BIG GEOSPATIAL DATA PROCESSING IN THE IQMULUS CLOUD

Angéla Olasz[a], *Binh Nguyen Thai*[b,a], *Roberto Giachetta*[b,a], *Dániel Kristóf*[a]

[a]Institute of Geodesy, Cartography and Remote Sensing (FÖMI), Budapest, Hungary
[b]Eötvös Loránd University (ELTE), Faculty of Informatics, Budapest, Hungary

ABSTRACT

Remote sensing instruments are continuously evolving in terms of spatial, spectral and temporal resolutions and hence provide exponentially increasing amounts of raw data. These volumes increase significantly faster than computing speeds. All these techniques record lots of data, yet in different data models and representations; therefore, resulting datasets require harmonization and integration prior to deriving meaningful information from them. All in all, huge datasets are available but raw data is almost of no value if not processed, semantically enriched and quality checked. The derived information need to be transferred and published to all level of possible users (from decision makers to citizens). Up to now, there are only limited automatic procedures for this; thus, a wealth of information is latent in many datasets. This paper presents the first achievements of the IQmulus EU FP7 research and development project with respect to processing and analysis of big geospatial data in the context of flood and waterlogging detection.

Index Terms **—** Distributed computing, flood detection, Big Data, raster data partitioning,

1. INTRODUCTION

The four-year IQmulus project (http://www.iqmulus.eu), funded by the 7th Framework Programme of the European Union, is targeting to enable optimized use of large, heterogeneous geo-spatial data sets ("Big Geospatial Data") for better decision making through a high-volume fusion and analysis information management platform. An end-to-end involvement of users is ensured through the implementation of two concrete "test beds" (Maritime Spatial Planning & Land Applications for Rapid Response and Territorial Management) to show the benefits of the approach. The contribution of large number of users from different geospatial segments, application areas, institutions and countries have already been achieved. The consortium consist of partners representing numerous different facets of the geospatial world to ensure a value-creating process requiring collaboration among academia, authorities, national research institutes and industry.

User requirement collection along with scientific and technical state of the art analysis was carried out in the first phase of the project to identify relevant development directions, in which IQmulus can yield significant improvements. The project starts providing concrete results in terms of processing services and system architecture. Services consist of algorithms and workflows focusing on the following aspects: spatio-temporal data fusion, feature extraction, classification and correlation, multivariate surface generation, change detection and dynamics. System integration and testing is currently being done. The IQmulus system will be deployed as a federation of these modular services, fulfilling the needs of the above-mentioned scenarios but also suitable for the construction of further solutions thanks to the modular approach.

2. METHODS AND MATERIAL

According to IQmulus specification, services should run on distributed environment. The Apache Hadoop (http://hadoop.apache.org/) open source framework has been selected as a basis for architecture development. This framework offers the execution of large data sets across clusters of computers using simple programming model known as MapReduce. It is supplying the next generation cloud-based thinking and is designed to scale up from single servers to hundreds of workstations offering local computation and storage capacity. Thanks to this solution, users can achieve rapid response to their needs used by their own datasets in cloud processing [1]. However to take advantage of today's state of the art computing, previous data processing methodologies and workflows have to be revisited and redesigned. Actual deployment is based on a distributed architecture, with platform-level services and also basic processing services invoked from clusters and conventional computing environments.

2.1 Distributed File System (Hadoop Eco-system)

Data have become a part of everyday's work. Depending on application areas, some of them need high data availability; others need high computational functionality on large data sets. Scalability of any framework is a crucial point of

success or failure, however not everyone can afford large hardware investments in a short period of time, not to mention the processing power is not increased proportionally with hardware investment.
Hadoop is an open source framework/eco-system designed for capturing, organizing, storing and sharing, analyzing and processing data in a clustered environment giving the solution for linear scalability of both storage and computational power. The ability of storing and analyzing large amount of data on clustered computers reduces the invested cost remarkably [2].

