TOWARD PEER-TO-PEER DATA INTERCHANGE IN WEB-BASED GISs

DOI: http://dx.doi.org/10.18509/GBP.2018.61
UDC: 528.8/.9:004.774.2

Evgeny Panidi
Mark Mukhamedzyanov
Evgeny Kapralov
Saint Petersburg State University, **Russia**

ABSTRACT

Development of the peer-to-peer interaction technologies is one of significant development planes of the information systems and World Wide Web. Particularly, such modern paradigms as Internet of Things and Fog Computing assume the need of direct interactions between client nodes on local or wide area networks as well as on the Web. However, peer-to-peer technologies are poorly studied and implemented in geospatial domain, despite obvious applicability in different thematic domains, for example to build monitoring grids.

The paper observes issues of peer-to-peer architecture design for Web-based GISs and geospatial Web services, and our preliminary conclusions on application of contemporary technologies as a basis for peer-to-peer interchange of geospatial data on the Web.

Keywords: Web-based GIS, Geospatial Web Service, Peer-to-peer, WebRTC, Geospatial Fog Computing

INTRODUCTION

Previously we executed a number of studies devoted to the architecture design of Web services for client-side processing of geospatial data. We developed a software prototype for online client-side geospatial data processing using software components delivered onto client computer through the Web service [9]. Some investigations we devoted also to the issue of Web publication of the geospatial data onto client hardware platforms using lightweight geospatial Web servers [10]. These studies gave possibility for approbation of distributed client-side computational and storage resources as a possible and workable part of hardware segment of the Spatial Data Infrastructures (SDIs).

Currently we are focusing on discover of architectures and technologies which can be convenient to implement peer-to-peer (client-to-client) interactions in processing chains of geospatial Web services and Web-based Geographic Information Systems (GISs) which use client-side data storage and/or processing. Such an approach is called as Fog Computing [7], [8] or Geospatial Fog Computing, and assumes that clients incorporated into system become able to store, process and transmit geospatial data directly, using their own resources.

However, implementation of the peer-to-peer interactions becomes an issue due to general differences between the server (traditional bulky server) and client (used as client only or as a lightweight server) nodes of the network. First significant difference consists in the higher applicability of high-performing server-type hardware for deployment of software Web server and specialized server-side software of geospatial Web service or Web-based GIS. Web server software is needed for initial handle of requests arrived to the data storage/processing node of Web-based GIS or to the separated geospatial Web service. Second one difference is the addressing of the bulky servers (static IP address is used usually) and clients (desktop-type or mobile-type) where dynamic addressing is used in general case, which brings difficulties in search and allocation of needed client-type node that is used as a lightweight data server for example [12].

POSSIBLE ARCHITECTURE FOR PEER-TO-PEER WEB-BASED GIS

Three main approaches can be proposed to ensure peer-to-peer interchange and distributed storage of geospatial data, which are:

- Extension of currently used Web service standards (OGC WMS [3], WFS [5], WCS [4], WPS [6], etc.)
- Torrent-like data interchange
- Blockchain-based data interchange

First approach assumes centralized architecture, as the master server is needed to store metadata records for geospatial datasets stored on distributed client nodes. Second approach makes decentralization possible, which can be achieved through the use of more than one server to store metadata records. Finally, third approach assumes fully distributed architecture, while the blockchain [1] paradigm is applied.

Extension of currently used Web service standards (first of all, OGC standards – http://www.opengeospatial.org) can be an effective way for implementation of new geospatial data distribution and interchange techniques, as it ensures inverse compatibility with currently used infrastructures [12]. As all mentioned OGC standards demand the use of getCapabilities request to initiate interaction with geospatial Web service (Fig. 1), the request can be extended with declaration of direct (from other client) data download possibility.

