CORRIGENDUM

The following paper has to be considered as a replacement of the corresponding abstract that the Editors wrongly sent for publication to the Copernicus Service Provider. The Editors apologize with the Authors for the inconvenience.

10 August 2017

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W2-137-2017-corrigendum

PROCESSING BIG REMOTE SENSING DATA FOR FAST FLOOD DETECTION IN A DISTRIBUTED COMPUTING ENVIRONMENT

A. Olasz [a]*, D. Kristóf [a], B. Nguyen Thai [b], M. Belényesi[a], R. Giachetta[b]

[a] Dept. of Geodesy, Remote Sensing and Land Administration, Government Office of the Capital City Budapest, 5. Bosnyák sqr. Budapest, 1149 Hungary (olasz.angela, kristof.daniel, belenyesi.marta)@bfkh.gov.hu
[b] Dept. of Cartography and Geoinformatics, Eötvös Loránd University (ELTE), 1/A Pázmány Péter sétány, Budapest, 1117 Hungary, (ntb, groberto)@inf.elte.hu

Commission IV, WG IV/4.

KEY WORDS: Distributed Computing, Geospatial Big Data, Cloud Computing, Flood detection, Big Earth Observation Data

ABSTRACT

The Earth observation (EO) missions of the space agencies and space industry (ESA, NASA, international, national and commercial companies) are evolving as never before. These missions aim to develop and launch next-generation series of satellites and sensors providing huge amounts of data, even free of charge, to enable novel monitoring services. The CEOS's Earth Observation handbook (Petiteville et al., 2015) emphasises the role of Earth Observation Data in risk reduction on global and local level. The wide geospatial sector is targeted to handle new challenges to store, process and visualize these geospatial data, reaching the level of Big Data by their volume, variety, velocity, veracity, along with the need of multi-source spatio-temporal geospatial data processing. Nevertheless, handling and analysis of remote sensing data has always been a cumbersome task due to the ever-increasing size, heterogeneity and frequency of collected information. This paper presents the achievements of the IQmulus EU FP7 research and development project with respect to processing and analysis of geospatial big data in the context of flood and waterlogging detection.

1 INTRODUCTION

Big Data era has arrived to the geospatial sector in recent years. The available data amounts are exploding due to transitions towards free and open data policies (EU 0pen data, 2016) including opening the archives of satellite imagery (M. Miller, 2016), national geospatial databases, initiatives to create international spatial data infrastructures (e.g. INSPIRE, Council of the European Union, 2007), and last but not least, the new previously mentioned new era of EO missions (Petiteville et al., 2015). The "Big Data" term is often used to refer solely to the volume of data although it incorporates further important aspects. Hence, the widely-known "V"s (volume, velocity, variety, etc.) give more insight into data and underlying processes (Kambatla et al., 2014). In mainstream IT, the term Big Data represents not only the features but also the way how the valuable information is derived (Manyika et al., 2011). In the geospatial sector we also have to the tackle the increase in amounts of data but the efficient methodologies for data processing and information extraction (technological background, out-of-the-box solutions, etc.) are still under heavy development (Olasz et al., 2016; Igor Ivan et al., 2017). Huge datasets are available but up to now there are only limited automatic procedures for processing; thus, as a huge amount of data remains unprocessed, a wealth of information is latent in many datasets in different locations. Although a number of distributed data processing solutions exist in the mainstream IT sector, those have primarily been developed by focusing on processing simple data structures (e.g. documents) rather than complex geospatial data (Agrawal et al., 2012). Several well-established definitions exist to describe and characterise Geospatial Big Data (Lee and Kang, 2015, Li et al., 2015, Olasz et al., 2016) . Cloud computing is considered as a key factor in Big Data research, providing a solid architectural basis for operational solutions. It enables fundamental features (Yang et al., 2017) from the low level of infrastructures up to advanced and ready-to-use flexible environments. Yang et al. also admits that Big Data and spatiotemporal thinking drive the advancement of cloud computing with new requirements (Yang et al., 2017). We share this opinion when concluding our experience gathered during the IQmulus Project (The IQmulus Consortium, 2013). This paper presents a multi-sensor remote sensing processing solution to realize fast flood and waterlogging detection based on the IQmulus platform, a cloud-based geospatial big data processing environment.