Primary approach of Hadoop: "Data should be distributed and minimizing network latency by moving algorithms across clusters in a distributed manner". Hadoop have the following properties:

1. Data sits on one or more machines.
2. All nodes contain both distributed file system and data processing capability.

Hadoop is completely modular. Its core components are HDFS and MapReduce which is required; others are optional depending on your need (Figure 1). Some of the widely used components are HBase, Hive, Pig, and SpatialHadoop. Hadoop distributed file system have been designed to store text-based data. Incoming data are split into smaller chunks, namely data blocks which are 64MB by default, this limit have been increased to 256MB from version 2.x [3].

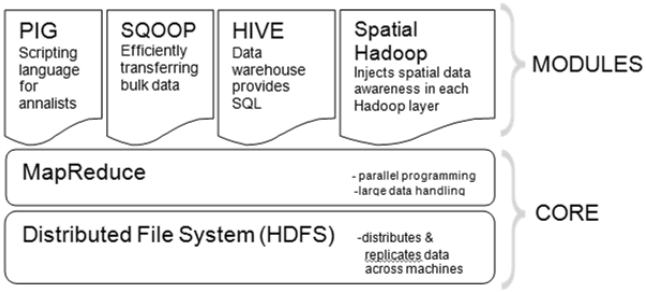

Figure 1. Architecture of Hadoop

Originally, HDFS is designed to store text-based data, however a large portion of our geospatial datasets are stored as structured binary files. Hence, data partitioning over HDFS had be solved in the first place. On the other hand, services are implemented in different languages varying from script to object-oriented languages, for example LIDAR processing services are implemented in C++, general processing services are implemented in Matlab, Java, Visual C++ and remote sensing algorithms related to land-based scenarios are implemented in C#. Therefore, the distributed framework has to support a wide range of runtime environments.

On architecture layer, we have to find solutions for data partitioning and supporting a runtime environment for services written in different programming languages.

2.2 Data Partitioning

To process raster data in distributed environment data decomposition is crucial. In distributed computing environment access data in a certain location, decompose to smaller chunks ('sub-raster blocks') to achieve optimal fast processing, collect the result of the algorithms from processing units and rebuild resulting data. Many approaches exist with different pattern of coordination and communication between the processing steps and results. The main aspect is how to decompose input raster to smaller partitions. A number of alternative methods exist to raster data partitioning as well. Single pixel algorithms are processed for each pixel in isolation. Defined extent algorithms are processed for each pixel in a context of a pre-determined number of surrounding pixels. Region-based algorithms are considered to apply in a geographical application where a greater neighbouring area has significance where the study area is grouped into homogeneous segment or stratum. The processing algorithms and parameters are designed for every homogeneous strata [4]. This stratum considered as a landscape unit which is homogeneous in aspect of physical geographical factors, or any targeted application area.

One of the most trivial ways to decompose data is to spatially split the original data into smaller data chunks, which can be processed independently on processing units. However decomposition methods are tightly related to processing methods and the number of processing units available in the distributed environment.

As a proof of concept, we have developed a GeoTIFF decomposer application using GDAL library. Decomposer takes a source file or directory, looking for GeoTIFF files and decompose them into smaller partitions. It takes two parameters, namely a predefined grid size (N x N) and number of processing units available in the distributed computing system. The number of processing unit is needed later for data distribution. After decomposition has been successfully performed, decomposed data are being uploaded to computation nodes.Within the next year, we will work on more sophisticated splitting algorithms to suffice user stories and showcases gathered in user requirement analysis phase [5].

3. RESULTS

3.1 Development of processing services

IQmulus develops functional and domain processing services in order to: maximize the use of big geospatial data, provide task-specific packing and delivery of data sets, support data quality evaluation, and provide support for

analyzing quickly changing environmental conditions. Services are developed to fulfill requirements related to data preparation, feature detection, change detection and presentation/visualization of big geospatial data.