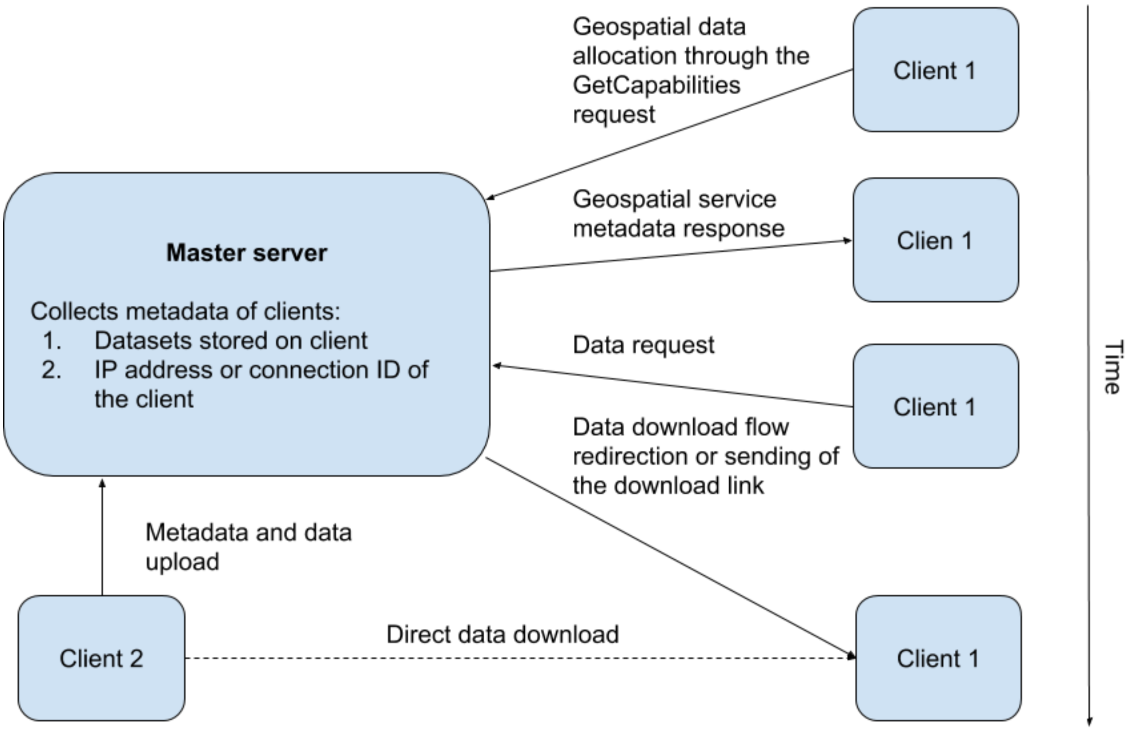
Geospatial data allocation through the GetCapabilities request
Client 1
Master server
Collects metadata of clients:
1. Datasets stored on client
2. IP address or connection ID of the client
Geospatial service metadata response
Clien 1
Data request
Client 1
Data download flow redirection or sending of the download link
Metadata and data upload
Client 2
Direct data download
Client 1
Time

Figure 1. Client-server interaction schema of geospatial Web service for the case of distributed client-side data storage

In more general case, server-side software of Web service have to be able to request needed data from one or many data-owning clients and to redirect uploaded data flow as download flow to the consuming client that requested the data. Such a technique of data flow redirection can ensure design of fully (direct and inverse) compatible geospatial Web services, able to store and process data in distributed manner. It can be time consuming however, as the data transmitted two times (from storage to the master server and from the master server to the consuming client).

To resolve the problem of software Web server deployment on the data-owning client and to make it a lightweight data server, it is needed to discover context and facilities in every case separately. Then it is possible to select from deployment of full-size Web server (e.g., Nginx - http://nginx.org), mobile application with Web server functionality (e.g., kWS – https://play.google.com/store/apps/details?id=org.xeustechnologies.android.kws ; Palapa – https://play.google.com/store/apps/details?id=com.alfanla.android.pws), or Web browser-based framework that realizes Web server functionality in the browser context (e.g., Browserver-client – https://github.com/jed/browserver-client ; Nohost – https://github.com/humphd/nohost)

BitTorrent protocol [2] assumes decentralized data storage on many clients. A metadata file is used to store description of some dataset published in peer-to-peer BitTorrent network, which is called torrent file. This file is transmitted to the consuming client at the first step and is used to establish direct (peer-to-peer) connection with other client and then to download needed data.

