Rethinking Spatial Processing in Data-Intensive Science

Christian Authmann, Christian Beilschmidt,
Johannes Drönner, Michael Mattig, Bernhard Seeger

Department of Mathematics and Computer Science
University of Marburg, Germany
{authmanc, beilschmidt, droenner, mattig, seeger}@mathematik.uni-marburg.de

Abstract: In this paper we address the problem of supporting data-driven science in geo-scientific application domains where scientists need to filter, analyze, and correlate large data sets in an interactive manner. We identify ten fundamental requirements for such a new type of processing system. They span from supporting spatial data types over using workflows for data lineage, to fast and flexible computations for exploratory analysis. Our study of related work looks at a range of established systems of different domains and shows significant drawbacks with respect to our requirements. We propose the **V**isualization, **A**ggregation & **T**ransformation system (VAT) as an architecture to overcome current limitations. It bases on standard software and implements performance-relevant parts using state-of-the-art technology like GPU computing. Our first comparison with other systems shows the validity of our approach.

1 Introduction

The management of data has become a critical issue in challenging problems of mankind related to climate change and biodiversity. An example is the current dramatic loss of species on earth. The Task Group on Data and Scenario Support for Impact and Climate Analysis of the Intergovernmental Panel on Climate Change (IPCC), which is the leading international body for the assessment of climate change, is dedicated to identifying the information needs in global research projects on climate impacts, adaptation, and mitigation. In the context of biodiversity, governments have given substantial financial support to build up large data centers for archiving and processing to enable scientists identifying the key indicators of species loss and deriving accurate models about biodiversity.

Our work is motivated by two ongoing biodiversity projects with different focus. GFBio[1] [DGG$^+$14] is an integrative German platform to provide a sustainable, service oriented, national data infrastructure facilitating data sharing and stimulating data-intensive science in the fields of biological and environmental research. Our task in the project is to provide added-value services and data management tools for analysis and visualization. IDESSA[2] is a research project that addresses the real problem of sustainable rangeland management in South African savannas. In both projects, we have to deal with very large heterogeneous data and scientists interested in extracting hidden information in an exploratory way.

[1] www.gfbio.org

[2] www.idessa.org

As exemplified in our projects, processing of spatial data has already met the three key properties – Variety, Volume and Velocity – of Big Data, even before the term was coined recently [MCB+11]. **The data is heterogeneous (Variety).** Researchers have to deal with varying data formats, coordinate reference systems, resolutions and uncertainties [BSE+12]. There are raster images which arrange the earth's surface in a regular grid where each cell contains a measurement value. They represent continuous phenomena like elevation, temperature or precipitation. There are point data sets that associate attributes to locations of interest, e.g. occurrences of a species or measurements of a sensor. In addition, there are line and polygonal data sets for modelling areas of interest like roads and the habitat of species. **The data is big (Volume).** A global raster with a cell size of 100 × 100 meters exceeds 100 gigabytes of data. Current satellite instruments like MODIS send 70 gigabytes of data per day, with the next generation of satellites increasing this by an order of magnitude. **Fast and responsive data processing (Velocity)** is difficult to achieve with traditional methods such that response times meet the expectations of interactive users. This is especially important for exploratory analyses and refinement.

The use of traditional tools on spatial data confines scientists to a limited set of questions that can be answered. Several approaches have been proposed to modernize the tools by applying Big Data principles to spatial data. We show the architecture of an integrated system, a holistic approach to vector and raster data, tiled processing, low-resolution previews and the efficient usage of modern hardware architectures.

The paper is structured as follows. It first examines requirements for an interactive exploratory research system in Section 2. Section 3 addresses related technologies and describes their shortcomings with respect to the requirements. The main part in Section 4 introduces our own approach for overcoming current limitations. The evaluation in Section 5 shows the validity of our approach by experimentally comparing it to other approaches.

2 Requirements

Scientists that are approaching emergent scientific challenges in biodiversity research face a diverse set of tasks to accomplish. We identified the following criteria (R1-R10) as mandatory for a system that effectively supports biodiversity science.

