Angéla Olasz
Department of Geoinformation, Institute of Geodesy, Cartography and Remote Sensing (FÖMI),5. Bosnyák sqr. Budapest, Hungary, olasz.angela@fomi.hu,

Binh Nguyen Thai
Department of Cartography and Geoinformatics, Eötvös Loránd University (ELTE), 1/A Pázmány Péter sétány, Budapest, Hungary, ntb@inf.elte.hu

Corresponding Author
Angéla Olasz
olasz.angela@fomi.hu

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2226v1 | CC BY 4.0 Open Access | rec: 4 Jul 2016, publ: 4 Jul 2016

Geospatial Big Data processing in an open source distributed computing environment

Abstract

In recent years, distributed computing has reached many areas of computer science including geographic and remote sensing information systems. However, distributed data processing solutions have primarily been focused on processing simple structured documents, rather than complex geospatial data. Hence, migrating current algorithms and data management to a distributed processing environment may require a great deal of effort. In data processing, different aspects are to be considered such as speed, precision or timeliness. All depending on data types and processing methods. Available data volume and variety evolving as never before which instantly exceeding the capabilities of traditional algorithm performance and hardware environment in the aspect of data storage, management and computation. Augmented efficiency is required to exploit the available information derived from Geospatial Big Data. Most of the current distributed computing frameworks have important limitations on transparent and flexible control on processing (and/or storage) nodes. Hence, this paper presents a prototype for distribution ("tiling"), aggregation ("stitching") and processing of Big Geospatial Data focusing the distribution and processing of raster data type. Furthermore, we introduce an own data and metadata catalogue enables to store the "lifecycle" of datasets, accessible for users and processes. The data distribution framework has no limitations on programming environment and can execute scripts (and workflows) written in different language (e.g. Python, R or C#). It is capable of processing raster, vector and point cloud data allowing full control of data distribution and processing. In this paper, the IQLib concept (https://github.com/posseidon/IQLib/) and background of practical realization as a prototype is presented, formulated within the IQmulus EU FP7 research and development project (http://www.iqmulus.eu). Further investigations on algorithmic and implementation details are in focus for the oral presentation.

Background

Geospatial data (also known as spatial data, geo-information, geodata, etc.) have many definitions depending from the background of the author. All of them agreed on a unique characteristic are the geographic location of the phenomena to be described as basic criteria. The nature of the digital representation of the continuous space can be grouped in 4 or 5 format. Traditionally we consider two format of geospatial data: vector and raster (Elek, 2006) owing to the development of information technology and approximation of the geospatial technology to the mainstream information technology the nowadays we can have higher abstraction representation of data such as point clouds, graph networks, etc. An additional particular kind of location-aware data is also examined by analysts; social media-like data which requires a particular approach to collect and process as well. Along with Big Data theory geospatial big data is defined as volume, variety and update frequency rate that exceed the capability of spatial computing technology (Lee and Kang, 2015, Li et al., 2015, Kambatla et al., 2014). In the following table (Table 1.) we tried to illustrate the difference between "Big Data", "Geospatial Big Data" and "Geospatial Data" to clarify what we consider under these designations. Others may introduce numerical limitations for those categories with the million instructions per second (MIPS) to be able to process with those systems. We would avoid find exact limits to separate them. Especially, we think the category margins are permeable depending on quite valuable number of variables such as: the amount, the variety of the processed data, the actual use case, extent and data density and of course what are the parameters of the actual processing capabilities of hardware and software environment so we admit those margins are fuzzier border zones. Actually the content of the table is valid for a user with average computational capability environment for 2016.

