

FOG COMPUTING PERSPECTIVES IN CONNECTION WITH THE CURRENT GEOSPATIAL STANDARDS

E. Panidi

Department of Cartography and Geoinformatics, Saint Petersburg State University, St. Petersburg, Russia – panidi@ya.ru,
e.panidi@spbu.ru

KEY WORDS: Open Source Standards for Geospatial, Geospatial Fog Computing, Geospatial Web Services, Geospatial Data Interoperability

ABSTRACT:

Cloud Computing technologies and cloud-based Geographic Information Systems have become widely used in recent decades. However, the complexity and size of geospatial datasets remains growing and sometimes become going out of the cloud infrastructure paradigm. Additionally, many of currently used client devices have sufficient computational resources to store and process some amounts of data directly. Consequently, multilevel management techniques are demanded that support capabilities of horizontal (client-to-client) data flows in addition to vertical (cloud-to-client) data flows. These tendencies in information technologies (in general) have led to the appearance of Fog Computing paradigm that extends a cloud infrastructure with the computational resources of client devices and implements client-side data storage, management and interchange. This position paper summarizes and discusses mentioned tendencies in connection with a number of available Open Geospatial Consortium standards. The paper highlights the standards, which can be recognized as the platform for the Fog Computing implementation into geospatial domain, and analyzing their strong and weak features from the Fog Computing point of view. The analysis is built upon author's experience in implementation of the client-side geospatial Web services.

1. INTRODUCTION

Fog Computing paradigm (FC) was proposed in 2012 (Bonomi et al., 2012). Currently, the paradigm is formalized sufficiently to be applied to different domain areas. Next needed step is the design and development of its applications and corresponding software prototypes.

However, the domain of geospatial data management is not mentioned usually in publications as the one of possible application areas for the FC. Nevertheless, processes of collection, storage and processing of the geospatial data are implemented in many cases in a geographically distributed and geographically referenced manner. This feature underlines great relevance of the FC paradigm to the geospatial domain, while FC is focused also on design of the geographically distributed information systems.

The FC paradigm is based upon the idea of enabling of the client devices as a part of Cloud Computing (CC) hardware infrastructures, which are widely used today in many domains including storage and management of the geospatial data. FC architecture assumes the possibilities of data storage and processing on client devices or on the cloud core (processing center or data center), as well as data interchange between client devices and cloud core, or between client devices directly, selecting approach that is most relevant to the thematic domain (OpenFog Consortium Architecture Working Group, 2016, 2017). Such a flexible architecture demanded for example when collecting data generated by users (commonly used term in this case is Volunteered Geographic Information), or in the systems of monitoring of the natural objects (Panidi, 2016).

2. CURRENT STANDARDS

Most important issues in the context of FC implementation into the geospatial domain are the improvement of existing and development of lacking technologies needed for distributed geospatial data storage, processing and interchange on client devices (Panidi, 2016). Other important tasks are the development and standardization of data models and data formats applicable for distributed data storage, as well as interfaces for direct client-to-client data interchange.

2.1 Interface Standards

Cloud-based geographic information systems (GISs) are implemented widely. Significant number of geospatial standards establish rules and restrictions for cloud-based GISs and geospatial services, first of all the OGC standards (Open Geospatial Consortium – <http://www.opengeospatial.org/>). The tasks of data models, formats and data interchange interfaces development for geospatial FC include also the issues of ensuring of the backward compatibility with existing cloud-based GISs and services.

A range of OGC standards composed particularly of six standards designed to unify interfaces (and partially architecture) of geospatial Web services, which provide access to geospatial data stored on the cloud (or on dedicated server) storage (Open Geospatial Consortium, 2006, 2010a, 2010b, 2014b, 2015, 2016):

- Web Map Service (WMS) provides access to rendered raster maps or other georeferenced raster imagery (e.g., satellite imagery)
- Web Map Tile Service (WMTS) also provides access to raster maps, but implies a preliminary splitting of the raster grid into the tiles (i.e., fragments, usually

squares) and the transmission of the image as a set of tiles

- Web Feature Service (WFS) provides access to vector datasets (in general case, only for vector layers without description of the graphic design)
- Web Coverage Service (WCS) provides access to raster coverages (so-called scientific raster datasets, which store quantitative characteristics of the area or qualitative codes of the area elements)
- Web Processing Service (WPS) provides access to the data processing algorithms that can be executed on the server side (in fact, the client gets links to spatial data processed on the server side in accordance with the initial request)
- Catalogue Service (CS) provides access to catalogs of geospatial metadata

OGC Web services use the HTTP (HyperText Transfer Protocol) to send and receive requests and responses, which is the standardized communication protocol in the Web. The client software can be implemented in the form of a Web page (HTTP requests are executed by a Web browser) or in the form of a standalone desktop application (a capability of HTTP requests execution have to be implemented directly in the application, in this case). On the server side, HTTP requests processed and responded by general-purpose software Web server, which passes them to the corresponding server-side Web service software.