A support for raster images, point and polygon data is obligatory (R1). In addition to the geographical information, temporal information is inherent for nearly all data items. They either have a point in time of creation or a validity interval which must be considered for semantically sound computations. This leads to the necessity of a model of time and temporal semantics (R2). Data from different sources often has different formats and reference systems. A system should provide automatic data transformation techniques that allow matching data nonetheless (R3). It should recognize the need for such operations and apply them transparently to take the responsibility from the user.

The system has to support a rich set of low-level *operators* which perform well-defined transformations on data (R4). Point and polygon intersection, and filtering points by their attributes are two examples of such operators. To achieve higher level functionality, users can combine these operators into a *workflow* (R5). A workflow consists of a set of operators and their parameters, as well as a specification of the data flow between these operators. They are often modeled and visualized as directed graphs.

A standardized workflow format allows re-using, sharing and publishing of workflows (R6). Since a workflow contains all processing steps from raw measurements to the scientific results, they can provide *data lineage*. This is the traceability of the involved data. Publishing a workflow allows other scientists to verify the calculations and reproduce the results. Moreover, users are interested in downloading results of the computation in a desired format and use them within their preferred working environment.

Scientific research is often an iterative approach of forming an idea, testing a hypothesis, reviewing the results and forming a new idea which starts the iteration again. To support fast iterative research, the system must support flexible composition and modification of workflows. Results must be available in a near real-time fashion (R7).

With growing data sizes, it is increasingly expensive for researchers to keep local copies of relevant data and perform computations on their workstations. All data should thus be stored and processed on a central system which also provides commonly used data sets (R8). Researchers must also be able to upload custom data (R9). An intuitive and efficient Web interface shall allow researchers to explore data and generate workflows (R10).

3 Related Work

In this section we review existing systems and software components in whether they fulfill the aforementioned requirements. The traditional way to implement a workflow is to program its functionality manually. Scripting languages like R[3] offer functionality to load and process spatial data. We now look at systems from different categories like GIS, scientific databases, workflow systems and Big Data solutions.

Geographic Information Systems (GIS) are usually desktop applications with a rich user interface which allow loading, visualization and processing of spatial data. Systems like QGIS[4] offer a vast set of low and high level operators. Users however often need to unify the format of their input data manually beforehand (R4). A few GIS implementations offer basic lineage information and rudimentary workflows, but not to the extent required for exploratory data analysis (R7). As they are installed on a user's local machine they are limited to the available resources. Most current GIS implementations are not designed to handle data that exceeds a computer's main memory.

Recently more and more cloud GIS become available. They overcome the limitations of local hardware, but are mostly highly specialized applications that focus e.g. on data visualization rather than processing. One example is Map Of Life[5] that utilizes the Google Earth Engine[6]. However, it is difficult to extend these systems for other use cases (R9, R10).

[3] www.r-project.org

[4] www.qgis.org

[5] www.mol.org

[6] earthengine.google.org

Spatial data frameworks are on the other end of the spectrum. They offer low level functionality but are not feasible for end users as standalone systems (R10). Representatives are GDAL[7] for reading, writing and re-projecting spatial data and GEOS[8] for vector operations. They are very useful in a backend that hides the complicated usage from the user. However they do not exploit recent hardware developments sufficiently well (R7).

Scientific databases for spatial data focus on storing and processing rather than on visualization. There are specialized systems for handling large rasters like RasDaMan [BDF$^+$98] and SciDB [Bro10]. While they provide scalability, they lack support for vector data and other spatial data types (R1). PostGIS[9], as an extender for PostgreSQL, is much more of a general purpose system. Being based on a classical RDBMS, it lacks the expressiveness and performance required for spatial data analysis (R5, R7). It is suitable for storing and retrieving spatial data, not for analytical processing.

Scientific workflow systems [CVD$^+$12, VER$^+$15] are very prominent in the field of biodiversity informatics. They allow the creation of a process by specifying a data flow using a visual drag and drop editor. Taverna[10] is one of the most popular systems. It allows to orchestrate Web services and manages the data transmissions. Users can share workflows on platforms like BiodiversityCatalogue[11]. Others can then incorporate them e.g. as sub workflows in their own processes. Thus there is a lot of functionality already available. However, users are bound to the availability of external Web services and have limited control over changes of the services. This hinders reproducibility of computations (R6). Additionally Web services have protocol overhead and the orchestration of multiple distributed services leads to high network traffic. This causes slow execution of larger workloads and hinders the exploratory usage of such systems (R7).

