UDC 004.42

OPENSTREETMAP DATA RENDERING

ANTON SHUBKO, RYKHARD BOHUSH
Polotsk State University, Belarus

A map as a symbolic depiction emphasizing relationships between elements of some space was invented many centuries ago. And it is not surprising that online maps have become a common and popular thing in our age of high technologies. In the modern world people extensively use online maps for different purposes such as the search of a route for a walk, search of the necessary address or assistance during an excursion in an unfamiliar city.

There is a huge number of existing online map products, free or paid, with different accuracy and detailing levels, however I would like to focus on the particular one, the OpenStreetMap. OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world. Rather than the map itself, the data generated by the project is considered its primary output. The creation and growth of OSM has been motivated by restrictions on use or availability of map information across most of the world, and the advent of inexpensive portable satellite navigation devices.

Conceptual difference of the OpenStreetMap is in the fact that OSM is not just a map, but a set of geographical information. Together with the fact that the project is not commercial, this opens broad horizons for further development based on this platform.

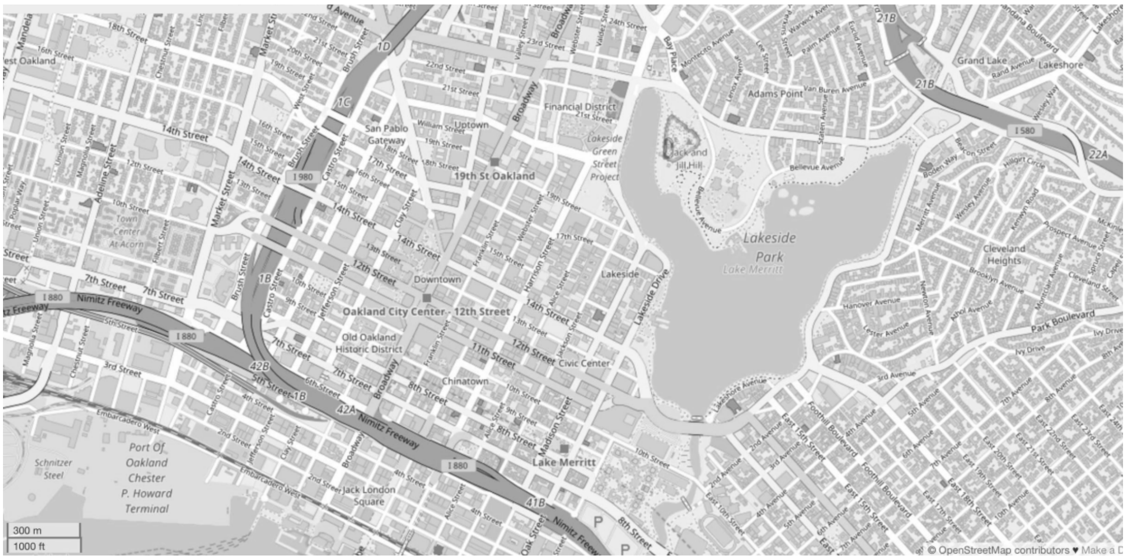

Figure 1. – Example of OSM map

The main objective of the paper is to define the process of development of a product based on the OSM for rendering geographical maps.

Analysis of the subject domain made it possible to delineate the following technical requirements for a typical product based on the OpenStreetMap data:

– Transformation of XML data to object-oriented data and their rendering as cartographic images;

– Development of a user-friendly web-resources for displaying the maps (using such features as cached data for fast access, ability to re-render the new data on-the-fly).

– Implementation of ability to customize appearance of the map (e.g. to change designations, the displayed objects layers, etc.).

These requirements would let the product if not beat then at least correspond to functionality of the product competitors which are available in the market. However, even the mentioned range of requirements lead to a number of problems that need to be resolved during the development process.

Firstly, work with large volumes of data is the real issue for such software products. At the moment, size of the archived OSM of data is nearly 80 GB, and the unpacked data size is more than 200GB. Also it is important to point out that the data constantly and dynamically change (about 3GB of the archived data per week).

The problem defined above can be solved using several specific technologies and frameworks. For example, usage of the Java-based StAX API allows to avoid issues with memory overflow while parsing of XML data. Furthermore, for the reduction of memory consumption in such a case it is possible to apply such well-known software development pattern as Flyweight. Despite the fact that the Java programming language has been chosen for the development, and the existence of a quite popular opinion that JVM-based languages are oriented to high memory consumption in runtime, there is a great variety of options for its optimization on different levels, from the highly specialized libraries for implementation of a particular part of logic, to alternative implementations of the Java Virtual Machine itself that would provide developers with precise control of memory usage and a set of options of how it could be reduced.