Abovementioned standards establish a number of mandatory and optional request types for corresponding Web services and their recipients (client applications). The list of request types supported by the service varies from standard to standard. However, in all cases the data access is preceded by transmitting and processing of a series of initial requests. The first in the series is the GetCapabilities request, which responded with the metadata of a Web service including a list of available datasets or processing tools (in the case of WPS). After obtaining of the metadata, a client usually can obtain metadata of particular dataset provided by the Web service, or obtain the dataset itself (in some cases part of the dataset, depending on the type of service).

This schema is well applicable when it is needed to implement new functionality and to ensure backward compatibility with already developed and deployed Web services in parallel. It is significant in the case of geospatial FC due to the wide range of currently used OGC-compliant cloud-based Web services. Abovementioned standards can be easily and effectively extended with new request as author's experience shows (Kazakov et al., 2015; Panidi et al., 2015).

2.2 Data Formats and Data Models

Different eXtensible Markup Language (XML) schemas are implemented also by OGC to define geospatial data models and storage formats and to provide data interchange across the information systems, including general-purpose GML (Open Geospatial Consortium, 2007a) and domain-specific SensorML (Open Geospatial Consortium, 2007b), WaterML (Open Geospatial Consortium, 2014a) and others. Application of XML implies human/machine readable data representation and gives the possibility of easy extending of the data models (also with backward compatibility ensuring) by implementing new XML tags.

It is important to underline, that distributed storage of geospatial data (in the case of FC) can demand data models and data management techniques, which enable datasets to be simultaneously big (aggregated from multiple client devices), distributed (stored partially on a number of different client devices), real-time (updated in real-time mode on different client devices), and multisource (composed of different available data segments in each time, when client devices are connected or disconnected). Such datasets can be denoted as geospatial hyperdatasets.

XML-based data storage can ensure support of hyperdataset characteristics through the mutual references in the separately stored segments of common dataset. However, from the storage point of view, the XML-based formats are not effective being the text format. In this way, a lack remains in the fog-based geospatial data storage, and additional investigations are needed.

3. CURRENT ISSUES

The issues of client-side geospatial data processing and client-to-client data interchange can be recognized as fundamental issues of FC implementation into geospatial domain.

3.1 Client-Side Data Processing

Contemporary cloud-based geospatial Web services assume server-side data processing when the client request is executed. For example, in the case of WCS GetCoverage request it can be needed to extract some limited set of the coverage layers and(or) to cut the coverage using requested extent. In this case, WCS server has to produce a cut operation immediately before uploading data onto the client. Similarly, the CS server has to interact with the database management system that operates metadata database when processing queries of metadata records.

The essence of the problem in the context of FC architecture consists in the necessity of implementation of the data management software suitable for running on client devices, which typically employ other types of operating systems and have limited computing resources (in comparison to cloud core).

Currently, in the case of Web-based publishing of the geospatial data on client devices it is needed to develop new software platforms to enable processing of geospatial data in the client operating environments (operating systems and hardware platforms). These software tools should provide the ability of deployment of geospatial services accordingly to the OGC standards. However, the amounts of data published in this way have to be consistent with computational facilities of client devices. Due to the limited computational resources of client devices, these services should provide access to the limited-size geospatial datasets (by analogy with the Big Data these datasets can be denoted as Small Geospatial Data). Such datasets generated in many cases by client devices itself (e.g., data collected by sensors of the client device).

A comprehensive solution to this problem is not proposed currently. An important aspect that has to be considered during design and development of such software platforms for client devices (including mobile devices) is the need of platform-independent execution of the software in different operating systems. The application also have to be deployed easily (Panidi, Efimov, 2015; Panidi et al., 2016).

As a possible solution in this context, JavaScript programming language technology can be mentioned. Program code created in this language can be executed in any Web browser (including mobile Web browsers), which makes it universal. For example, Geolib (<https://github.com/manuelbieh/Geolib>) and Turf (<http://turfjs.org/>) JavaScript software libraries can be referred as the GIS enabling tools.

3.2 Client-to-Client Data Interchange

The issue of client-to-client data interchange that appears when implementing fog-type geospatial Web service consists in the need of Web server deployment together with software of a Web service on the client device. This problem can be solved currently through the use of existing Web servers designed for mobile devices, for example: kWS (<https://play.google.com/store/apps/details?id=org.xeustechnologies.android.kws>) or Palapa (<https://play.google.com/store/apps/details?id=com.alfanla.android.pws>) for the Android operating system. However, this approach does not look universal, as IP address of the client device (which is used to address the Web server) is not static as a rule and consequently is inaccessible from the external networks. This may complicate or make impossible the access to the Web server and to the corresponding Web services via HTTP.

In connection with this feature of mobile devices, the implementation of geospatial FC demands alternative addressing technologies capable to ensure direct communication between client devices. WebRTC (W3C, 2017) technology can be denoted as a promising technology in this case. It allows to establish a direct real-time connection between devices (browser-to-server or browser-to-browser connection) through the HTTP or alternatively through the Web Sockets protocol, which provides more flexible options of data interchange.

