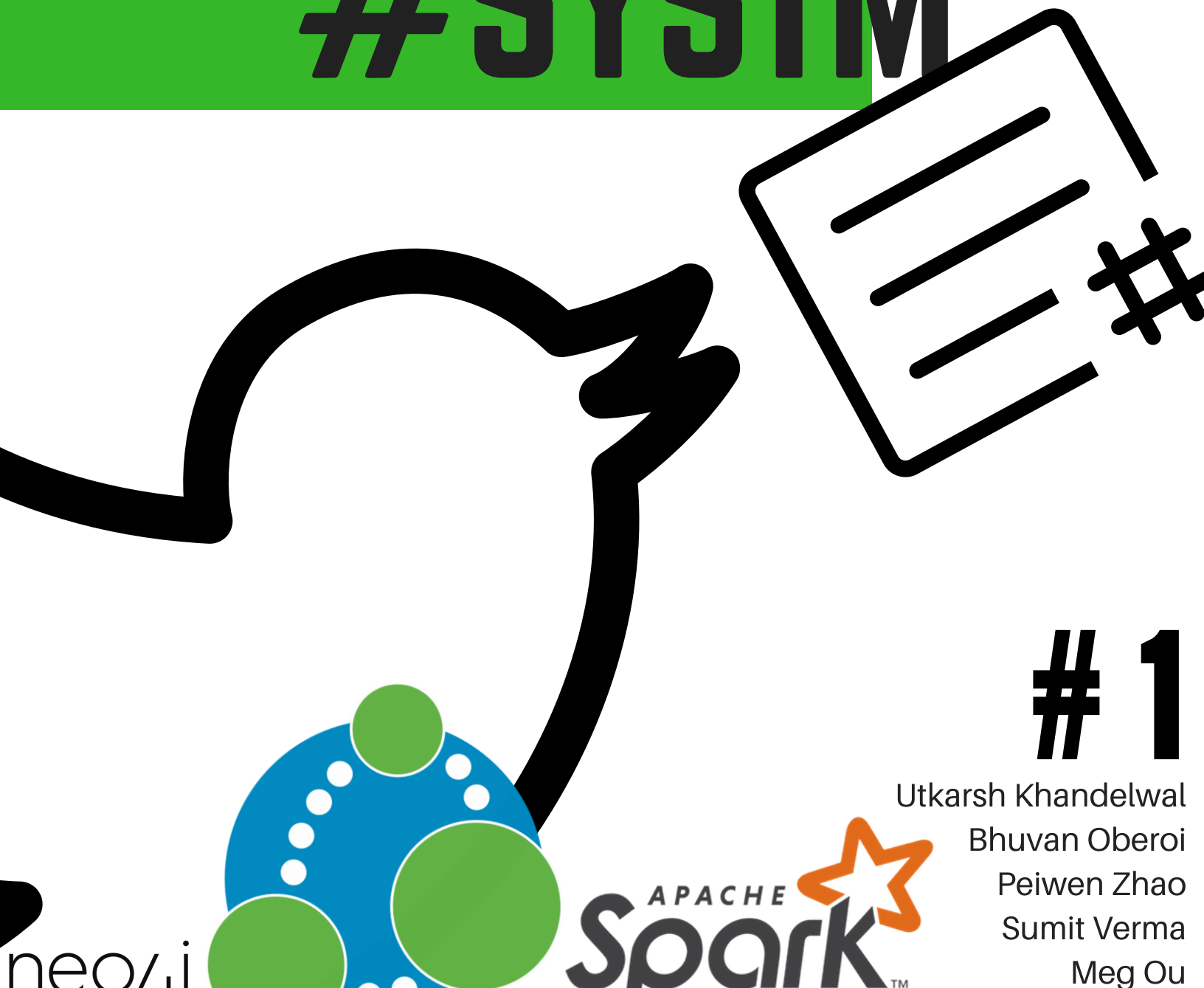
#HASHTAG #RECMND #SYSTM

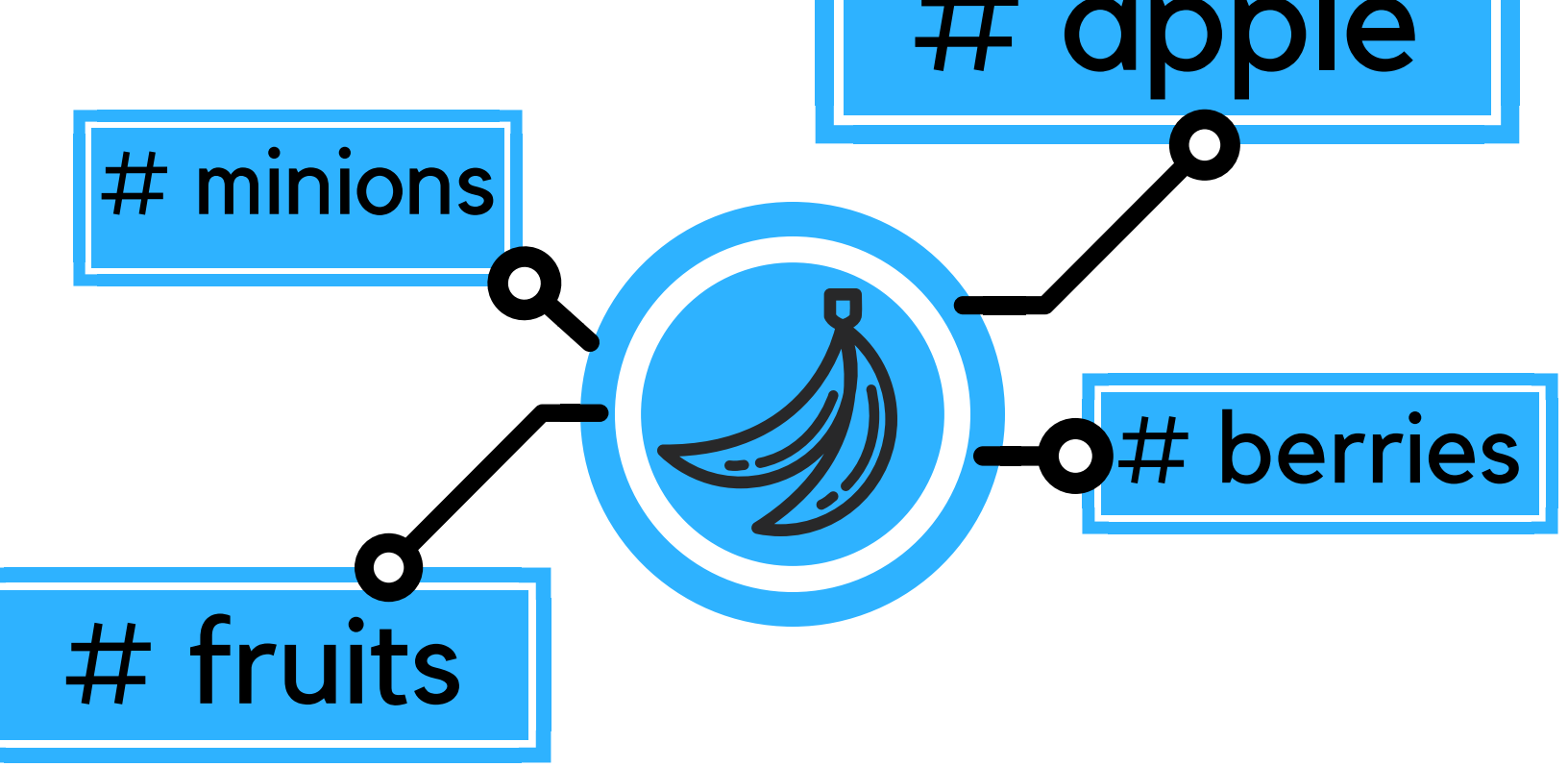


#1

Utkarsh Khandelwal
Bhuvan Oberoi
Peiwen Zhao
Sumit Verma
Meg Ou

neo4j

APACHE Spark™



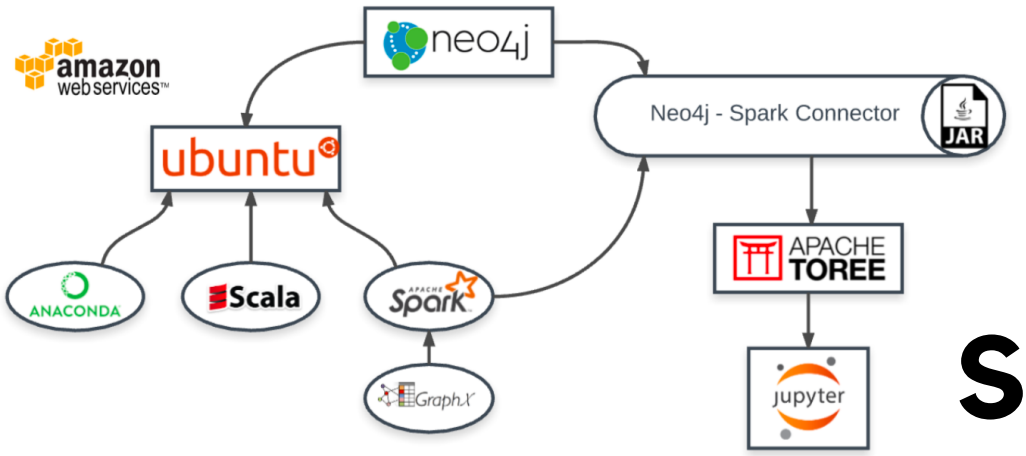
The problem we want to solve

Most social networks only provide hashtag "auto-complete" systems but there are no recommendations. For the user to become part of an intended hashtag group, it requires them to write exactly the right case and spelling in the hashtag.

Our recommendation system finds the most relevant AND trending hashtags based on user's first hashtag as input which allows them to maximize public exposure.

There are two business use cases for this recommendation system:

- **Building a highly scalable product based recommendation system**
- **Understanding what are people associating a particular hashtag, which could be a company or a product, with to gather a high-level sentiment around it**