One more problem that might be come across during the implementation of such product is high server load while rendering images and also load on a communication channel during transmission of images to a user.

As options for solution of the problem, popular concepts of microservice and serverless architecture can be considered. In general, the described architecture approaches could allow to reduce time expenses related to rendering, because they imply easy ways to organize horizontal scaling of a system, multithreading design, and help to organize stable load balancing to support a great number of the web-resource users. Currently a lot of successful commercial products tend to be created based on the microservice architecture, because it is flexible, easily modified and easily expandable. Moreover, splitting responsibilities for particular parts of logic between different services allows to increase resilience of the product, because particular small web-services can be easily restarted/replaced, and there are a lot of existing frameworks and products that allow to organize several hosts with the same application as a single cluster under the reverse proxy and manage their availability dynamically to correspond to current client needs and load.

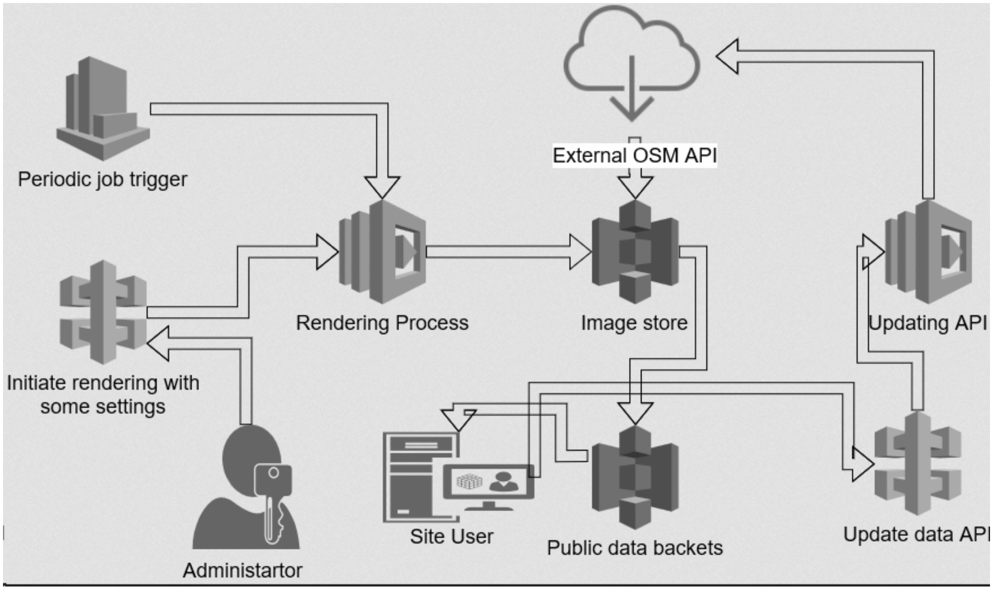

Figure 2. – A chart of high-level product architecture

With respect to scaling opportunities of microservices and economy of resources (i.e. an ability to use additional capacities only when they are really necessary, like peaks with high load of users or while rendering process is started), it is possible to consider cloud solutions such as:

- Microsoft Azure;
- Google Cloud Platform;
- Amazon Web Services.

All the enumerated options provide solutions from-the-scratch which allows to avoid expenses for labor-consuming hardware maintenance and to save a lot of money and resources during idle times. Besides the simple provisioning of computing capacities or simply remote servers for web-service deployment, now they allow to organize stable, highly scalable web services, either serverless or server-based, with databases and message queuing services deployed in the same cloud, maintaining guaranteed resilience and advanced options for cost management.

Finally, it is worth mentioning that there is a need to constantly update image database. Such dynamic data as maps should be easily updated in real time with the minimum or even without downtime to provide users with the correct and topical displayed information in time. Users of today's web expect that the pages they visit will be interactive and smooth and that is where the developer needs to increasingly focus the time and effort. Pages should not only load quickly, but also run well; scrolling should be stick-to-finger fast, and animations and interactions should be silky smooth.

This problem can be solved by applying several advanced mechanisms for data processing. More precisely, it means that, for example, there would be a global task to redraw the whole map, and a set of small background subtasks, for updating only a certain zone of the map. In combination with the versioning of displayed images, it becomes possible to update the map (both entirely and making small amendments) in a comfortable way for the users who have already started to use the map.

The developed product could have a great range of application areas, such as tourism and recreation, scientific researches, etc.