

Google Style Rank Based Searching for Investment Funds with Thousands of Data Attributes

Mr. Fakruddin Mohammed
fam281@g.harvard.edu

Motivation

Searching (text based) for information has become essential part in the era of digital world. We all know how the search should work but the reality is implementing a good search engine is extremely challenging task. Therefore, in this short tutorial I wanted to explore the search technologies.

Problem Statement

Nowadays every business have a Digital Front End providing informational services to the end users. When the user is looking for information, the ability to give the most relevant information is key ingredient to the success of the business who purely rely on digital data services. There are number of ways one can add text based search features to the application such as Relational Database queries or No-SQL based data queries or Remotely hosted CGI scripts. But the issues with these approaches is that it doesn't use contextual information from what the user is asking for, it doesn't give the results sorted by relevance and the free text based search implementation is extremely difficult task, often simplified or compromised to make implementation easier therefore, it doesn't retrieve the results accurately what the user is actually looking for. Therefore, in this short article the study explores the alternate ways to implement the search engine features using the investment funds data (mutual funds, exchange traded funds and hedge funds data). The reason for choosing the financial instruments data is due to the fact that a given investment product can be described in using thousands of attributes, therefore, it challenges the implementation. The data is web scraped from the publicly available websites and where data is not available, initialised with random data and normalised to the OpenFunds standards.

Technology

The following software stack is used to build the device.

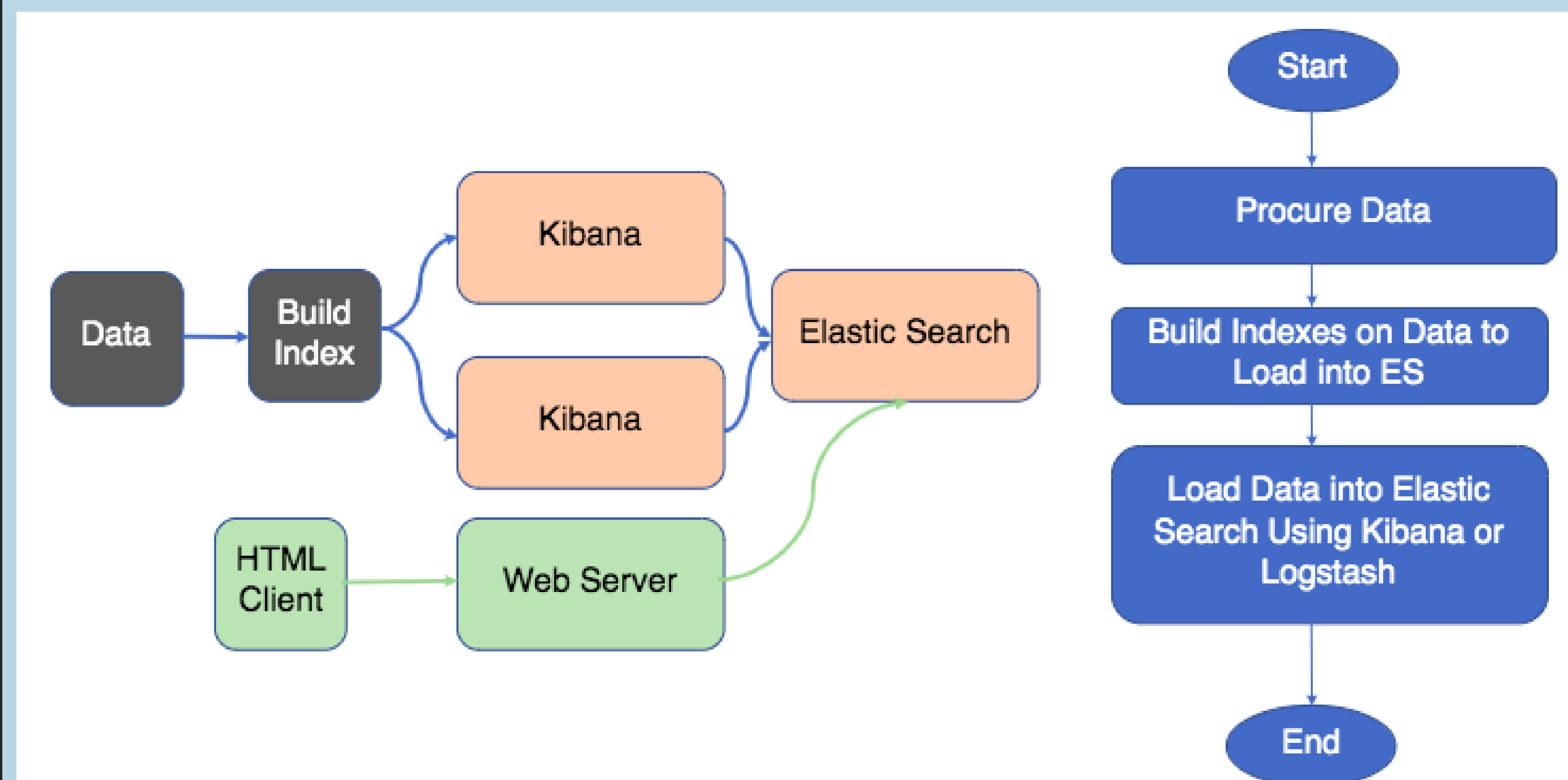
- ELK Stack (ElasticSearch, Logstash Kibana)
- HTML, CSS
- Java Script
- Python
- Apache
- Excel/VBA