To ensure the operation of WebRTC connection, the broker server is required that is used as the hand-shaking facility. This server is not used for the data transmitting, however the connection parameters for each of WebRTC nodes are provided via it. Such architecture does not contradict with the FC paradigm, which assumes the possibility of establishing of the cloud core as a control and management subsystem. It should be mentioned additionally, that connection and interaction with broker server can be provided through the implementation of extra types of requests for the OGC-compliant Web services, as it mentioned above.

4. CONCLUSIONS

Implementation of the FC principles in the domain of geospatial data management is very promising. Currently, most obvious application areas for geospatial FC are the distributed systems of environmental monitoring, and areas of collection and accumulation of data generated by users.

Significant issue is the need to ensure the integrity and backward compatibility with existing cloud-based systems and services. In particular, when designing and developing fog-based geospatial Web services it is needed to ensure the compatibility with existing OGC standards.

Two fundamental issues are presented currently on the way of compatibility ensuring with existing standards. The issues are related to the implementation of geospatial data processing on the host platforms of fog-based Web services, and to the need

of Web server use on the host platform to handle HTTP requests. Both issues are concerned with the use of client devices and especially of mobile devices as the host platforms. The issues can be resolved basing on existing software platforms, but require further research and testing.

ACKNOWLEDGEMENTS

The study was partially supported in the frames of joint grant program of the Russian Foundation for Basic Research (RFBR) and Russian Geographical Society (RGS), research project No. 17-05-41118 RGO_a.

REFERENCES

- Bonomi F., Milito R., Zhu J., Addepalli S., 2012. Fog Computing and Its Role in the Internet of Things. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13-16. doi:10.1145/2342509.2342512
- Open Geospatial Consortium, 2006. OpenGIS Web Map Server Implementation Specification. OGC 06-042, Version: 1.3.0, 85 p.
- Open Geospatial Consortium, 2007a. OpenGIS Geography Markup Language (GML) Encoding Standard. OGC 07-036, Version: 3.2.1, 437 p.
- Open Geospatial Consortium, 2007b. OpenGIS Sensor Model Language (SensorML) Implementation Specification. OGC 07-000, Version: 1.0.0, 180 p.
- Open Geospatial Consortium, 2010a. OGC WCS 2.0 Interface Standard - Core. OGC 09-110r3, Version 2.0.0, 54 p.
- Open Geospatial Consortium, 2010b. OpenGIS Web Map Tile Service Implementation Standard. OGC 07-057r7, Version: 1.0.0, 129 p.
- Open Geospatial Consortium, 2014a. OGC WaterML 2.0: Part 1- Timeseries. OGC 10-126r4, Version: 2.0.1, 151 p.
- Open Geospatial Consortium, 2014b. OpenGIS Web Feature Service 2.0 Interface Standard – With Corrigendum. OGC 09-025r2, Version: 2.0.2, 254 p.
- Open Geospatial Consortium, 2015. OGC WPS 2.0 Interface Standard. OGC Implementation Standard. OGC 14-065, Version 2.0, 133 p.
- Open Geospatial Consortium, 2016. OGC Catalogue Services 3.0 Specification - HTTP Protocol Binding. 12-176r7, Version 3.0, 130 p.
- OpenFog Consortium Architecture Working Group, 2016. OpenFog Architecture Overview. White Paper OPFWP001.0216, 35 p.
- OpenFog Consortium Architecture Working Group, 2017. OpenFog Reference Architecture for Fog Computing. OPFRA001.020817, 162 p.
- Kazakov E., Terekhov A., Kapralov E., Panidi E., 2015. WPS-based technology for client-side remote sensing data

processing. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XL-7/W3, pp. 643-649. doi:10.5194/isprsarchives-XL-7-W3-643-2015

Panidi E.A., 2016. Towards implementation of the Fog Computing concept into the geospatial data infrastructures. *Proceedings of the International conference "InterCarto/InterGIS"*, Vol. 1, pp. 34-42. (In Russian) doi:10.24057/2414-9179-2016-1-22-34-42

Panidi E.A., Efimov D.S., 2015. Using of the client-side computations when implementing Web services for spatial data processing, the case study of Java Web Start Technology. *Proceedings of the International conference "InterCarto/InterGIS"*, Vol. 1, pp. 485-491. (In Russian) doi:10.24057/2414-9179-2015-1-21-485-491

Panidi E., Kazakov E., Kapralov E., Terekhov A., 2015. Hybrid geoprocessing Web services. *SGEM2015 Conference Proceedings*, Book 2, Vol. 1, pp. 669-676. doi:10.5593/SGEM2015/B21/S8.084

Panidi E., Terekhov A., Mukhamedzyanov M., 2016. Implementation Ways for Web Coverage Service Standard. *2nd International Scientific Conference GEOBALCANICA 2016 Proceedings*, pp. 3-10. doi:10.18509/GBP.2016.01

W3C, 2017. WebRTC 1.0: Real-time Communication Between Browsers. W3C Working Draft. URL: <http://www.w3.org/TR/webrtc/>

Revised May 2017