STACK OVERFLOW KEYWORD EXTRACTION SYSTEM

GROUP 9

Group Members:

Pavithra Iyer (piyer3)
Sanya Kathuria (skathur2)
Unnati Agrawal (uagrawa)

PROBLEM STATEMENT

Develop a web application that extracts and discovers the trend of keywords, languages or technologies over the desired time period using the Stack Overflow data.

Have a closer contact with modern technologies used to process big data.

Automate the process and store data in a NoSQL database.

Build an application which scales with user traffic.

MOTIVATION

Data **volumes** are getting larger!

Number of data sources is exploding!

Data is arriving faster!

Immense **Variety** in Data!

SCALING UP IS BECOMING INFEASIBLE!

Stack overflow is a much **loved programmer question and answer** site written by two guys nobody has ever heard of before.

Well, not exactly. Created by top programmer and blog stars **Jeff Atwood and Joel Spolsky**.

WHY DISTRIBUTED PROCESSING FOR STATIC DATA ANALYSIS ?

DATA VOLUME AND VARIETY !

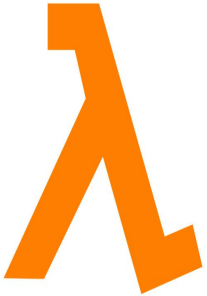
SERVERS CANNOT TAKE IT ANYMORE!

FEATURES OF THE SYSTEM

STATIC DATA ANALYSIS - Building a distributed data processing framework can be a complex task!

- Low Latency
- Scalable
- Faster in-memory Processing
- Storing the results
- Reporting the results real-time on REST API
- Ad-hoc Queries.

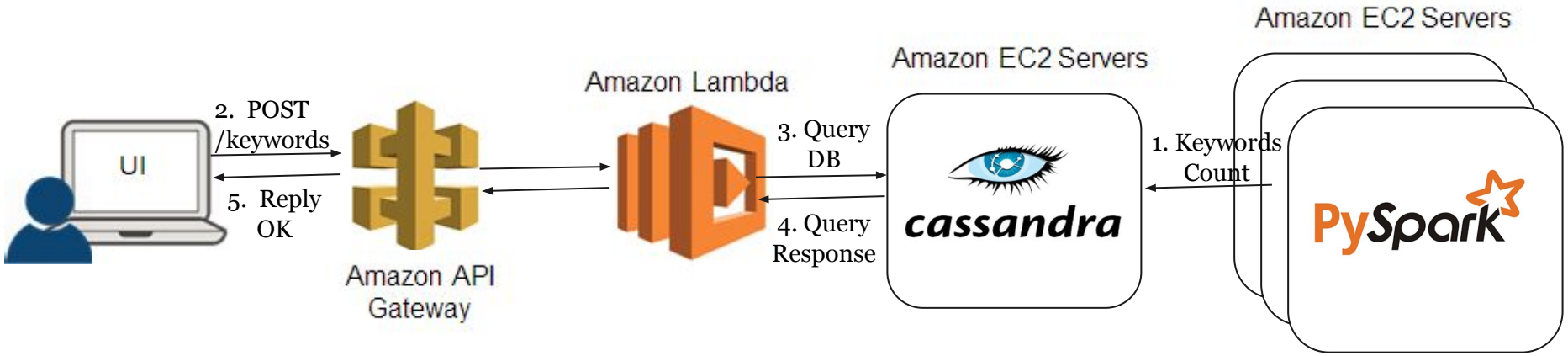
TECHNOLOGIES USED



COMPONENTS

- Infrastructure - AWS
- Data - Kaggle
- Data Cleaning - Python
- Distributed database - Cassandra
- Fast and parallel computations - Spark
- Cloud Infrastructure - AWS EC2
- Scalable serverless computing for backend microservice - AWS Lambda

ARCHITECTURE



STATIC DATA PROCESSING PIPELINE

- Data Cleaning - Kaggle dataset
- Data Transformation - Removing Tags
- Data Processing Spark Approaches-
  - Remove stop words
  - Better approach -
    - Fix keywords as you can’t come up with an exhaustive list of stopwords
    - Spark code flow

WHY SPARK?