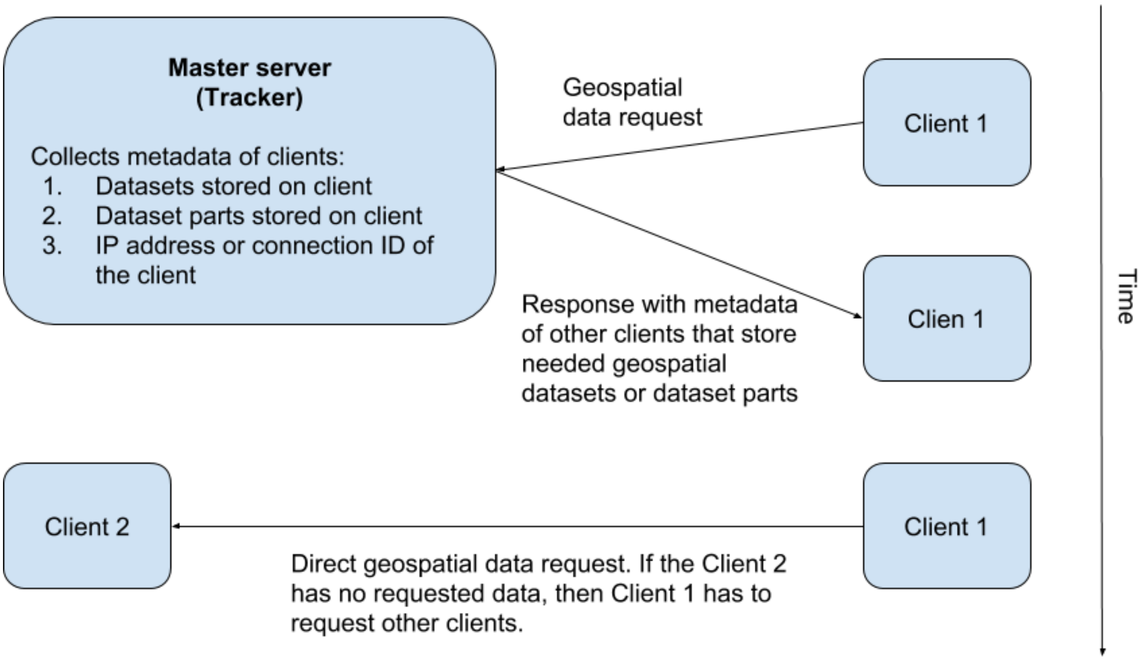
Master server
(Tracker)
Collects metadata of clients:
1. Datasets stored on client
2. Dataset parts stored on client
3. IP address or connection ID of the client
Geospatial data request
Client 1
Response with metadata of other clients that store needed geospatial datasets or dataset parts
Clien 1
Time
Client 2
Client 1
Direct geospatial data request. If the Client 2 has no requested data, then Client 1 has to request other clients.

Figure 2. Client-server interaction schema of torrent-like distributed client-side data storage

To optimize downloading of the dataset from data-owning client, the dataset is divided logically into small parts (pieces), and hash code for every part is computed and incorporated into the torrent file. Finally, the master server is called tracker server and is used for dynamically updated storage of list of the datasets stored on different clients and available for immediate download. Web link to the tracker is also incorporated into the torrent file. Thus, only the torrent file and BitTorrent-compatible downloader software are needed to be presented on consuming client to receive the dataset. A number of tracker servers can be used to find currently available data-owning clients, as well as a number of data-owning clients can be used to receive directly different parts of the dataset, while the parts a controlled and combined using hash codes available in torrent file.

In the case of blockchain-based data interchange, the blockchain itself (which is register composed of some metadata records and divided into separated parts called blocks) stored as a fully replicated distributed database. This means that full copies of the blockchain are stored on more than one of nodes of the blockchain network. Such a duplication makes it possible to ensure stability of the network and is used as one of data spoofing prevention instruments. In general case, blockchain network is not use the tracker servers, as the connection to the network is realized through the address broadcasting by every network node and receiving of addressed broadcasted by other nodes on predefined port. Any node is connected usually to the limited number of other nodes (not to all available), which broadcasted addresses it can allocate.

The blocks are separated to make the blockchain splittable and to give the possibility to store not all the blockchain but its needed part on some node. At the same time, every next-generated block stores hash code of previously generated composing consistent chain due to this. Blocks itself can be coded in different ways, for example the JSON (https://json.org) notation can be used that is native for Web. To publish some metadata record in currently constructed block, client need to push the record to the nodes it connected. Then the record is transmitted through the network between different nodes, checked and incorporated into all copies of blockchain.

