PingER data warehousing report for 2016

PingER Data Warehousing

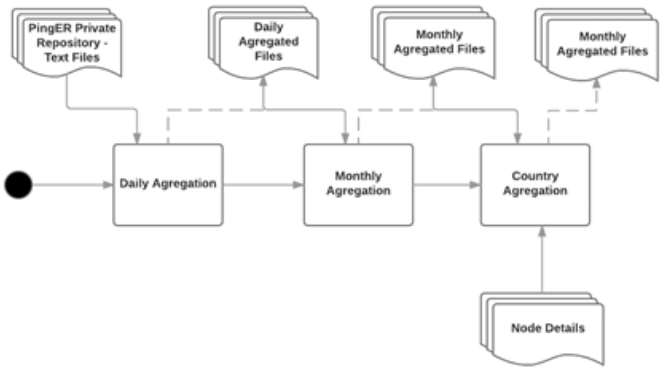
During the summer of 2016 was developed an application to keep a PingER data warehouse, a system where all the PingER data would be stored in a efficient way. This task was divided in three main phases: the creation of a script that automates the update of the PingER public FTP, the creation of a four node cluster where the Hadoop file system (HDFS) is running and the creation of an ETL (Extraction, Transformation, Loading) process to populate the data warehouse everyday. This process allows the users to perform queries involving millions of rows in a much faster way.

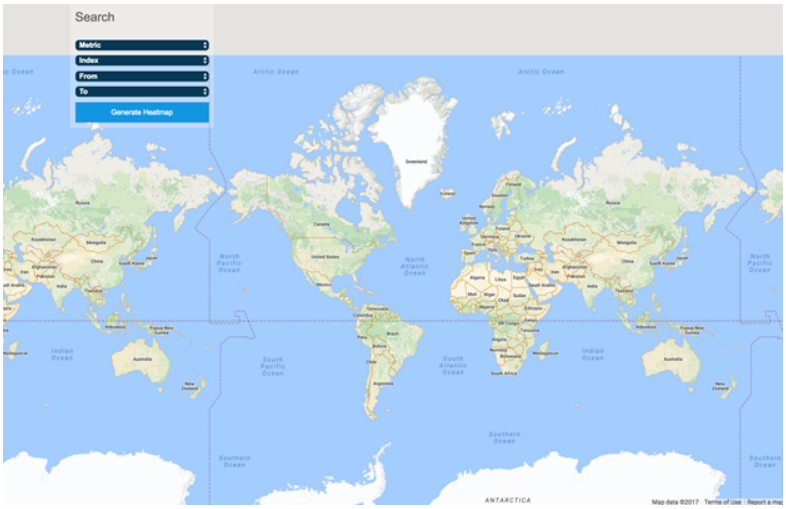
PingER Visualization

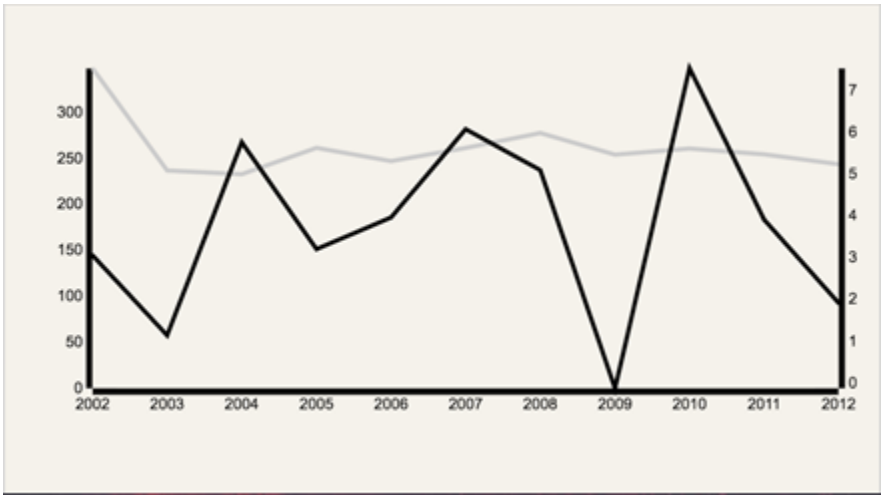
Since september of 2016 the project PingERVis is under development. This project aims in develop an interactive visualization system to PingER Data where the user can select some attributes, generates different kinds of visualization and relate PingER data with a bunch of other metrics (economic and social index, for example). This project is focused on reduce the query response time, giving the user a very efficient system to explore PingER data and provide different approaches to data exploration. This project was divided in two main phases: data normalization and data visualization.

The data normalization pipeline was developed using Python programming language and consists in three main phases: daily aggregation, monthly aggregation and country aggregation. The goal of this pipeline is keep a smaller, but representative, set of data which any kind of relational database can consume and perform queries in a low response time. This pipeline also includes a algorithm to remove outliers from the PingER dataset. This pipeline can be configured to run everyday using SciCumulus (tool to manage scientific workflows).

The data visualization phase allows the user to perform spatial data exploration and timely data exploration. The spatial data exploration consists in let the user choose some set of countries and then visualize the data only for the selected countries. The timely data exploration allows the user to select a time range to visualize the data. Besides the capabilities mentioned above, the system can also consume any set of data, following a specific standard, to be related with PingER data (economic and social index, for example). During the development, the main tool used was Flask Framework to generate the web application and Javascript libraries like D3.js and Jquery.







Performance

All the queries are taking under than 30 milliseconds.

I'm trying now to automate the data aggregation, so I can run it everyday using crontab and populate my relational database.

I'm looking for some student good in front-end design to help me with the page layout.