Distributed generation of image pyramids for web map services

Daniel Eggert
GFZ German Research Centre for Geosciences
Telegrafenberg
Potsdam, Germany
daniel.eggert@gfz-potsdam.de

Mike Sips
GFZ German Research Centre for Geosciences
Telegrafenberg
Potsdam, Germany
mike.sips@gfz-potsdam.de

Doris Dransch
GFZ German Research Centre for Geosciences
Telegrafenberg
Potsdam, Germany
doris.dransch@gfz-potsdam.de

Abstract

Web map services (WMS) are a common technique to publish high-resolution satellite images, allowing fast pan and zoom operations on the client-side. To reduce the computational costs on the server-side, WMS services usually process image data into a tiled image pyramid in advance. The generation of such an image pyramid from a big high resolution source image, usually distributed over multiple storage and processing nodes in a cloud environment, causes a high network traffic when executed by a single node. We present a new approach that distributes the generation of the image pyramid across all nodes, reducing the induced network traffic by exploiting data locality.

Keywords: web map services, wms, wmts, image pyramid, distributed parallel processing

1 Web Map Services

A web map service (WMS), e.g. provided by the GeoServer software (Open Source Geospatial Foundation, 2014), is a widely utilized technique to publish satellite images. The basic idea is to extract specific regions of an image at a particular zoom level and transfer this region to the requesting client. WMS provides two extraction strategies. First, the WMS extracts a particular region from an image on demand. The challenge with this strategy is the high computational cost to extract arbitrary regions from images. In contrast to the first approach, WMS might publish image data that is split up into multiple image tiles. Here, the service is called a web map tile service (WMTS). These image tiles are generated in advance for all zoom-levels and stored in a hierarchical manner called an image pyramid (Adelson et. al, 1984). Various tools and frameworks like gdal (GDAL/OGR contributors, 2018), GeoTools (Open Source Geospatial Foundation, 2017) provide implementations of the image pyramid algorithm, and this algorithm is often also built-in into the WMS software. Publishing the image data through a preprocessed image pyramid utilizes the available disk space to reduce the computational costs and therefore enables fast panning and zooming for the requesting client. Many services like Google Maps and OpenstreetMap employ this approach.

2 Big Data Challenge

Recent technological advances in parallel analysis engines such as Hadoop, Apache Flink, and Spark allow users to process and to analyse satellite images at high spatial resolutions; e.g., 10 meters. To process high-resolution satellite images, parallel analysis engines divide satellite images into multiple chunks and distribute these chunks to the available computing nodes of a cloud environment. The division and distribution of chunks constitute a challenge for the current approach to compute the image pyramid. To compute the image pyramid for a particular image, current tools execute the following steps: a) merge all distributed chunks into a single high-resolution image; b) subsample this image according to the current zoom-level and c) divide the subsampled image into tiles of the image pyramid. Those steps are repeated for each zoom-level. This approach generates a high network traffic within the cloud environment, when executed by a single processing node, making it impractical for cloud environments with distributed data storage.

3 Distributed Generation Approach

To address this challenge, we developed an approach to generate image pyramids in a cloud environment. The basic idea is to omit the need for a single high-resolution image, which is the main source for the emerging network traffic. Hence, we subsample and divide the individual chunks independently on each computing node into tiles of the image pyramid. In contrast to the original algorithm, we do not need to generate the single high-resolution image on one node to compute the tiles of the image pyramid. The challenge with our approach is that the chunks generated by parallel analysis engines do not necessarily align with the tiles of the image pyramid; e.g., a tile might cover multiple chunks. Hence, a tile is generated multiple times on different nodes. To address this challenge, we collect all instances of the same tile and combine them into a single tile in an additional reduce step. We implemented and tested the described technique with the Flink stream processing framework (Apache Software Foundation, 2017).

4 Conclusion

The presented approach enables us to distribute the generation of the image pyramid over all cluster nodes, and to reduce the necessary network traffic to a minimum.

This enables us to integrate the generation of the image pyramids as processing step into cloud analysis workflows. We can publish the resulting image pyramid via WMTS to provide users access to analysis results, e.g., to explore the pixel-wise classification of land use.

References

Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J. and Ogden, J. M. (1984) *Pyramid methods in image processing*, RCA engineer, 29(6), 33-41.

Apache Software Foundation (2017) *Apache Flink: Scalable Stream and Batch Data Processing.* [Online] Available from: https://flink.apache.org [Accessed 15th February 2018].

GDAL/OGR contributors (2018). *The GDAL/OGR Geospatial Data Abstraction software Library*. The Open Source Geospatial Foundation. [Online] URL http://gdal.org [Accessed 10th April 2018]

Open Source Geospatial Foundation (2017) *GeoTools - an open source library providing standards compliant methods for the manipulation of geospatial data*. [Online] Available from: http://geotools.org [Accessed 10th April 2018].

Open Source Geospatial Foundation (2014) *GeoServer - an open source server for sharing geospatial data*. [Online] Available from: http://geoserver.org [Accessed 15th February 2018].