In the context of Big Data, Hadoop [Lam10] is omnipresent. Hadoop-GIS extends Hadoop with GIS features [AWV$^+$13] which could make it a good fit for our scenario. It is however a batch processing framework. Long start-up times [DG08] and constant disk usage for intermediate results hinder our desired near real-time computation of workflows (R7).

Concluding our survey we did not find a single system that matches all of our requirements. A combination of the key features of the above mentioned systems is necessary to create an interactive processing system.

4 The VAT System

Our goal is to develop a system that meets the ten requirements of Section 2. This section gives a brief overview of our concepts and the state of development we started in 2014. Firstly, it introduces the architecture of the system and sets the scope for the rest of the section. Secondly, it describes our core design decision regarding the functionality and performance of the system.

[7] www.gdal.org

[8] geos.osgeo.org

[9] www.postgis.net

[10] www.taverna.org.uk

[11] www.biodiversitycatalogue.org

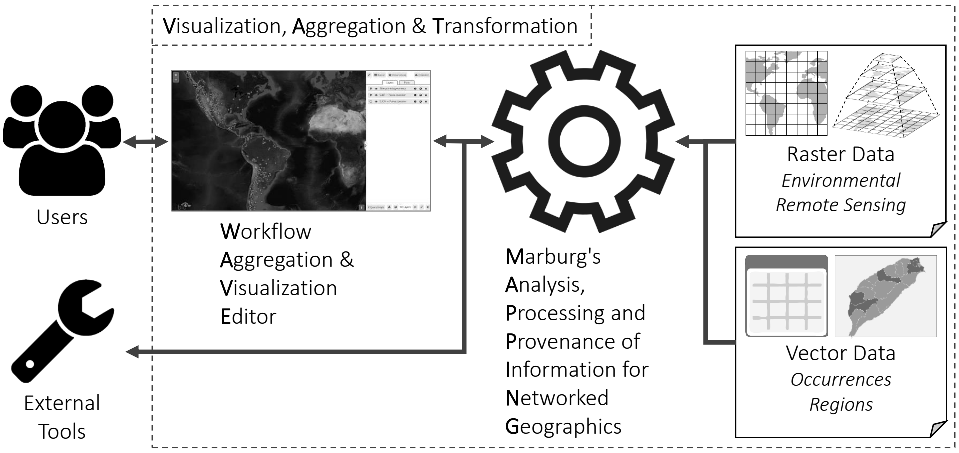
Visualization, Aggregation & Transformation
Users
External Tools
Workflow Aggregation & Visualization Editor
Marburg's Analysis, Processing and Provenance of Information for Networked Geographics
Raster Data
Environmental Remote Sensing
Vector Data
Occurrences Regions

Figure 1: Overview of the architecture of the VAT system.

Figure 1 depicts the architecture of our **V**isualization, **A**ggregation & **T**ransformation system (VAT). It consists of two main parts. The back end, called MAPPING, is responsible for managing and processing of data. The front end, called WAVE, is a web-based application for visualization and manipulation of data. It allows the definition of workflows. WAVE facilitates the interactive usage of MAPPING, but is not in the scope of the rest of this section. The producer side (right) shows the integration of different data types like raster and vector data from presumably different sources. The consumer (left) is either a user that communicates via WAVE or an external application that can interact with MAPPING directly via Web services and programming interfaces.

4.1 Functionality

VAT is an integrated system that allows us to have full control of each individual processing step. It follows a holistic approach and supports both raster and vector data. We store rasters in a read-efficient way using a custom engine with tiling and pyramids of different resolutions. Different levels of compression allow us to specify trade-offs between CPU time and storage space. We currently use PostGIS for the storage of vector data.

We make use of GDAL and GEOS for import, export and low-level operations whenever applicable. The system implements several standard operators as known from GIS programs, e.g. for the correlation of data from different types. Future work will include more operators, both for general and specialized applications.