The specific services and solutions put in the focus of the current presentation have been developed on the basis of the AEGIS open-source framework, a geospatial toolkit developed in a joint work between the Institute of Geodesy, Cartography and Remote Sensing (FÖMI) and the Eötvös Loránd University (ELTE), Faculty of Informatics [7].

3.2 Data management in the cloud

To enable geospatial processing in the cloud, the spatial data is preferred to be stored in the distributed file system. The storage forms are general spatial file formats (such as Shapefile, LAS, or GeoTIFF) to enable the direct consumption by most geospatial toolkits [8]. It must also be taken into account that large files are divided by the file system into smaller blocks before distribution to enable the parts to be processed individually. These blocks should be separately readable by the libraries, requiring all blocks to have a readable format. In contrary, the general partitioning methodology of HDFS does not take the file structure into account when splitting the input file.

A custom partitioning solution is provided, that performs partitioning and allocation of the datasets in HDFS using predefined strategies. Instead of the file blocks originally created by HDFS, these constructed items are individual files that can be interpreted element-wise. Strategies can be based on spatial extent, spectral space (in case of imagery) or any other property. Some strategies may have special purpose, e.g. creation of a pyramid image for visualization.

3.3 Processing framework

For implementation of the flood detection services of the IQmulus project (considered as Land Showcase 3) the AEGIS framework was used. The AEGIS framework is geospatial toolkit for both vector and raster data [8]. It is based on well-known standards and state of the art programming methodologies [9]. Due to its extensible infrastructure, it can support a variety of execution environments, including distributed platforms. A built in execution environment is responsible for handling algorithms, and managing the execution over the designated platform. In case the execution platform is Hadoop, AEGIS transforms the operations to use the MapReduce scheme. The MapReduce paradigm can be considered as the Map phase performing the initial operation, whilst the Reduce phase is responsible for post-processing the results [10, 11]. Thus, operations can be adapted using the following process:

1. When executing an operation based on specific spatial extent or other properties, the required files are selected using indices and the metadata catalogue.
2. Based on the specified method and input data, the operation is determined from the operation catalogue.
3. The operation and input are forwarded to an operations engine working over the Hadoop environment. Each input geometry is processed in parallel by separate MapReduce tasks.
4. If required, the results of the operations applied to multiple geometries can be merged using post-processing.
5. The result is written to the distributed file system by the Hadoop environment.

Post-processing is a key step, which is usually a kind of merge operation performed on the initial results. In many cases there is no need for post-processing. For example, binary thresholding of an image can be performed element wise, thus when processing partial images the results are independently valid without the need of any merging.
Some operations may require a special approach and cannot be applied directly. One such case is histogram equalization, which can be performed in two steps. The first step computes the histogram of the individual image parts in the Map phase, then merges the histograms and computes the mapping of the values in the Reduce phase. In the second step the mapping is applied to the parts using the Map stage, resulting in the transformed images. Thus, no Reduce function is required for the second part. Due to these circumstances, it cannot be stated that all existing algorithms can be applied directly in the MapReduce environment without any prior investigation, but the required additions and development are marginal compared to complete reimplementation.
Nevertheless, there is a need for an operation environment enabling the execution of algorithms as Map functions and performing optional post-processing as well, which is an operations engine run by AEGIS [7].

2.3. Workflow definition and execution

Domain-specific Languages (DSLs)
Concerning workflow definition and execution, the concept is to develop a set of domain-specific languages (DSLs) that make the definition of several types of spatial data processing and analysis tasks simpler. DSL can be very close to natural language, which is descriptive and expressive. A well designed domain specific language can be very expressive, useful and understandable even for non-software developers. However, modeling a natural language or processing natural languages is not an easy task; therefore we have to look for existing programming languages with very high descriptive and expressive power [6]. Workflows defined in a DSL can also be compiled to run on parallel processing environments. The compiled algorithms can be

packaged as declarative services to run on the respective environment they have been compiled for.