TECH STACK

Hashtag Recommendation system using Spark GraphX on Amazon AWS

GOLD TEAM 1

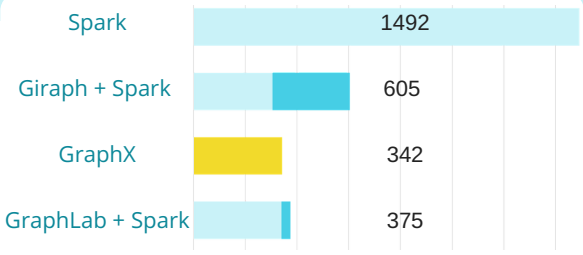
Graph database vs Relational database

Relational Database

- Less connected data
- Only present in tabular format
- SQL language
- Slower for very large datasets

Graph Database

- More connected data
- Graph visualization & tabular form
- Cypher Query Language
- Faster for very large dataset



Timed end-to-end GraphX is *faster* than other graph computing tools

Hashtag recommendation system

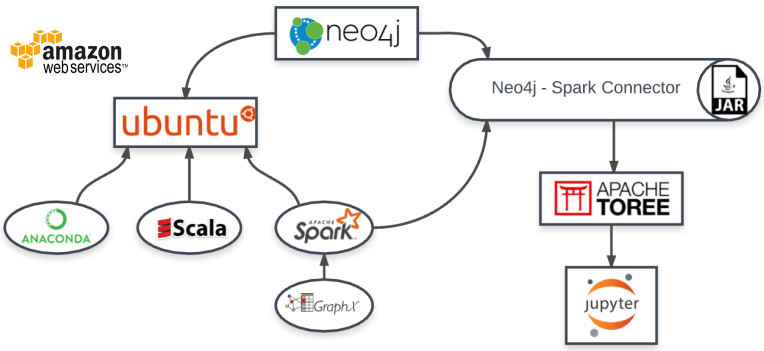
Download Tweets & Hashtag through streaming API

Pre-process raw data in R

Store data in neo4j Graph database (AWS)

Use GraphX -- Apache Spark's API for graphs computation

Parallel computing implement process



To implement hashtag recommendation system on **1.4 million** twitters, we combined graph technology and the power of **Spark** installed on an **AWS cluster** that ensuring scalability of the system.

Anaconda, Scala and Spark are installed on **Ubuntu** Linux cluster. Graph queries are written in **Pyspark Scala** on **Jupyter notebook** and results are stored to a neo4j database.

Neo4j database snapshot

Cypher query language in pyspark:

```
"MATCH (h:Hashtag)<-[:HASHTAG]-
(:Tweet)-[:HASHTAG]->(:Hashtag
{hashtag:{hashtag}})
      WHERE h.hashtag <> {hashtag}
      RETURN h.hashtag AS hashtag,
      COUNT(*) AS count
      ORDER BY count DESC"
```

Bhuvan Oberoi obero013@umn.edu

Peiwen Zhao zhao0885@umn.edu

Yi-Hsiu Ou ou000003@umn.edu

Utkarsh Khandelwal khand061@umn.edu

Sumit Verma verma106@umn.edu

Github Technical Tutorial:
https://github.umn.edu/zhao0885/Bigdata-project-Hashtag-Recommendation-System/blob/master/README.md