		Big Data	Geospatial Big Data	Geospatial Data
Representation	raster	photos, graphics, security surveillance images, traffic sensor images, medical records (x-ray, retinal scans, fingerprints, etc.)	time-series of satellite images, orthophotos global, country-wide, regional, local coverage, required by space borne, airborne or UAs, global topography data, etc.	thematic cartographic maps, topographical maps, orthophotos, satellite images, hyperspectral images, DEMs in raster format
	vector	2D, 3D graphics in vector format	global land cover, earth observation data, environmental data, national cadastral data, watercourse, utility and transportation network with its attributes etc.	national, regional, local administrative data, earth observation data, socioeconomic data, environmental data
	point cloud	3D scans of objects (in robotics, medical, automotive, art, archaeology, geology, etc.)	terrestrial MMS data, LiDAR data,	classified, filtered point clouds (subset of LiDAR or MMS data)
	text based	social network, comments, text messages, business data, logs, administrative data, sensors text data, transportation/travel/trade data, life science data	text based big data complemented with geolocation, geosocial data, (coordinates, address, geographical names, etc.)	track logs, coordinates, attributes, indices stored in text files

Storage and processing background	raster	Wide column store, Distributed file system,	Array database/Key-value store, RDD, wide column store	OB-RDBMS with extension to raster or traditional file based image storage processing software
	vector	Relational DBMS, Wide column store	Distributed file system, relational DBMS complemented with Spatial extensions, or wide column store and key-value store with GIS functions	OB-RDBMS
	point cloud	Key-value store	Key-value store, RDD	OB-RDBMS with extension to point cloud storage and processing or conventional software solutions
	text based	Distributed file system, Document Store DBMS, Wide-column store,	Distributed file system, Document Store DBMS, Wide-column store, RDD	conventional GIS processing applied (often with format conversion)
Common solutions	raster	Apache Accumulo, Cloudera	Rasdaman, SciDB, GeoTrellis, GeoMesa, Geowave operating on the top of different DB-engines	Grass GIS, Saga GIS, Orfeo, OSSIM, gvSIG, QGIS, PostGIS Raster etc.
	vector	Cassandra, HBase,Distributed file system	Apache Hadoop, Hive, HBase, Accumulo, MongoDB, Neo4j with extension for spatial functions and existing libraries (e.g.MapR)	PostGIS, SpatiaLite, MySQL, QGIS,
	point cloud	Distributed file system	Apache Spark with extension for spatial functions (e.g.Spark Lidar)	PostGIS, LasTools, rLiDAR, Geo-Plus, Grass GIS- LiDAR Tools
	text based	Cassandra, Cloudera, HBase, Neo4j, CouchDB, MongoDB, Hortonworks, MillWheel	Apache Storm, S4, Spark, Hive	Desktop software (GPS tracklog processing, etc.)

Table 1.: Characteristics of Big Data, Geospatial Big Data and Geospatial Data with common solutions

Defining distributed geospatial computing or processing is also a challenge. The Encyclopedia of GIS (Phil Yang 2008,) defines distributed geospatial computing (DGC) as “geospatial computing that resides on multiple computers connected through computer networks”. So “geospatial computing communicates through wrapping applications, such as web server and web browser”. To translate it, distributed geospatial computing is when geoprocessing is done within a distributed computing environment. In the Handbook of Geoinformatics (2009) Yu et al. focus on the multi-agent system with ontology to perform distributed processing of geospatial data. The distributed processing of geospatial data is continuously evolving together with the evaluation of computer networks. A single milestone we would like to emphasize from the evaluation progress is when Google Earth was issued in 2004; by reason of it caused a life changing effect on the citizens’ everyday life and made popular geospatial applications. Furthermore, until nowadays Google’s solutions are leading in the process of massive dataset along with the development of easy-to-use interface (e.g., Google BigTable) and play an important role in the open source community developments. Consequently, distributed systems supports heterogeneous network and infrastructural background, cloud solutions have been developed to exploit the advantages of distributed systems and made available services for geospatial computing as well.

