-driven databases and using distributed computing resources have been very successfully adopted for different use cases. The implementation offers a wide range of applications for dynamical visualisation and to remotely controlled and multimedia cartography and presentation using computational intensive processes.

Based on the concept put into practice for Grid Computing, Cluster Computing, and HPC the prototype offers a lot of flexibility for application and steering of resource usage.
Obstacles for the use of GIS with Grid Computing and HPC have been overcome with the present concept, although the conformity with standards will have to go on.

Integrability, portability, interfaces, computing framework, availability, extendability, application of methods, and reusability have been concisely demonstrated.

Basic work has been done for showing the direction of developments. Future interdisciplinary developments will more closely combine existing means with the use of Web Services and Geographic Data Infrastructures in order to encourage the ongoing achievements from the interaction of GIS, Grid Computing, and HPC and build the “Grid-GIS house” for the geo-information market.

Acknowledgement. With over ten years now, this research has been going on for a long period of time and a large number of people have been involved. I am grateful to all the persons supporting these efforts, especially Prof. Dr. Thomas Kreuser and all my colleagues at the Geological and Palaeontological Institute at the Westfälische Wilhelms-Universität Münster (WWU), Prof. Dr. Guido Wirtz and the colleagues at the Institute for Computer Science (WWU), Prof. Dr. Ulrich Streit an the colleagues at the Institute for Geoinformatics (WWU) all the persons involved at the Zentrum für Informationsverarbeitung (ZIV) in Münster for providing Grid and computing resources, the Umweltamt City of Münster for providing data, Vectaport Inc., Redwood City, California for discussion and cooperation, Hansa Luftbild Consulting International GmbH, Münster, for providing aerial data, Landesvermessungsamt Nordrhein-Westfalen for providing cartographic data, the members of the D-Grid initiative, especially Prof. Dr. Wolfgang Gentzsch for the coordination of D-Grid, and my colleagues at the Regionales Rechenzentrum für Niedersachsen (RRZN), Leibniz Universität Hannover, at HLRN, HLRS, LRZ, UniBwM, LMU, CERN, DESY, ZIB, FhG IAO, and FZK for fruitful discussions.

References