MAPPING represents workflows internally as a directed tree which WAVE is able to visualize for comprehensibility. The result of the workflow is the root node, while the leafs represent the inputs. When one operator consumes the result of another operator that has a different coordinate reference systems than expected, VAT introduces a new re-projection operator that is plugged in automatically. This converts the output of the source operator such that the sink operator is able to use it. In the future it will be possible to perform parameter sweeps by automatically repeating computations for parameter values in a certain range. An example is to perform a computation for every month in a given year.

The result of a workflow must be reproducible in order to support the scientific method. This means if a user runs a workflow again at some point, it must yield the same results. The first step is to store workflows such that they can be re-used. Additionally, both data and operators have to be versioned. This way, users can execute workflows on old data with old versions of the operators, retrieving the same results as in the original execution. The system can recompute all results at any point in time and does not necessarily have to store them. To encourage sharing and collaboration, it will be possible to deliver workflows to others and also make use of existing workflows.

Standard-compliant interfaces are very important. We support import and export over standard OGC[12] protocols, like Web Map Service (WMS), Web Coverage Service (WCS) and Web Feature Service (WFS). This allows using our system as part of a larger workflow if desired. A researcher might use our system for computationally heavy tasks, then export the results to a local machine for analysis with specialized tools.

4.2 Performance

Section 2 disclosed that an exploratory approach highly benefits the research process. This makes fast response times a key aspect in getting users to accept the VAT system for their purposes. Performance is orthogonal to the functional requirements, but influenced our design decisions. We focus on optimizing performance on a single machine first. The distribution of workload across multiple machines will be addressed in our future work.

In general, only a subset of the data is needed for answering a query, e.g. if the region of interest is restricted to Germany. This is especially crucial in the context of rasters because of their size. We employ optimization techniques of current database technologies to optimize the processing and minimize the size of temporary results. To avoid reading unneeded data we push the region of interest in form of a query rectangle down to the source operators. This is comparable to the selection push-down in the relational database world. We can exploit this to constraint the data transmission as early as possible. In cases where different coordinate reference systems are involved, we transform the area of interest into other coordinate systems as required.

As mentioned in Subsection 4.1 we store pre-aggregated versions of rasters in pyramids. We exploit them to compute on lower resolutions. This accelerates the exploratory processing significantly, giving a researcher often enough information to decide whether to follow up on an idea. Users can pan and zoom on their map, allowing to calculate results only for the currently viewed area in the screen's resolution. When an interesting correlation is found, results can be computed in full detail.

[12] www.opengeospatial.org

From a performance perspective it is not desirable to move large data over the network to perform computations on it. Under the assumption that the gain in computational power is negligible in contrast to I/O costs, it is more reasonable to process the data where it resides. Our approach is therefore to choose function shipping over data shipping to minimize network traffic. We allow users to upload scripts (currently we support R) to the server, where they are executed close to the data. This allows users to implement custom processing steps without downloading the data and processing locally. The system executes a script by an operator as part of a workflow, allowing lineage and reproducibility even on workflows with custom scripts.

In order to enable fast and efficient data processing, it is essential to exploit a single machine's resources effectively. On modern hardware this means to parallelize work as much as possible to incorporate all CPU and GPU cores. GPUs are predestined for processing rasters, even though current GIS do not use them extensively. To combine CPU based operators with GPU accelerated ones, we are looking at Heterogeneous System Architecture[13] (HSA), where CPU and GPU share a common main memory address space. This avoids expensive copies of data from one memory space to the other.

5 Evaluation

To demonstrate the validity of our approach, we designed a use case for the processing of spatial data and compared ease of use as well as processing time in different systems. The use case is a plausible real-world scenario suitable for benchmarks and comparisons. We do not claim its scientific relevance – that is a question for biodiversity researchers.

5.1 Description of the Use Case

The goal of our use case is to find possible habitats for a species, in this case the European wildcat (*Felis silvestris silvestris*). According to the IUCN/SSC Cat Specialist Group[14], they have three requirements on their environment: 1) They avoid areas with human activity and prefer forests. 2) Areas with heavy snow cover are avoided. 3) They have been observed up to 2250m.

