YouTube Data Analyzer

Team Kongqueror

Ross Kugler, Benjamin Bordon, Huy (Harry) Ky

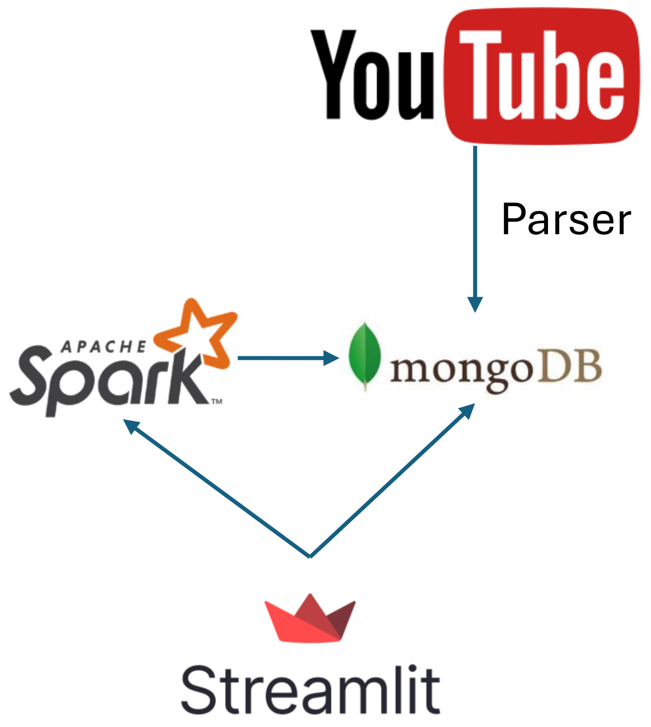
Problem Statement

- Implement a Youtube data analyzer supported by NoSQL database, Hadoop MapReduce, and Spark GraphX/GraphFrame. The analyzer provides basic data analytics functions to Youtube media datasets.
- Efficiently storing, processing, and analyzing large-scale YouTube video network data to extract meaningful insights

Team

- 3 senior Software Engineering students at WSU Everett
    - Ross
    - Ben
    - Harry

Architecture



- Parse the crawls from the dataset
- Insert into MongoDB
- Spark connects to MongoDB via connector
- Python GUI application using Streamlit framework
  - Connects to both MongoDB and Spark
  - Loads precomputed algorithm results from MongoDB
  - Can dynamically run Spark algorithms based on user queries
- Spark Algorithms written in Python with PySpark
- Apache Spark
  - Run in Standalone mode require at least 1 worker with 4GB of memory

Dataset Parsing

- Initially took 2 crawls from the initial dataset: https://netsg.cs.sfu.ca/youtubedata/
- Parse through individual crawls
- Clean to remove malformed/incomplete data
- Create indexes
- Insert into 4 MongoDB collections
  - crawls
  - edges
  - video_snapshots
  - videos
- Raw Data (245 Mb) -> Inserted Data (1.77 GiB)

Spark Cluster

- Spark 3.5.7
- Runs in standalone mode in Windows
- Number of workers based on local resources (# of CPU cores and memory)
- We reserve 4 cores for OS
- Optimized the remaining CPU cores for the Spark worker nodes.
    - Prioritize higher memory per worker than multiple workers with lower memory.

Algorithms & Queries

- Network Aggregation

  A high-level summary of the entire video interaction graph, capturing its scale and connectivity.
- Categorized Video Stats

  Performance metrics broken down by video categories (entertainment, education, comedy, etc.)
- Top-K
  - # of videos in a Category
  - By view
  - By rating
  - By range
- Subgraph Patterns

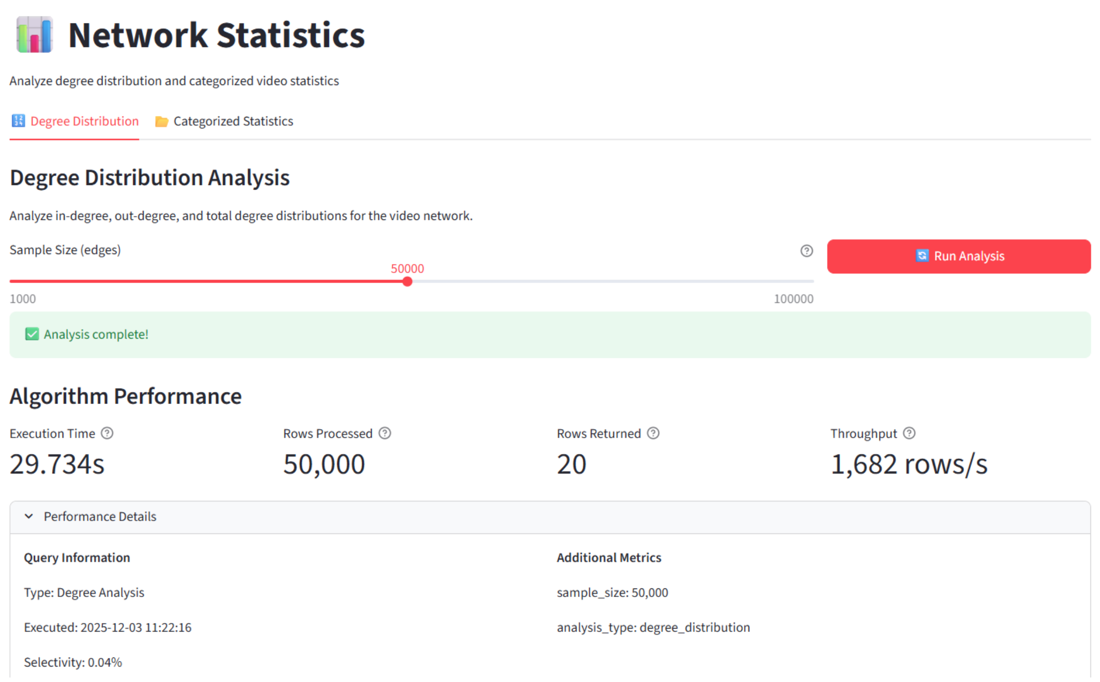
Detects common structural motifs—hubs, chains, cliques, etc.—to uncover hidden audience or influencer dynamics.