2 THE IQMULUS PROJECT

The four-year IQmulus project (http://www.iqmulus.eu), targeted the optimized use of large, heterogeneous geospatial datasets ("Geospatial Big Data") for better decision making through a high-volume fusion and analysis information management platform. The consortium was made 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. Started in October 2012 and ended in October 2016.
An end-to-end involvement of users was ensured through the implementation of three 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 were achieved. 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 (The IQmulus Consortium, 2013), in which IQmulus can yield significant improvements. Services consist of algorithms and workflows focusing on the following aspects: algorithms and

spatio-temporal data fusion, feature extraction, classification and correlation, multivariate surface generation, change detection and dynamics. System integration and testing were carried out in the last phase of the project. The IQmulus platform was 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. In this study we are focusing on the Land Application for Rapid Response test bed, which includes the showcase of Detection and characterization of flood and waterlogging. This paper documents the latest results in terms of processing services and system architecture.

3 IMPLEMENTATION DETAILS

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 via cloud processing. However, to take full advantage of this technology, existing data processing methodologies and workflows have to be revisited and redesigned.
IQmulus developed functional and domain processing services in order to maximize the use of geospatial big data and provide support for analysing quickly changing environmental conditions. The platform (together with a web user interface) supports the processing of point clouds, vector and raster data (especially remotely sensed images) using a wide variety of algorithms. During development, specific emphasis was put on using open source components and solutions.

3.1 Distributed File System (HDFS) and Distributed Processing (MapReduce)

Hadoop relies on two major components, a distributed file system (HDFS) and MapReduce (Dean and Ghemawat, 2004) distributed algorithm. Hadoop distributed file system have been designed to store text-based data (Borthakur, 2013). 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.
Originally, HDFS was 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 had to support a wide range of runtime environments.

3.2 The AEGIS framework

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, initially developed by Eötvös Loránd University (ELTE), Faculty of Informatics. It has been adopted to the needs of IQmulus processing services by cooperation with Institute of Geodesy, Cartography and Remote Sensing (FÖMI).
The AEGIS geospatial framework was initially developed for education and research goals, and is currently used as a learning tool for computer science students at ELTE. It is based on well-known standards and state of the art programming methodologies. It has been developed by taking adaptability and extensibility in mind. The component based infrastructure enables the separation of working fields, and the interchangeability of data models, methods and algorithms.
AEGIS supports both vector data and remotely sensed images. Based on the OGC Simple Feature Access standard (Herring, 2011) all spatial data is considered as a form of geometry. Multiple realizations of the abstract geometries are enabled by using the abstract factory pattern and inversion of control containers (Cooper, 2010) is utilized to handle multiple factories. To enable support for remotely sensed imagery, spectral geometries containing raster datasets are introduced as a subtype of geometry. Rasters can also be transformed to topology graph representation and combined with vector data (Fowler, 2004). The processing module contains the processing algorithms and the execution environment. Algorithms are objects described using a meta descriptor system, which is based on the Identified Object scheme, and is a generalization of the coordinate operation model of the OGC Spatial Referencing by Coordinates standard (Winter and Frank, 2000).
The execution environment deals with monitoring and cataloguing methods and operations based on the meta descriptors. This metadata enables the environment to validate and optimize execution of the method. The heart of the environment is the operations engine, which is responsible for executing operations.
As both data and processing models are extendible, the addition of support for the Hadoop environment could be implemented as an extension to the framework. However, the implementation required multiple considerations with respect to data management and operation execution.

3.3 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 generic spatial file formats (such as Shapefile, LAS, or GeoTIFF) to enable the direct consumption by most geospatial toolkits. 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.
Hence, 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.

