DATA VISUALIZATION

ON UNSTRUCTURED DISTRIBUTED DATA

Omar Rostom

Yehia Arafa

Nada Essam

Esraa Hefny

Ali Rashad

Bassant Ibrahim

• Introduction and Project outline

Data visualization is about conveying data as efficiently as possible, data scientists and analysts always use visualization techniques to explore patterns in their data and show their result effectively.

And so choosing an appropriate type of visual that correctly describes the data is very important to understand the information in these data in an easy and straightforward fashion. However, the hard part is identifying visualizations that show relevant in the desired dataset.

We introduce a system that recommends interesting visualizations which helps data scientists and analysts connecting things in interesting ways and look at data from different angles.

Our system operates on a large scale unstructured data distributed over a cluster using impala in which storage devices are not all attached to a common processor and doesn't reside in a traditional row-column database, it may include word-processor documents, the body of emails and many other kinds of business documents, Our system then takes these data and converts it into structured data as those in any DBMS which resides in a fixed field to be simple and easy to use, later we run our many operations on top of this dataset.

We tackle the problem of identifying and showing the appropriate type of visual according to some user's input so we help them understand what they want from the data easily.

Then our system works as recommendation engine which intelligently recommends visualizations which are most interesting and useful.

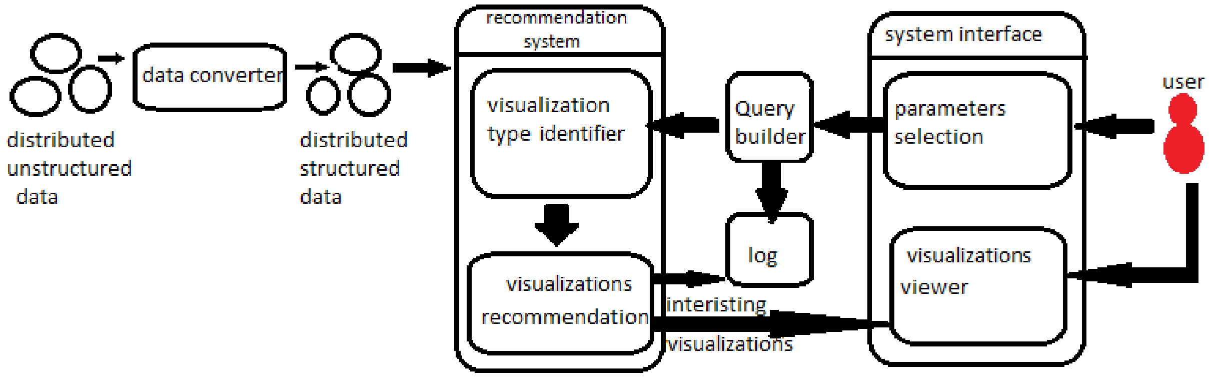
So we can divide our problem here into two minor problems:

1. Converting large scale unstructured data to a structured one.
2. Showing appropriate type of visual.

And a major problem:

3. Recommending interesting and useful visualizations.

• Architecture


distributed unstructured data
data converter
distributed structured data
recommendation system
visualization type identifier
visualizations recommendation
Query builder
log
interisting visualizations
system interface
parameters selection
visualizations viewer
user

- Data converter: To convert unstructured data into structured we need to construct a parser for the specific inputs, and use that parser to generate structure from what it discovers about the input.

- Query builder: it builds a query from some parameter specified by the user, which is then used by the recommendation system.

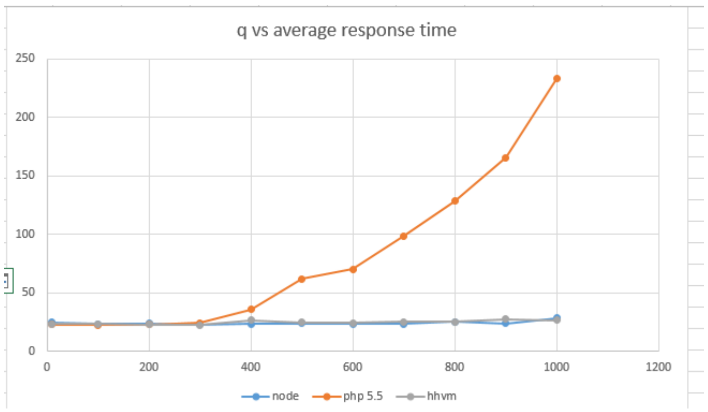
-Log: To speed the process of visualization we need the system to be smart and improves its performance by considering the previous work it did so that it can spend less time in processing. We use the log for this purpose. It saves some information about the queries and the visualizations the users found interesting to send it later to the recommendations system when operating on similar data with similar query for more fast and efficient results.

- Recommendations system: It is divided into two parts: The first part is the **visualization type identifier** which identifies the best visualization type (e.g. bar chart, scatter plot or histogram) depend on the data it operates on and make the user choose which type he want. The second part is the **visualization recommender** itself which executes the query on the data and then recommend the top-k visualizations using some techniques to determine the most interesting visualizations for the user. This system uses some optimization techniques to return results in interactive time. Finally, the system will send some information about the query and the visualizations the user found interesting to the **log** to use it later when operating on similar data set with similar query.

• Display Visualized Data

Displaying visualized data is our main aim in our tool. Our main aim in our tool is to provide automated visualizations that will help our users and cut down time used to choose and select proper visualizations. Many types of visualizations will be available for selection, bar chart, histograms, scatter diagrams … etc, each of which will be displayed upon the best for the user through his previous queries and his past dataset history. Machine learning will be play an important role, as the log used for each profile will maintain a steady record for each user to simplify his future queries and optimize his current experience. We will use a very powerful tool D3.js built with the powers of javascript to enhance our visualization algorithms, and display it on to the user ui, which he will have more options to change the visualization shape, and change the aggregate function used and/or where each value goes to which axis.

• Which technologies used ?



We will use javascript as our main language for both backend and frontend. Javascript has always proved that it is the most powerful and fastest language existing right now, offering a wide range of frameworks that work flawlessly on both sides backend and frontend.

The api will be written on the basis of node.js which proved it is one of the best in the current time in terms of time and performance *figure provided shows javascript vs php vs ruby* the thing that made our choice clearly understood.

In terms of delivering the interface, we decided upon using one of the new comers, a very powerful library vue.js way more faster than angular2 and angular and much lighter, having a popular reputition and documentation we decided to go with it in terms of front end.

One of our tool's most concerning issue, is security related to each own profile, and how data is secured from outer attacks, and even from other users on the same domain, we will use secure tokens for each profile, using one of the best available right now, JWT (json web token) written for most existing programming language, and having a wonderful reputitoin in terms of security, we decided to go with it.