Scalability of OD-data visualizations

Ilya Boyandin

Teralytics

Keywords: geographic data visualization, movement of people, mobility, transportation, flow maps, scalability

1. Introduction

Understanding mobility patterns is important for fields like migration studies, urban and transportation planning, epidemiology, ecology, and disaster response. Origin-destination data (OD-data) are often used in this context for analysing numbers of movements between geographic locations. We believe that many OD-datasets remain under-analysed. The reason for that is that not many analysis tools are available today that are designed specifically for this kind of data and that are, at the same time, easy to use.

To improve this situation we published and open-sourced Flowmap.blue[1], an online tool which makes it very easy to create interactive geographic flow maps from datasets uploaded to Google spreadsheets. Since it was published few months ago, hundreds of datasets from all around the world have been visualized with it.

This simple tool, however, only works well for relatively small datasets (tens of thousands rows). The size of OD-data depends quadratically on the number of locations involved. It can quickly grow to hundreds of millions of rows when flow attributes like time, mode of transport, duration are added to the dataset. Such large datasets cannot be entirely visualized in one image. Their analysis requires the use of summarization and interactive exploration techniques. Moreover, ensuring smooth interactivity and short query response times necessary for such interactive analysis requires using an efficient database for executing the queries.

At the company Teralytics we are building exploratory tools for the analysis of aggregated data on movement of people in cities and countries with the purpose of improving transportation and mobility services. Scalability to the growing data sizes is an important challenge we are facing. In this article we discuss some of the technological solutions we have been working on to address this challenge and the tools we have published and open-sourced to make these solutions available to the broad public.

2. Scalability of flow maps

In our tools we use flow maps as the main geographic representation of OD-data. They represent numbers of movements between locations as lines of varying thickness drawn on a geographic map. Flow map is the most straightforward and often used representation of OD-data, but it has limitations.

One important problem with flow maps is that, depending on the nature of the dataset and the number of lines drawn, there can be a significant overlap caused by the line crossings (Figure 1a). This problem is sometimes addressed by applying edge bundling techniques (Lhuillier et al, 2017, Graser et al, 2017).

Another problem is that short flows (which are often also the largest ones, because close-by regions are often more connected than those which are far apart) can be difficult to see (Figure 1c). Both these problems are more likely to arise, the more locations and flows are in the dataset.

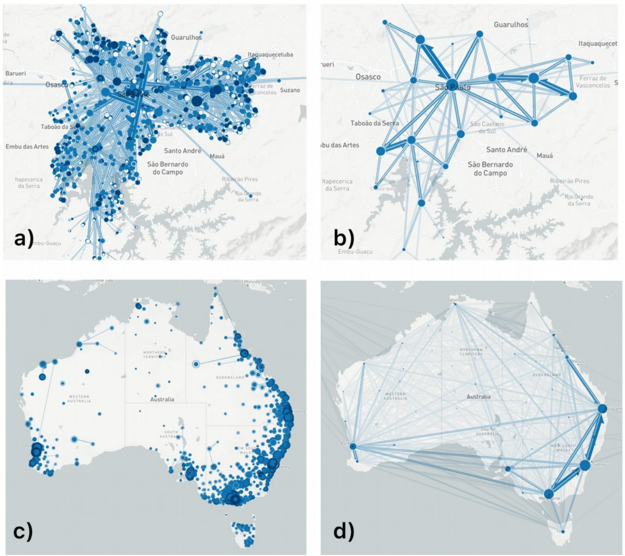


Figure 1. Bus rides in São Paulo. a) The original "messy" version with too many overlapping locations and flows. b) A clustered version of the same dataset showing high-level patterns.
Relocations in Australia c) The original version with the most important flows being too short to be visible. d) Clustered version making high-level patterns visible. The circle sizes show the locations' in-/out totals and include internal flows.

To address these two problems we implemented an adaptive clustering approach. Locations within a certain distance from each other (the distance depends on the current map zoom level) are clustered together (Figure 1b, d). The clusters are positioned in the centers of masses of the locations constituting them (the locations are weighted by their total in-/out flows). After the location clusters are formed, flows are aggregated by summing up the magnitudes of the flows connecting the constituents of the clusters.

We are using a simple and very efficient density-based clustering algorithm implemented in the Supercluster[2] library. Instead of the automated clustering approach, taking an existing administrative area hierarchy may result in more meaningful and familiar clusters. However, with the former it is possible to produce a separate clustering for each map zoom level providing for a smoother user experience. In any case, the flow aggregation step doesn't depend on any particular algorithm and can be applied to any clustering.

The clustering level adapts to the map zoom making sure that not too many flows need to be drawn and that all the drawn flows are not too short. Flow lines which are too short are summarized as cluster-internal flows and are represented as part of the location totals by the circles of varying sizes. When zooming in, the clusters will gradually expand, so the level of summarization will automatically adapt to the map viewport.

[1] https://flowmap.blue

[2] https://github.com/mapbox/supercluster

This approach makes it possible to visualize and explore very large OD-datasets providing a useful summary at first and allowing the users to zoom in to see detailed flows within specific regions of interest. For an efficient implementation the approach can be combined with map tiling[3], so that only the data for the visible map tiles of the current summarization level needs to be fetched.