3.5 MapReduce-based Data Processing

Operations in Hadoop are performed as MapReduce processes consisting of two phases, whilst libraries provide single phase

operations. However, the MapReduce paradigm can also be considered as the Map phase performing the initial operation, whilst the Reduce phase is responsible for post-processing the results (Cary et al., 2009, Golpayegani and Halem, 2009). 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 elementwise, thus when processing partial images the results are independently valid without the need of any merging. In other cases a simple aggregation function can be executed.
The proper post-processing method can be chosen based on the operation metadata.
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. The overview of the operation can be seen in Figure 1.

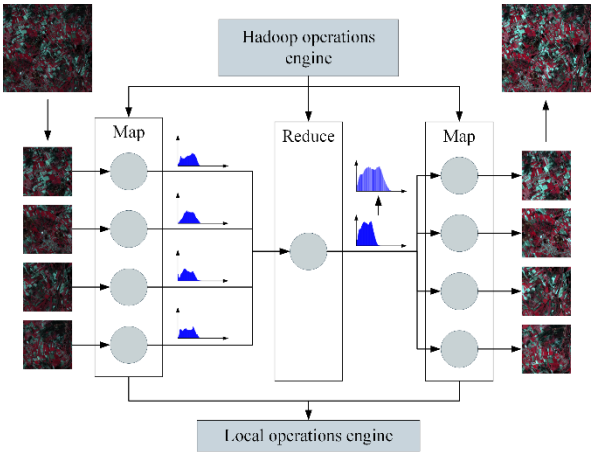

Figure 1: Histogram equalization performed on a partitioned image

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. The overview of the system can be seen in Figure 2.

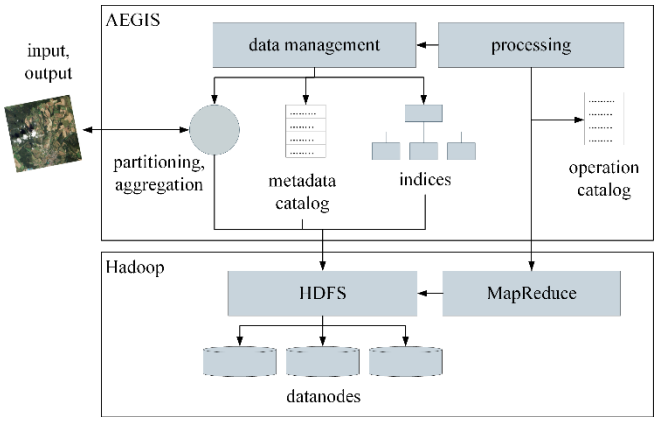

Figure 2: Overview of the spatial data processing framework on Hadoop

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, after the selection of datasets; it can be launched directly from the IQmulus Graphical User Interface (Figure 3). Results are created in the cloud and can be downloaded for further processing and analysis. Interactive visualization solutions are also developed.

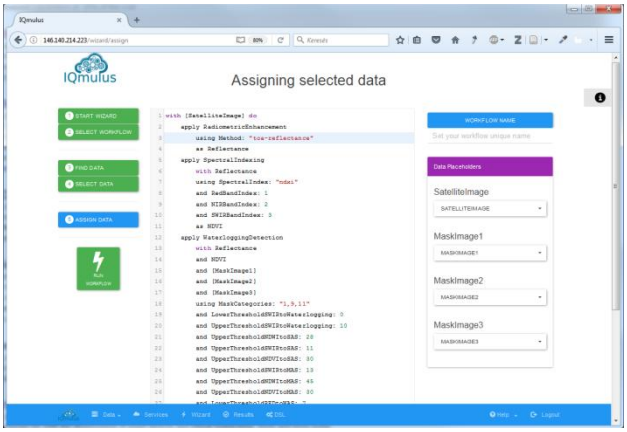

Figure 3: Workflow creation, data selection and execution on the IQmulus graphical user interface

5 RESULTS AND CONCLUSION

Our presentation will focus on the solutions on the developments and results related to preprocessing and classification of satellite images for the detection of flooded and waterlogged areas. The developed solution provides a higher level of automation via smarter algorithms supporting the process of multi-sensor remote sensing resources (SPOT, Landsat, and Sentinel 2). Our presentation will include description of the newly available operative processing workflow.

