**Fast process and analyze big data using cloud based services**

*Siyu Zang s3534987*

*Zhaowei Wang s3443552*

**Contribution**

Siyu Zang: 50%

Zhaowei Wang: 50%

**Links**

AWS EC2 DNS: ec2-54-206-88-178.ap-southeast-2.compute.amazonaws.com

**Summary**

The first idea of this project is to process big data very fast with very limited hardware resources and AWS cloud services, that is to say, by using 1G memory and 1GHz single core CPU, 10 million orders can be processed less than 1 second!

The second idea of the idea is to analyze those data and generate graphical results which brings more convenience for the stakeholders.

**Introduction**

**What are the motivations behind our idea?**

The number of web and mobile applications are booming these days, however many of those applications do not have a powerful built in data analyzer during the development, and most existing third party data analyzers are quite expensive to use. So here comes this project: A not only low cost, but also powerful cloud based data analyzer.

**What it does?**

Process and analyze data rapidly with no extra cost.

**Why it is required?**

To help the web/mobile applications owners have better understanding of what happened behind the data so that the services they provided can be improved. In addition, plain data files for example CSV and JSON are not easy to read, while graphical drawing are more user friendly.

**How it can be used as real life application?**

Since this is a cloud based solution, it’s easy to deploy to any web application.

**The positive and advantages of this application?**

Low cost, fast, user friendly, stable.

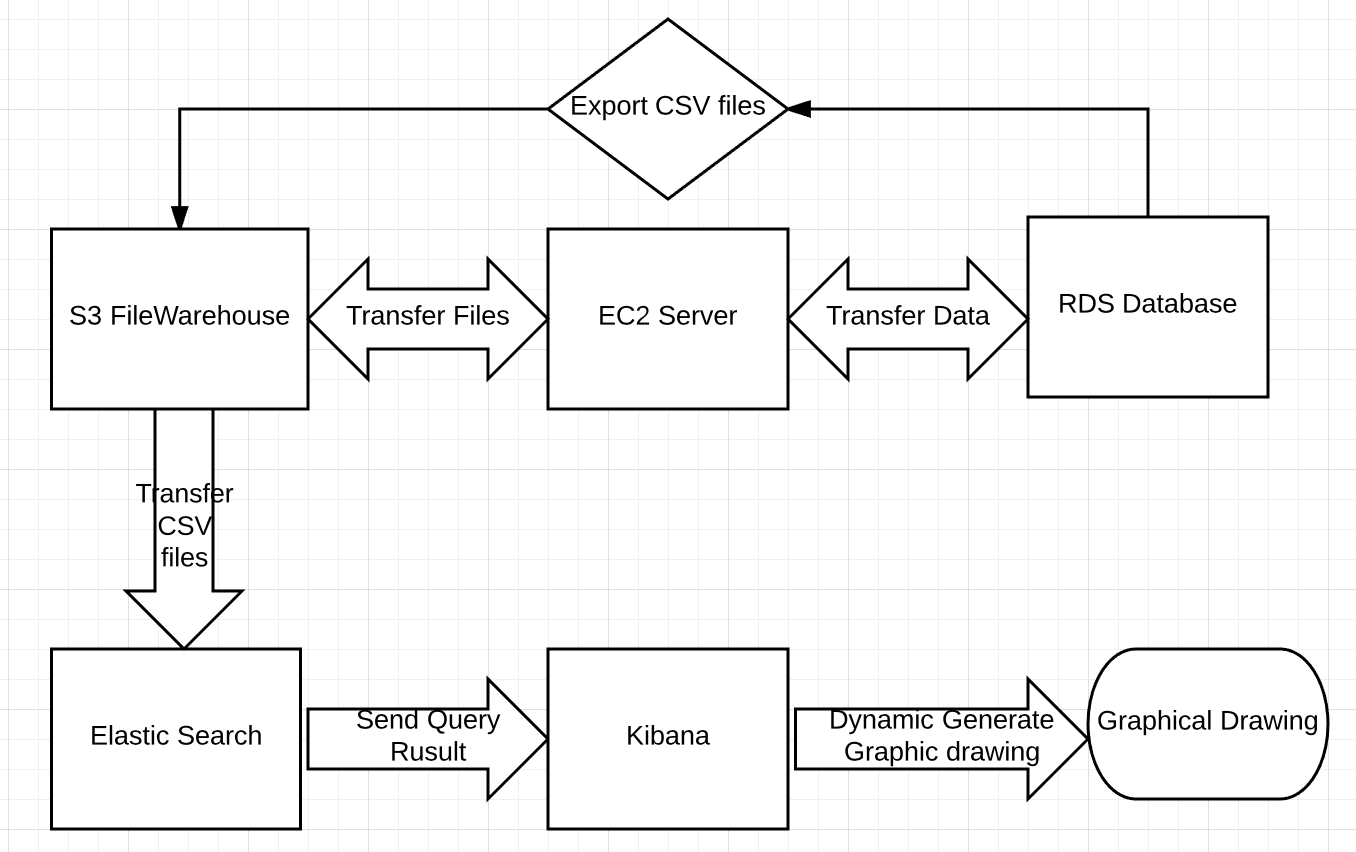
**Related work**

Google Big Query, Amazon Redshift, Hadoop MapReduce all provides big data process and analyze functionalities.

**Why not Hadoop**: First of all, the sample web application was using MySQL database, not MongoDB. Which means the data is quite structured, so the SQL liked query can run fast. The MapReduce’s advantage are not that obvious. Because the MapReduce is ideally for the un-structured data featured from different API, for example the Tweets from twitter and Facebook’s post. One slave node did not run fast; it took around 20 second (20 time slower than current solution) to process the same data. It could be faster by using multiple nodes, however it increasing the cost.

**Why not redshift** (same as big query in google): Not open sourced, which means not free to use. Charge $0.7 per hour. And do not support RESTFUL API

**Software Architecture**



**Implementation**

A web or mobile application need to be host remotely with at least one database to provide source data. In this project, a AWS EC2 instance was lunched as the cloud server and a AWS RDS database was built to store all data.

Then a AWS S3 file warehouse need to be establish so that it can:

1. Store all files relevant to the web application. E.g. images, audios.
2. Import CSV files from database and then automatically transfer the data to Amazon Elastic Search.

In addition, Elastic search should be configured to retrieve data from S3 and then automatically generate JSON key value pairs and index all the possibilities when first running. Once this procedure finished, a Kinesis firehouse client which installed on ec2 server will remove the CSV files on S3 immediately without any human operations.

Kibana then will receive the results from Elastic search through RESTFUL API and graphic drawings will be generate automatically.

**User Manual**

1. Connect the website’s database using DBMS for example PHPMyAdmin
2. Export the database data to CSV files directly to S3 through API
3. Use S3 send files to Elastic Search through API
4. Use Elastic Search send files to Kibana through API