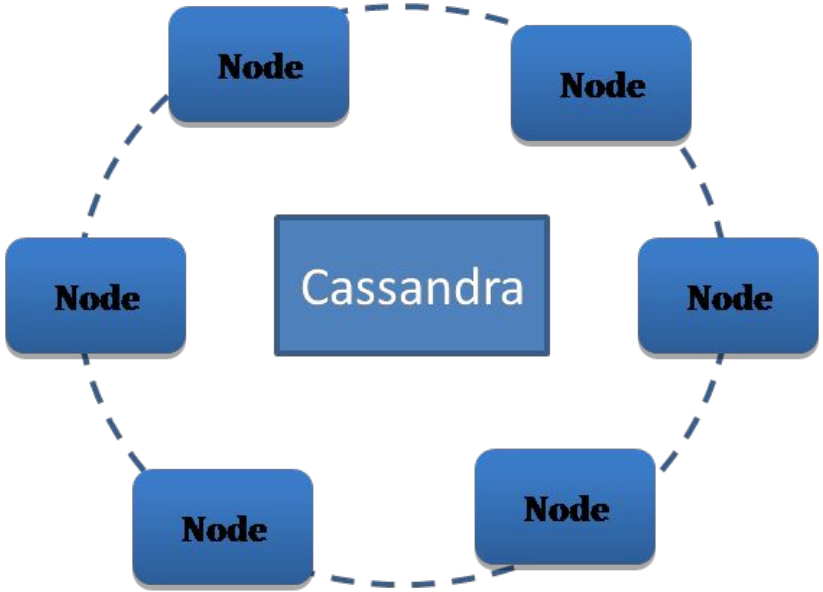
- Distributed in-memory analytic processing.
- Batch and streaming analytics.
- Fast - 10x-100x faster than Hadoop MapReduce.
- Rich Scala, Java and Python APIs

- Previous world record held by Hadoop MapReduce cluster of 2100 nodes, in 72 minutes.
- Spark completed the benchmark in 23 minutes on just 206 EC2 nodes. (2004)

= 3X faster using 10X fewer machines.

WHY CASSANDRA?

- Very fast.
- Extremely Resilient.
- Linear scale and easy to operate.
- Continuous availability and Disaster avoidance.



ISSUES

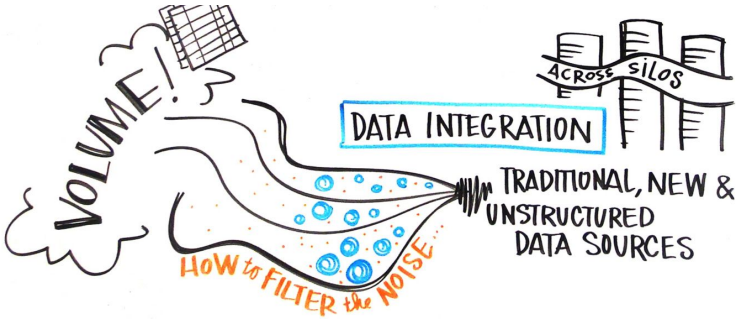
- **Setting up Spark is difficult!**
  - Go for Apache Spark on AWS EMR.

- **Cassandra configuration** to make it accessible from other nodes.

- **Setting up AWS Lambda to interact with EC2 servers**.
  - Added IAM role with EC2 access
  - Configure Lambda and all EC2 instances in same VPC
  - AWS Educate does not give you permissions to edit IAM roles!

CONCLUSION

- Data Preprocessing and Configuration set up took 80% of the time and effort.
- Spark cluster when configured properly is fault - tolerant.
- Spark cluster took hardly half an hour to extract top keywords from our dataset.
- Overall, Distributed Computing was way faster than using single machine for large datasets.



FUTURE WORK

- Stream data from Stackoverflow.
- Handle large data by adding more nodes to Spark Cluster.
- Extract keywords from technical blogs.

POST QUESTIONS ON STACKOVERFLOW AND TEST OUR SYSTEM!