Method

In previous work we have made a feasibility study on technological and conceptual aspects. The outcome was presented in our previous paper (Nguyen Thai and Olasz, 2015), where we have described the architecture of this demo application as well as processing results on calculating NDVI index using Landsat8 satellite images for the territory of Hungary. The processing results seemed really convincing, so we have started to design and implement IQLib framework. This framework should be able store metadata information on our dataset, tracking partitioned data, their location, partitioning method. It should distribute data to processing nodes as well as deploying existing processing services on processing nodes and execute them in parallel.
As a result IQLib has three major modules; each module is responsible for a major step in GIS data processing. The first module is called Data Catalogue, second module is Tiling and Stitching, the last module is called Distributed Processing module.
Data Catalogue module is responsible for storing metadata corresponding to survey areas. A survey area contains all the dataset that are logically related to inspection area, regardless of their data format and purpose of usage. We would like to store all the available, known and useful information on those data for processing.
Tiling and Stitching module does exactly what its name defines. Usually tiling algorithms are performed on raw datasets before running a specific processing service on given data. Stitching usually runs after processing services have successfully done their job. Tiling algorithms usually process raw data, after these tiled data are distributed across processing nodes by data distribution component. Metadata of tiled dataset are registered into Data Catalogue. With this step we always know the parents of tiled data.
Distributed Processing module is responsible for running processing services on tiled dataset.

Conclusions

In this paper we study the distributed system design to handle big geospatial data. We found this area of research is very active in terms of review, development and usage of the existing solutions with a continuous implementation for specific use cases and application areas such as disaster management environmental monitoring, earth observation data analysis and distribution, etc. We have attempted to compare Big Data, Geospatial Big Data and Geospatial Data to clarify the possible features of differences, compare them in the term of storage and processing background for different data representation and tried to collect and categorized the existing common system solutions. The second part of the paper provided overview about our previous work and implementation the new framework called IQLib. We have introduced the 3 main modules of the system and described in brief their technical realization. Data Catalogue module is on higher level of preparation, hence we could provide details on the data model and data decomposition.
IQLib documentation on data model and data catalogue is available on Github at https://github.com/posseidon/iqlib.
We have decided not to publish Data Catalogue module's source code until it has been reviewed and finalized by IQmulus project partners. However, the RESTFUL API is available on Heroku cloud infrastructure for all project partners to test and give feedbacks and suggestions at http://iqlib.herokuapp.com. In our future work we are going to focus on further development of the framework along with the processing executables and experimental benchmarking of processing time. Next development phase is the implementation of Data Catalogue module and installation for testing phase. Future directions are the Processing module to be implemented together with GIS raster processing scripts for further analysis of open geodata (such as Sentinel 2 satellite imagery).

Acknowledgements

The research was also supported by the Hungarian Institute of Geodesy, Cartography and Remote Sensing (FÖMI). We would like to thank Michela Spagnuolo research director of Institute for Applied Mathematics and Information Technologies (CNR-IMATI) for her huge effort on making opportunities as well as support on IQLib.

References

Elek I. 2006. Bevezetés a geoinformatikába (Introduction to geoinformatics) ELTE Eötvös Kiadó, Budapest, pp. 22-40,

F. Chang et.al.: 2006. "Bigtable: A Distributed Storage System for Structured Data." OSDI'06: Seventh Symposium on Operating System Design and Implementation, Seattle, WA, November, 2006.

Jewell D. et al. 2014. IBM RedBook, Performance and Capacity Implications for Big Data, IBM Corp. pp. 7-20

Kambatla K., Giorgos K., Vipin K., and Ananth G. 2014. Trends in Big Data Analytics. Journal of Parallel and Distributed Computing 74 (7) pp. 2561–2573.

Karimi H. A. 2014. Big Data Techniques and Technologies in Geoinformatics. Taylor & Francis Group, LLC, pp. 149-153.

Lee J.-G. and Kang M. 2015. Geospatial Big Data: Challenges and Opportunities, Big Data Research, vol. 2, no. 2, pp. 74–81.

Li S. , Dragicevic S., Anton F., M. Sester, S. Winter, A. Coltekin, C. Pettit, B. Jiang, J. Haworth, A. Stein, and Cheng T. 2015. Geospatial Big Data Handling Theory and Methods: A Review and Research Challenges, pp. 2-19.

S. Shekar, and H. Xiong Encyclopedia of GIS. Springer. 2008. pp 550-558.

Nguyen Thai B. and Olasz A. 2015. Raster data partitioning for support distributed GIS processing, Proceedings of ISPRS.Vol. XL-3/W3, pp. 543-550.

Olasz A. and Nguyen Thai B. 2016. A new initiative for Tiling, Stitching and Processing Geospatial Big Data in distributed computing environment, ISPRS Annals of XXII ISPRS Congress 2016, pp.