3. Scalability of the data backend

Often flows in OD-datasets have additional attributes, e.g. time or mode of transport. This is useful for exploring differences between various types of flows or for comparing the movement patterns emerging at different times. Hence, an OD-dataset is a list of tuples *(origin, destination, magnitude, ...attributes)*. To support the exploration of such data we developed a system allowing the users to cross-filter flow data by the attributes or by selecting locations and then to visualize the resulting flows as a flow map.

In our first implementation we loaded the entire dataset in browser and did the cross-filtering there. However, we quickly realized that it was only fast enough for small datasets. We looked for a database solution to support large OD-datasets (~1 billion of rows) such that it would also be possible to scale it horizontally if a dataset was growing. Initially we used Postgres, but it didn't perform well enough to support interactive analysis.

We evaluated several scalable database solutions and found that Google BigQuery and ClickHouse fulfilled our response time requirements (queries shouldn't take more than a few seconds). Both support SQL and are designed with scalability in mind. They are also both column-oriented. This means that the actual data is stored on disk column-by-column, not row-by-row like in traditional relational databases. Hence, only the data for the columns referred to in the queries need to be read from disk, not all of the columns. In our queries we either refer to two *(attribute, magnitude)* or three columns *(origin, destination, magnitude)*, whereas the total number of columns can be much larger. Reading data from disk is often the most time consuming step of the query execution, so reducing it to the bare minimum has a significant positive effect on the performance.

Both BigQuery and ClickHouse scale horizontally, so dealing with larger datasets is a matter of adding more machines for query execution. However, BigQuery only offers a managed solution hosted in the cloud by Google. ClickHouse is an open-source database which we can host in our own data centers. Some of the data we work with at Teralytics is sensitive and cannot be uploaded to the cloud. For this reason, we decided to go with ClickHouse, and it has worked out very well for us.

The system architecture of our OD-data exploration tool backed by ClickHouse is shown in Figure 2. First, the API queries from the application front-end running in the browser arrive to the application back-end. There, SQL queries are formed and are sent to ClickHouse.

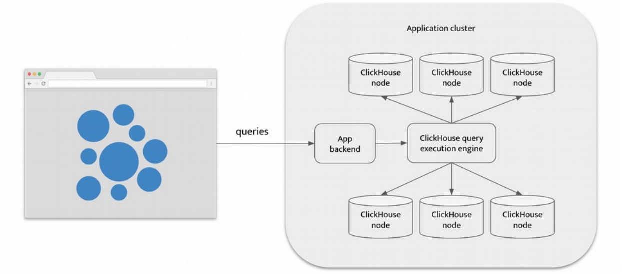
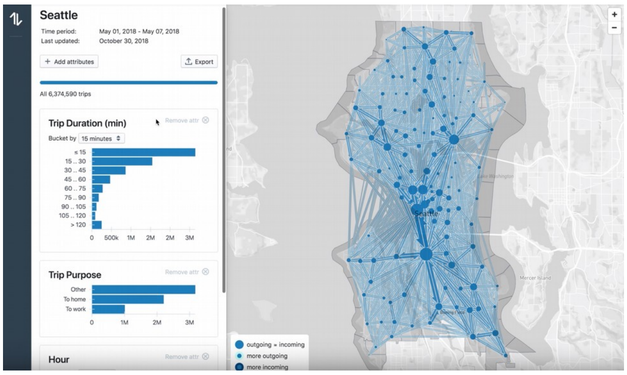


Figure 2. The system architecture of our scalable OD-data exploration application.

The ClickHouse query execution engine splits the work into multiple jobs and executes them in parallel on all the available machines so that the query results can be delivered as fast as possible with the available resources. The query performance for our OD-datasets has been very pleasing with the described set up.



Seattle

Time period: May 01, 2018 - May 07, 2018
Last updated: October 30, 2018

All 6,374,590 trips

Trip Duration (min)

Bucket by: 15 minutes

Trip Purpose

Hour

Figure 3. Flowmap.query, an exploratory visualization tool for the analysis of OD-data with attributes backed by an efficient database.

We are open-sourcing a demonstration version of this solution as Flowmap.query[4] (Figure 3). Currently, it only supports ClickHouse as its backend, but we plan to add BigQuery support soon.

4. Conclusion

In this paper we discussed two particular solutions for enabling scalability in OD-data exploration tools. One is the adaptive zoom-dependent location clustering for flow maps. Another is employing a scalable database backend for interactive querying. When used in combination, these two techniques make it possible to explore and analyse very large OD-datasets. We are open-sourcing parts of these technological solutions. Our goal is to make it easier for the large public to produce flow maps from OD-datasets of any size and to interactively explore them.

5. References

Lhuillier, Hurter, C., Telea, A. (2017) State of the art in edge and trail bundling techniques, Computer Graphics Forum

Graser, A., Schmidt, J., Roth, F., & Brändle, N. (2017) Untangling Origin-Destination Flows in Geographic Information Systems. Information Visualization 1-20

[3] https://en.wikipedia.org/wiki/Tiled_web_map

[4] https://github.com/teralytics/flowmap.query