To produce a map of the possible habitats we used environmental data from *WorldClim* [HCP$^+$05] and the *Global Landcover Map for the year 2000* (GLC2000[15]). Figure 2 shows the example workflow. We used GLC2000 to look up human activity and forests, and classify the area according to the first criterion. The snow cover is more difficult to calculate. Without access to global snow cover measurements, we approximate the snow cover using WorldClim's temperature and precipitation data. Since WorldClim has monthly aggregated data, the snow cover is calculated separately for December and January. The sum of

[13] www.hsafoundation.com

[14] catsg.org

[15] bioval.jrc.ec.europa.eu/products/glc2000/glc2000.php

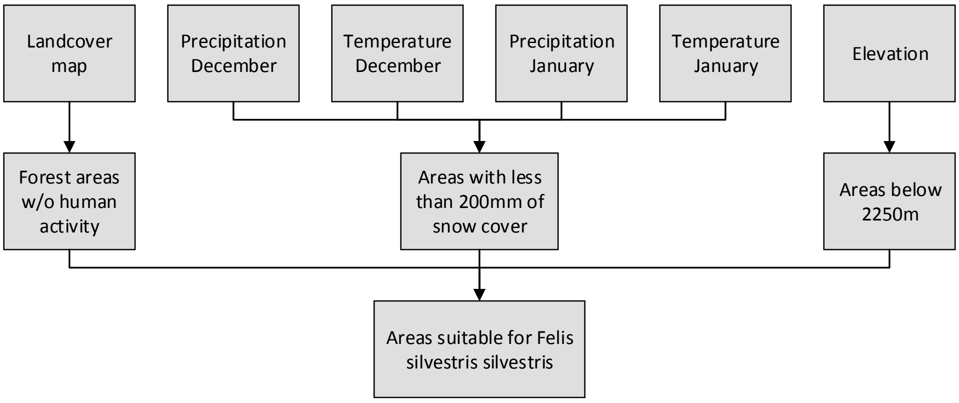

Figure 2: Workflow to find suitable habitats for wildcats

both values is compared to a threshold (below 200mm) to classify the area according to the second criterion. The third criterion uses the WorldClim elevation model to discard all areas with an elevation above 2250m. In a last processing step, we merge the classifications into a single result, showing the areas where all three classifications match.

5.2 Implementation

To facilitate comparison, we restricted the use case to raster data and transformed the data into the same size and coordinate system. While all tools should be able to handle different data types and coordinate systems, implementing the required conversions in multiple systems proved to be too laborious.

R: The *raster* package from R allows loading and processing of rasters. We implemented the workflow as a single-step computation. The multi-step computation implemented on all other systems proved to be too slow. The implementation of the computation required experience in R, but was otherwise straightforward.

QGIS: While QGIS itself provides a set of processing tools, it relies heavily on other GIS to use their algorithms. We used the *r.mapcalc* module from GRASS[16] to apply formulas to each pixel. The integrated graphical modeling tool was used to implement a workflow containing all steps of the computation. The visual tools were easy to use. The workflow was quick to implement and worked out of the box.

Taverna: Taverna is an orchestrator and requires external web services to handle the actual computations. We installed pyWPS[17] with GRASS as the processing back end. To avoid being limited by the network, the Web service was running on the same machine as Taverna.

[16] grass.osgeo.org

[17] pywps.wald.intevation.org

	MAPPING	QGIS	Taverna	R
Germany	0.7s	3.5s	53.1s	48s
Europe	3.8s	35s	-	3600s
World	29s	368s	-	18600s

Table 1: Measurement results of the use case experiment

The implementation using Taverna proved to be challenging. Setting up the Web service required manual programming and configuration. Implementing the workflow in Taverna was met with several difficult to debug errors, from XML encoding problems to out-of-memory errors. We managed to work around some of these problems, but were unable to run the workflow on the full data sets.

MAPPING: In our own system, all computation steps were implemented using built-in operators. The *expression* operator works similar to GRASS's *r.mapcalc* and was used for the majority of the computation. Our visual editor allows an easy implementation of the workflow, similar to QGIS. GPUs have not been exploited for this comparison.