The Workflow Editor

The Workflow Editor is a web-based application using domain specific language to define geo-processing services and working dataset developed in the framework of the IQmulus project. The actual look of the workflow editor can be seen on the Figure 3.

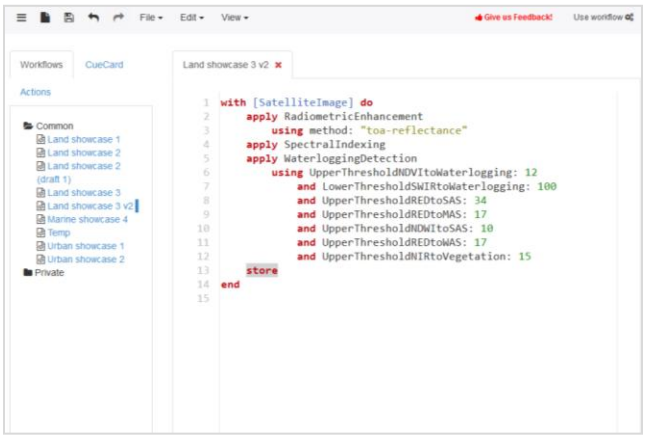

Figure 3: IQmulus workflow editor

On the figure the high-level DSL can be red for the Land showcase 3 which is a complex image processing workflow for waterlogging detection. The workflow implementation was originally done by GIS-experts and formulated using the DSL (which can be considered as in other solutions graphical modelers of a geo-processing workflow creation). Language designers have been analyzing the user requirements documentation, collecting all the keywords (nouns, verbs and adjectives), data types, service names, actors in order to create a language that meets user requirements in the respective domains.

Workflow designer is providing two user modes, for GIS-expert users and end-users. DSL user interface for GIS-experts provides extended language elements giving more control for experts over data sets and services, whereas for end-users a limited, more compact and lightweight version of DSL have been provided [7]. In this approach there is possibility to avoid very technical parameterization considering the user's needs, although an expert user has right also to change the previously implemented metrics. In the future vision on the CueCards section all the required information are provided to support the usage of the Workflow Editor. After the workflow is selected and edited by the user and also the dataset is selected for the appropriate workflow the process need to be executed. The process is goes through on three main steps until it reaches the services and the data in the cloud. The parser, the interpreter and the job manager are the three main components.

2.4 Visualization

Visualization techniques are always playing important role to communicate the results to the end-users of a project. Even in the geospatial world we can consider the applied visualization support have more efficient impact in the dissemination of the results. However, in the traditional GIS systems the next generation graphical solutions are not implemented. IQmulus project stated the goal to "leverage the power of modern graphics processing units (GPU) technology" in an application of the GIS domain. The targeted user groups of the project defined in two levels: expert users and decision makers/non-experts. In the implementation it has been also considered, Fat Client (IFC) visualization software as the playground of the experts and the Thin Client as the interactive visualization support for non expert users. The architecture of the IQmulus visualization can be seen on the figure 4. On the figure the commutation between the implemented modules of the project are also visible.

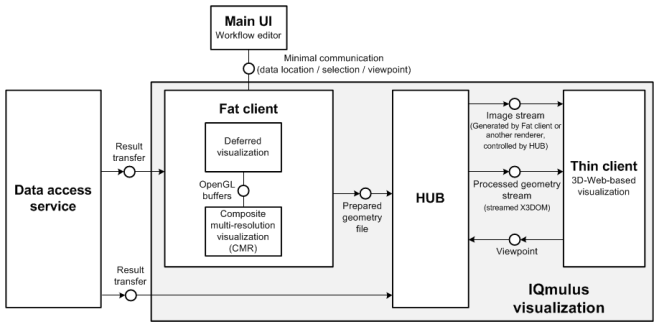

Figure 4: Architecture of the IQmulus visualization [12]

The Thin client is a web based 3D viewer for different geospatial data which are stored in the IQmulus cloud. The accesses of the results are realized by the automated visualization adaptation framework (HUB) which is in the implementation period.