Hardware

The experiment is conducted on the following hardware platform.

- MacOSX Laptop (4 CPU, 8GB RAM, 256 SSD Hard disk)
- Python Simple HTTP Server

Architecture

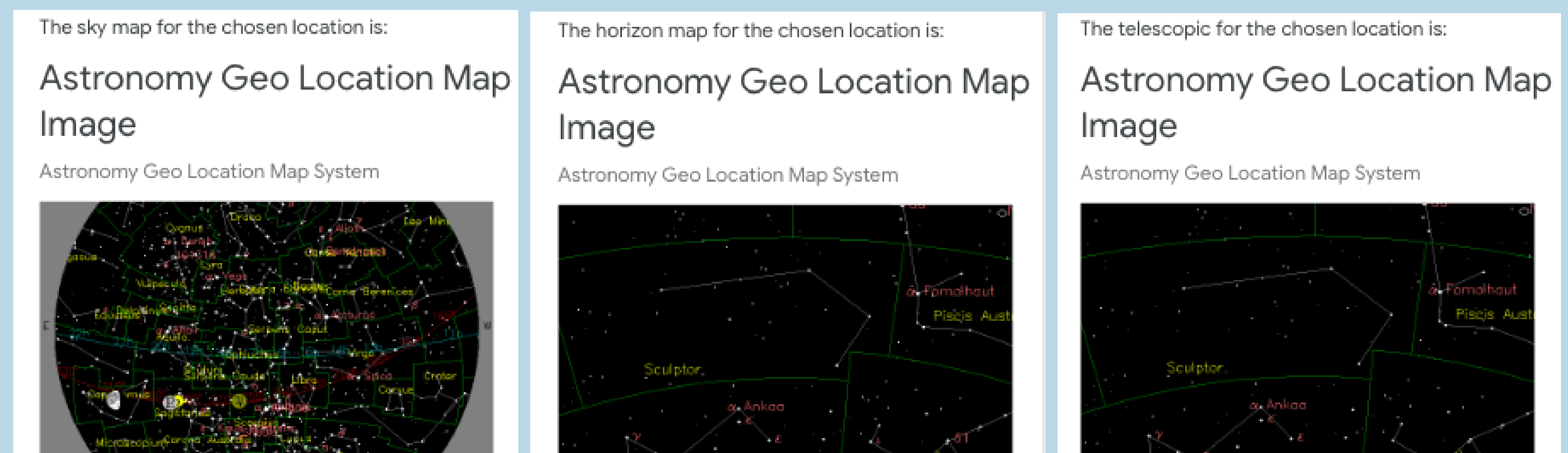


Data Flow, Current Status and Next Steps

End to End Data Flow The user can have human like conversation with the device by speaking into Google Assistant enabled handheld devices. The user has choice to ask the device any of the following three:

- Issue voice command to drive the model car
- Ask for details of 500+ clusters, constellation and stars
- Ask for sky or horizon or telescopic night sky maps

A function written in python language and deployed on Google Cloud Platform (GCP) is listening to all input requests from Dialogflow (NLP API platform), performs the web scraping [1][2] and responds back to the user with the requested details.



Current Status Using the Software (AI-NLP, Google Cloud Platform) and hardware (2 wheel driven model car equipped with GPS module) a low cost one-stop platform is successfully built where the user can ask for moving telescope platform to retrieve the details of celestial and localised maps.

Next Steps Add the voice to the search engine where the human voice is translated into text and return the search results to the user.

References

[1] Stuart Lowe's Website: <https://www.lifewire.com/searching-your-site-3466200/>, this website gives brief background on website search engine features (2017).