5.3 Results

We ran all tests in a virtual machine on a consumer PC with an SSD and 16 GB of main memory. Table 1 shows the run time of the workflow execution on differently sized areas. The rasters for Germany are 1200×1200 pixels in size. The ones for Europe are roughly 40× bigger than for Germany and the whole world again is 12× bigger than Europe. The results show substantial improvements of MAPPING in comparison to its competitors. For Germany MAPPING is between 5× and 75× faster than the other systems. The difference is even more apparent for larger regions.

6 Conclusion and Future Work

We presented a general architecture and selected key design decisions for a system to solve emerging computational requirements in the spatial research domain. Our experiments showed the potential speed up our system is able to achieve even in the early state of development. This makes our approach interesting for interactive research on large data. However the system is currently not distributed and not scalable to any amount of data.

In our future work we will achieve scalability in our system by distributing data and computation across multiple machines. A geographical partitioning will allow splitting up a query and computing it on multiple regions in parallel. A disjoint partitioning may result in heavy network traffic if neighbor data is needed. Storing overlapping regions (cf. [Bro10]) helps diminishing this problem. Different regions could vary heavily in data volume and request load. An example is an unexplored sea-region in contrast to a national park. In this sense it is important to choose regions according to these factors. Additionally, it is useful to log the activity to have a better prediction for workloads.

Since workflows have a well-defined and never-changing result, it is possible to cache both intermediate and final results of workflows. Caching these results will speed up execution of different workflows with identical parts. We will examine existing caching solutions for their suitability for geo-aware caching.

Our evaluation is only able to give a first impression of the performance of systems in one representative but also specific use case. To reduce the bias in such comparisons, there is a demand for a unified benchmark. It has to capture the variety of data and workloads scientists encounter in their everyday research in the context of spatio-temporal data. This facilitates the development and evaluation of systems that are used for supporting scientific research questions.

References

[AWV$^+$13] A. Aji, F. Wang, H. Vo, et al. Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce. Proc. of the VLDB Endowment, Vol. 6(11), 2013.

[BDF$^+$98] P. Baumann, A. Dehmel, P. Furtado, et al. The Multidimensional Database System RasDaMan. In *ACM SIGMOD Conference*, SIGMOD '98, pages 575–577. ACM, 1998.

[Bro10] Paul G. Brown. Overview of sciDB: Large Scale Array Storage, Processing and Analysis. In *Proc. of the ACM SIGMOD Conference*, pages 963–968. ACM, 2010.

[BSE$^+$12] K. Bach, D. Schäfer, N. Enke, et al. A comparative evaluation of technical solutions for long-term data repositories in integrative biodiversity research. *Ecological Informatics*, 11:16–24, 2012.

[CVD$^+$12] V. Cuevas-Vicenttín, S. Dey, et al. Scientific Workflows and Provenance: Introduction and Research Opportunities. *Datenbank-Spektrum*, 12(3):193–203, 2012.

[DG08] J. Dean and S. Ghemawat. MapReduce. *Comm. of the ACM*, 51(1):107, 2008.

[DGG$^+$14] M. Diepenbroek, F. Göckner, P Grobe, et al. Towards an Integrated Biodiversity and Ecological Research Data Management and Archiving Platform: The German Federation for the Curation of Biological Data (GFBio): Informatik 2014 – Big Data Komplexität meistern. GI-Edition: LNI(232):1711–1724, 2014.

[HCP$^+$05] R. J. Hijmans, S. E. Cameron, J. L. Parra, et al. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology*, 25(15):1965–1978, 2005.

[Lam10] C. Lam. *Hadoop in Action*. Manning Pub., Greenwich, CT, USA, 1st edition, 2010.

[MCB$^+$11] J. Manyika, M. Chui, B. Brown, et al. *Big Data: The Next Frontier for Innovation, Competition, and Productivity*. McKinsey Global Institute, 2011.

[VER$^+$15] C. Vitolo, Y. Elkhatib, D. Reusser, et al. Web technologies for environmental Big Data. *Environmental Modelling & Software*, 63:185–198, 2015.