The Fat client (IFC) is a desktop solution for the visualization of the results of several workflows. As it can be seen from the figure 4 the functions are going to be embedded and the users can reach each of the components through the main user interface. All the components of the whole system are acting like a black box for the end users but also provide access to certain parameters to serve the expert users needs as well. The client is an internally 3D rendering system with basic navigation mapping functions developed for the specific purposes of the IQmulus project. On the figure 5 the graphical interface of IFC is displayed with one resulting GeoTIFF from the flood and waterlogging detection workflow. The colors of the maps are representing the waterlogging classes like natural water surfaces, waterlog, seriously affected soil, moderately affected soil, weakly affected soil, vegetation in waterlog.

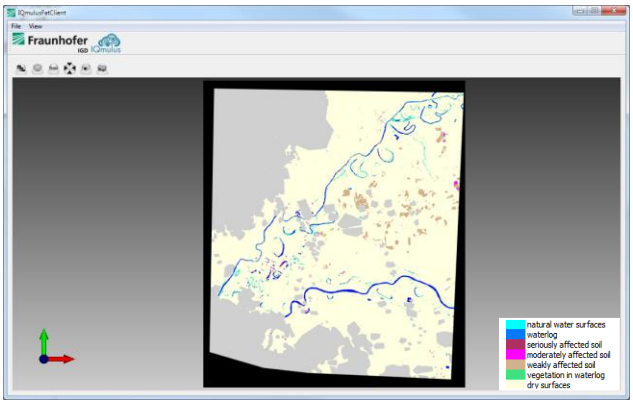

Figure 5: Interface of IQmulus Fat Client (IFC) displaying the GeoTIFF result of the LS3 workflow

4. FLOOD AND WATERLOGGING DETECTION: OPERATIONAL RESULTS

One of the case studies selected for early implementation within IQmulus focuses on the solutions for Rapid Response and Territorial Management. More specifically, we concentrate on the developments and results related to preprocessing and classification of satellite images for the detection of flooded and waterlogged areas. The original, currently used flood detection method developed by FÖMI consists of several steps including preprocessing (geometric transformations, cloud and cloud shadow filtering, radiometric calibration, calculation of spectral indices), feature detection (rule-based classification to provide thematic maps with several categories of water presence).
IQmulus plays a major role in increasing the degree of automation of this workflow. Detecting flood and waterlogging is a task that requires a series of satellite images of acquired for different areas, but almost at the same time. Quick response in such a situation is critical, thus the processing time of the satellite and/or aerial image series in a given time is of major importance. Original data needs calibration; other data has to be derived from the original (in this case spectral indices), and the images could be acquired by different sensors.

Based on the above needs, the flood and waterlogging detection workflow has been developed in the frame of the IQmulus project. It is a complex workflow covering all the aspects mentioned above. It consists of multiple algorithms (some of which are also available as separate services). Compared to the current solution, it provides a higher level of automation via smarter algorithms; therefore, it improves overall processing time and implies a better use of human resources.

Radiometric pre-processing (TOA reflectance calculation) and processing of spectral indices is now based on metadata files of satellite imagery, leading to an automatic instead of a manually induces process. Calculation of spectral indices needed for thematic classification is also based on metadata stored along the images. The whole process can be described and parameterized in Domain Specific Language in the Workflow Editor, and after the selection of datasets to be used it can be launched directly from the IQmulus Graphical User Interface (Figure 6). Results are created in the cloud and can be downloaded for further processing and analysis.