There is no any currently presented and widely known torrent or blockchain technology implementation in the domain of geospatial data storage. It is obvious that both torrent file as well as blockchain can be used to store and distribute metadata of geospatial datasets, and datasets as a consequence. However, until the last time the problem was to implement client-to-client connection through the Web browser, as the user interface for Web-based GISs and geospatial Web services is implemented in most cases through the browser.

Such a technological gap was filled in 2011 with the implementation of WebRTC technology [11], which made it possible to establish real-time browser-to-browser connection and to transfer media (or data) flows from browser to browser. The technology is supported by major Web browsers currently, and any specialized browser-based Web applications (Web interfaces) can be implemented in traditional way using the JavaScript programming language and ready-to-use JavaScript program libraries.

WebRTC resolves the problem of dynamic addressing of the client devices through the incorporated mechanism of client addresses allocating and transmitting between clients. Thus, the Web RTC can be used as a platform to build upon the client-side geospatial Web services and torrent-like geospatial data interchange networks. General-purpose examples of WebRTC-based torrent-like solutions are developed and being in use already. One of such solution if the WebTorrent file exchange platform (https://github.com/webtorrent).

CONCLUSIONS AND FUTURE WORK

Contemporary general-use distributed data storage technologies and platforms are convertible to the domain-specific platforms for Web-based geospatial data publication and processing. Three abovementioned ways (standards extension, torrent-like storage and blockchain-based storage) can be selected to ensure centralized, decentralized, or fully distributed architecture of geospatial Web services and Web-based GISs. However, all three approaches can be demanded in close future for geospatial data representation on the Web. While the standards extension serves centralized of data access and allows inverse compatibility with already used software and infrastructures, torrent-like and blockchain-based geospatial data storage can reduce significantly the costs of data storing and transmitting.

Such a reduction of the costs is highly demanded in scientific research and scientific datasets production, when data publishing and access maintaining budgets are small or empty. Good example is the monitoring of environment parameters (for example meteorological and hydrological) and analysis of monitored and retrospective data. In this domain, data sources are distributed physically and logically, datasets have relatively small size in many cases, and wide range of potential data consumers are presented, which can be participants of distributed data network and can maintain data storage on this network.

Most significant direction of future work is the development of protocols for geospatial data assimilation and manipulation on distributed networks, with respect to geospatial data models. Greatest issue is to propose solution that will be easy to implement into currently used software and technological chains to reduce implementation cost and to grow wide use potential.

ACKNOWLEDGEMENTS

The study was supported jointly by Russian Foundation for Basic Research (RFBR) and Russian Geographical Society, RFBR research project No. 17-05-41118 A.

REFERENCES

[1] Dasgupta A. The Game Changer of Geospatial Systems – Blockchain. GeospatialWorld, September 22, 2017.

[2] Johnsen J.A., Karlsen L.E., Birkeland S.S. Peer-to-peer networking with BitTorrent. Department of Telematics, NTNU, 22 p., 2005.

[3] Open Geospatial Consortium. OpenGIS Web Map Server Implementation Specification. OGC 06-042, Version: 1.3.0, 85 p., 2006.

[4] Open Geospatial Consortium. OGC WCS 2.0 Interface Standard - Core. OGC 09-110r3, Version 2.0.0, 54 p., 2010.

[5] Open Geospatial Consortium. OpenGIS Web Feature Service 2.0 Interface Standard – With Corrigendum. OGC 09-025r2, Version: 2.0.2, 254 p., 2014.

[6] Open Geospatial Consortium. OGC WPS 2.0 Interface Standard. OGC Implementation Standard. OGC 14-065, Version 2.0, 133 p., 2015.

[7] OpenFog Consortium Architecture Working Group. OpenFog Architecture Overview. White Paper OPFWP001.0216, 35 p., 2016.

[8] OpenFog Consortium Architecture Working Group. OpenFog Reference Architecture for Fog Computing. OPFRA001.020817, 162 p., 2017.

[9] Kazakov E., Terekhov A., Kapralov E., Panidi E. 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, 2015. doi:10.5194/isprsarchives-XL-7-W3-643-2015

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

[11] W3C. WebRTC 1.0: Real-time Communication Between Browsers. W3C Working Draft, 2017.

[12] Panidi E. Fog Computing Perspectives in Connection with the Current Geospatial Standards. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLII-3/W2, pp. 171-174, 2017. doi:10.5194/isprs-archives-XLII-3-W2-171-2017