ACKNOWLEDGEMENTS

This research is co-funded by the project "IQmulus" (A High-volume Fusion and Analysis Platform for Geospatial Point Clouds, Coverages and Volumetric Data Sets) funded from the 7th Framework Programme of the European Commission, call identifier FP7-ICT-2011-8.

REFERENCES

Agrawal, D., P. Bernstein, E. Bertino, S. Davidson, U. Dayal and M. Franklin (2012). *Challenges and Opportunities with Big Data*.

Borthakur, Dhruba (2013). *Hadoop Distributed File System Architecture Guide*, at https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html, [accessed 24 March 2017].

Cary, A., Z. Sun, V. Hristidis and N. Rishe (2009). Experiences on Processing Spatial Data with MapReduce, *Scientific and Statistical Database Management*, at https://link.springer.com/chapter/10.1007/978-3-642-02279-1_24, [accessed 26 May 2017].

Cooper, P. (2010). The OpenGIS Abstract Specification - Topic 2: *Spatial referencing by coordinates*.

Council of the European Union (2007). Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE).

Dean, J. and S. Ghemawat (2004). MapReduce: Simplifed Data Processing on Large Clusters, *Proceedings of the Sixth Symposium on Operating Systems Design & Implementation* (OSDI'04) December 6-8, 2004 San Francisco CA, USA, : 137–150.

EU 0pen data (2016). *EU 0pen data, The basics for EU data providers*, Luxembourg: Publications Office of the European Union.

Fowler, M. (2004). Inversion of control containers and the dependency injection pattern.

Golpayegani, N. and M. Halem (2009). Cloud Computing for Satellite Data Processing on High End Compute Clusters, *Proceedings of IEEE International Conference on Cloud Computing*, pp. 88–92.

Herring, J.R. (2011). OpenGIS Implementation Standard for Geographic Information: Simple Feature Access – Common Architecture.

Ivan I., A. Singleton, J. Horák and T. Inspektor (2017). *The Rise of Big Spatial Data*, Springer International Publishing.

Kambatla, K., G. Kollias, V. Kumar and A. Grama (2014). Trends in big data analytics, *Journal of Parallel and Distributed Computing*, 74(7): 2561–2573.

Lee, J.-G. and M. Kang (2015). Geospatial Big Data: Challenges and Opportunities, *Big Data Research*, 2(2): 74–81.

Li, S., S. Dragicevic, F. Anton, M. Sester, S. Winter, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein and T. Cheng (2015). Geospatial Big Data Handling Theory and Methods: A Review and Research Challenges, http://arxiv.org/abs/1511.03010, [accessed 13 April 2016].

M. Miller, H. (2016). Users and Uses of Landsat 8 Satellite Imagery, 2014 Survey Results, U.S. Geological Survey.

Manyika, J., M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh and A.H. Byers (2011). Big data: The next frontier for innovation, competition, and productivity, Report McKinsey & Company.

Olasz, A., B. Nguyen Thai and D. Kristóf (2016). A new initiative for Tiling, Stitching and Processing Geospatial Big Data in Distributed Computing Environments, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, Volume III-4: 111–118.

Petiteville, I., S. Ward, G. Dyke, M. Steventon and J. Harry (2015). Satellite Earth Observation in Support of Disaster Risk Reduction, *CEOS Earth Observation Handbook, 3th UN World*

Conference on Disaster Risk Reduction: European Space Agency.

The IQmulus Consortium (2013). State of the art analysis of the IQmulus Project. http://www.iqmulus.eu [accessed 24 March 2017].

Winter, S. and A.U. Frank (2000). Topology in Raster and Vector Representation, *GeoInformatica*, 4(1): 35–65.

Yang, C., Q. Huang, Z. Li, K. Liu and F. Hu (2017). Big Data and cloud computing: innovation opportunities and challenges, *International Journal of Digital Earth*, 10(1): 13–53.