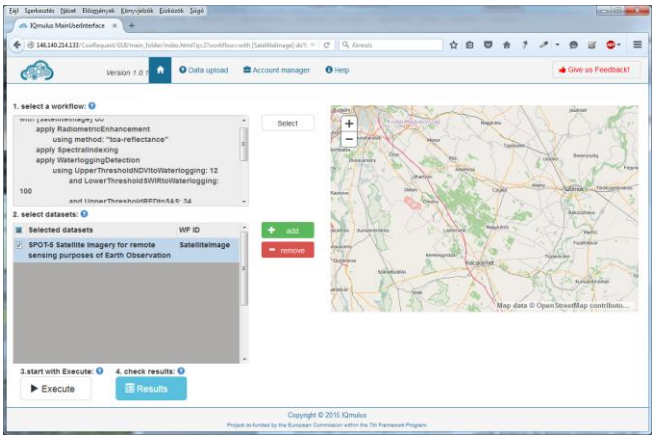

Figure 6: Workflow execution and data selection on the IQmulus graphical user interface

5. ACKNOWLEDGEMENTS

IQmulus (full name: A High-volume Fusion and Analysis Platform for Geospatial Point Clouds, Coverages and Volumetric Data Sets) is a 4-year Integrating Project (IP) partially funded by the European Commission under the Grant Agreement FP7-ICT-2011-318787. It is positioned in the area of Intelligent Information Management within the ICT 2011.4.4 Challenge 4: Technologies for Digital Content and Languages. IQmulus started on November 1, 2012, and will finish on October 31, 2016.

6. REFERENCES

[1] A. Olasz and B. Nguyen Thai, Decision support on distributed computing environment (IQmulus). OGRS 2014. Proceedings of the 3rd Open Source Geospatial Research & Education Symposium 2014. 06.10-13. pp.107-114.

[2] M. R. Ghazi and D. Gangodkar, Hadoop, Map/Reduce and HDFS: A Developers Perspective Procedia Computer Science, Volume 48, 2015, pp. 45-50.

[3] The Apache Software Foundation, "Apache Hadoop," Wiki 2007. Available at http://wiki.apache.org/hadoop/ [Nov 02, 2015].

[4] G. Wang and Q.Weng, 2014. Remote Sensing of Natural Resources. Taylor and Francis Ltd.,USA. pp 25.

[5] B. Nguyen Thai and A. Olasz, Raster data partitioning for support distributed GIS processing. ISPRS Archives Volume XL-3/W3, 2015 pp.543-551

[6] M. Krämer and I. Senner, A modular software architecture for processing of big geospatial data in the cloud, Computers & Graphics, vol. 49, pp. 69–81, 2015.

[7] D. Kristóf, R. Giachetta, A. Olasz, B. Nguyen Thai, Big geospatial data processing and analysis developments in the IQmulus project; Proceedings of the 2014 Conference on Big Data from Space (BiDS'14) pp. 214-217.

[8] R. Giachetta, AEGIS A framework for processing large scale geospatial and remote sensing data in MapReduce environment ,Computers & Graphics Volume 49, 2015, pp. 37–46

[9] J. R. Herring (ed.) The OpenGIS Implementation Standard for Geographic Information: Simple Feature Access – Common Architecture, version 1.2.1. 2011 Open Geospatial Consortium Inc., OpenGIS Project Document Number 06-103r4, Wayland, MA, USA

[10] N. Golpayegani and M. Halem, "Cloud Computing for Satellite Data Processing on High End Compute Clusters", IEEE International Conference on Cloud Computing, 2009 (CLOUD '09), IEEE, New York, NY, USA, pp. 88–92

[11] A. Cary, Z. Sun, V. Hristidis and N. Rishe, "Experiences on Processing Spatial Data with MapReduce", Lecture Notes in Computer Science, Springer-Verlag, Berlin, Heidelberg, Germany, 5566, 2009 pp. 302–319

[12] F.Michel, T. Franke, T. Gierlinger, A. Brodtkorb Interactive visual decision support techniques, Deliverable D5.1.1 IQmulus project documentation 2013.