Whatever-OZS

Thu Oct. 20 Update

Identified Challenges

- It is a RealTime Project
- The size of the Dataset (~ 4 million rows per hour. ~ 4GB per day)
- Storage capabilities
- Distributed System
- Different tasks (python scripts, MapReduce jobs, Spark, HTML displaying)
- How to automatize tasks? (Reduce humans running each step separately)
- Fast retrieval of huge amounts of information
- Money issues on computing resource

Project's Architecture

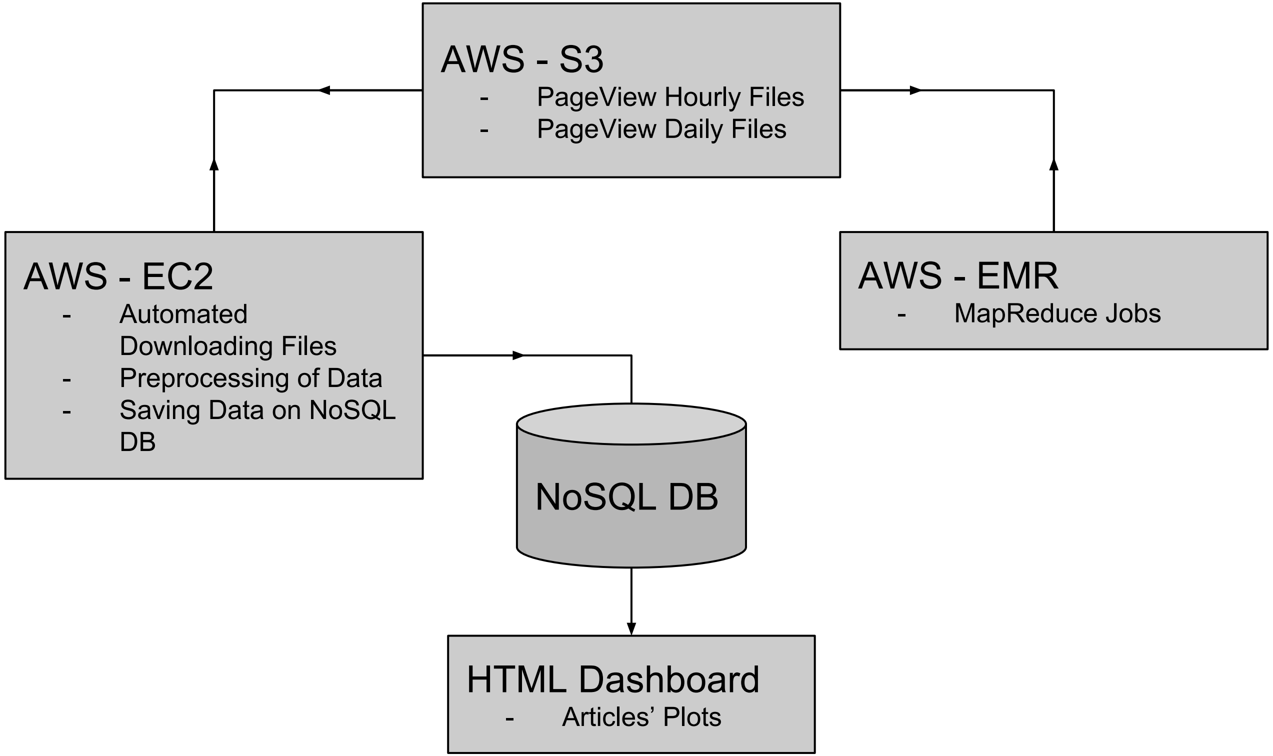
To deploy the project:

- Amazon Web Services

Main Factors:

- Storage and computing services.
- Scalability.
- Interconnection between modules and services.

Project’s Architecture



AWS - S3
- PageView Hourly Files
- PageView Daily Files

AWS - EC2
- Automated Downloading Files
- Preprocessing of Data
- Saving Data on NoSQL DB

AWS - EMR
- MapReduce Jobs

NoSQL DB

HTML Dashboard
- Articles’ Plots

Project’s Workflow

1. Automated Downloading of Pageview files from Wikipedia.
2. Storage on AWS S3 Buckets.
3. Preprocessing / Filtering of Pageview Files
4. MapReduce/Spark jobs to generate Pageviews per Day
5. Save the output data on a NoSQL DB
6. Peak Identification in Pageviews
7. Data displaying through an HTML Dashboard

Auto-Download Script

During this week, we have built our auto-download script which can download the wikipedia page views information every hour automatically.

This script can also automatically decompress the compressed file and put it into the target directory.

The following step is to create a Cron Job that runs the script every day and store the files directly into the AWS S3 Buckets.

Raw Data Analysis

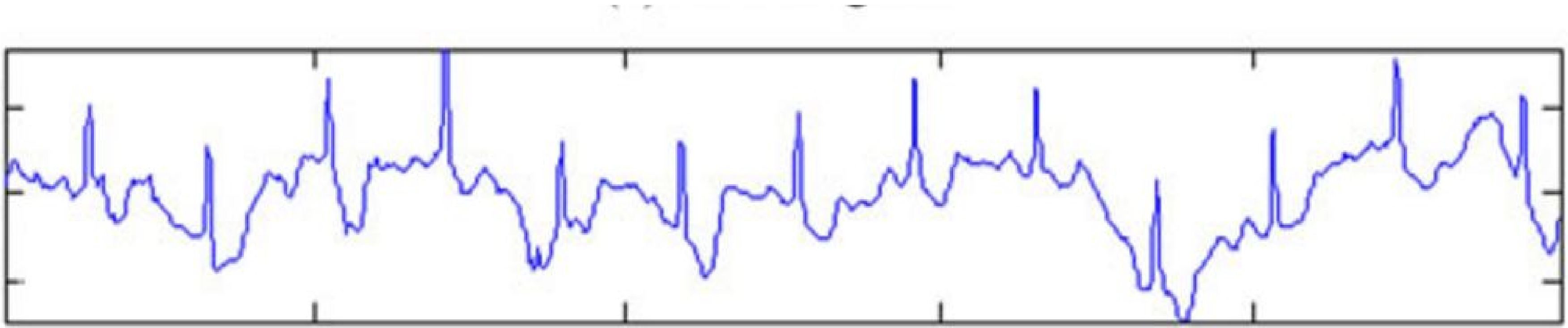
```
en.m Gatti's_Pizza 2 0
en.m Gattilusi 1 0
en.m Gattinara_DOCG 1 0
en.m Gattlin_Griffith 1 0
en.m Gatton,_Queensland 2 0
en.m Gatton,_Surrey 1 0
en.m Gatton_murders 1 0
```

The data is encoded in the following format:

- Language Code
- Wikipedia Project
- Page Article
- Hourly Views
- An unmeaningful ‘0’, which will be figured out later

Trend Analysis Plot for page views

For the next week, we are going to draw a trend plot for every page views just like the plot below.



We have identified the **PeakUtils** library (Python) that will help us in the peak detection task.

Thank you!