- Pagerank

Ranks videos by influence, based on the quality and quantity of connections from other important videos.

Performance Results

- Live performance results provided in app



Analyze degree distribution and categorized video statistics

Degree Distribution Analysis

Analyze in-degree, out-degree, and total degree distributions for the video network.

Sample Size (edges): 50000 (1000 – 100000)

Analysis complete!

Algorithm Performance

Execution Time	Rows Processed	Rows Returned	Throughput
29.734s	50,000	20	1,682 rows/s

Query Information

Type: Degree Analysis

Executed: 2025-12-03 11:22:16

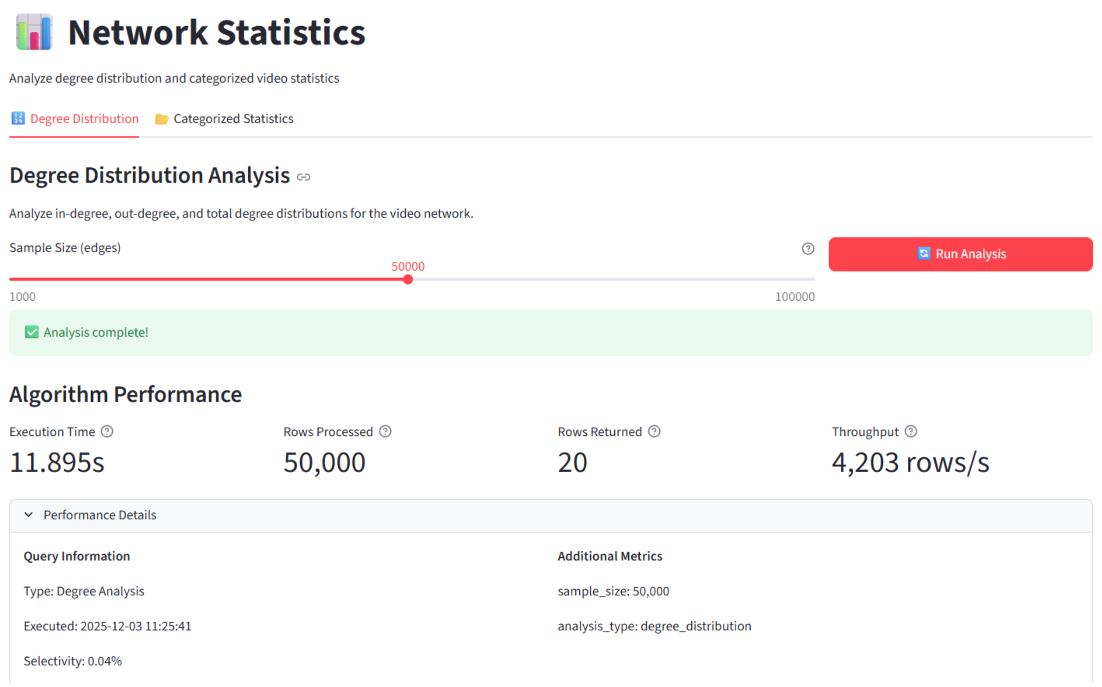
Selectivity: 0.04%

Additional Metrics

sample_size: 50,000

analysis_type: degree_distribution

Cold start (first run)



Analyze degree distribution and categorized video statistics

Degree Distribution Analysis

Analyze in-degree, out-degree, and total degree distributions for the video network.

Sample Size (edges): 50000 (1000 – 100000)

Analysis complete!

Algorithm Performance

Execution Time	Rows Processed	Rows Returned	Throughput
11.895s	50,000	20	4,203 rows/s

Query Information

Type: Degree Analysis

Executed: 2025-12-03 11:25:41

Selectivity: 0.04%

Additional Metrics

sample_size: 50,000

analysis_type: degree_distribution

After caching (second run)

Data Insights

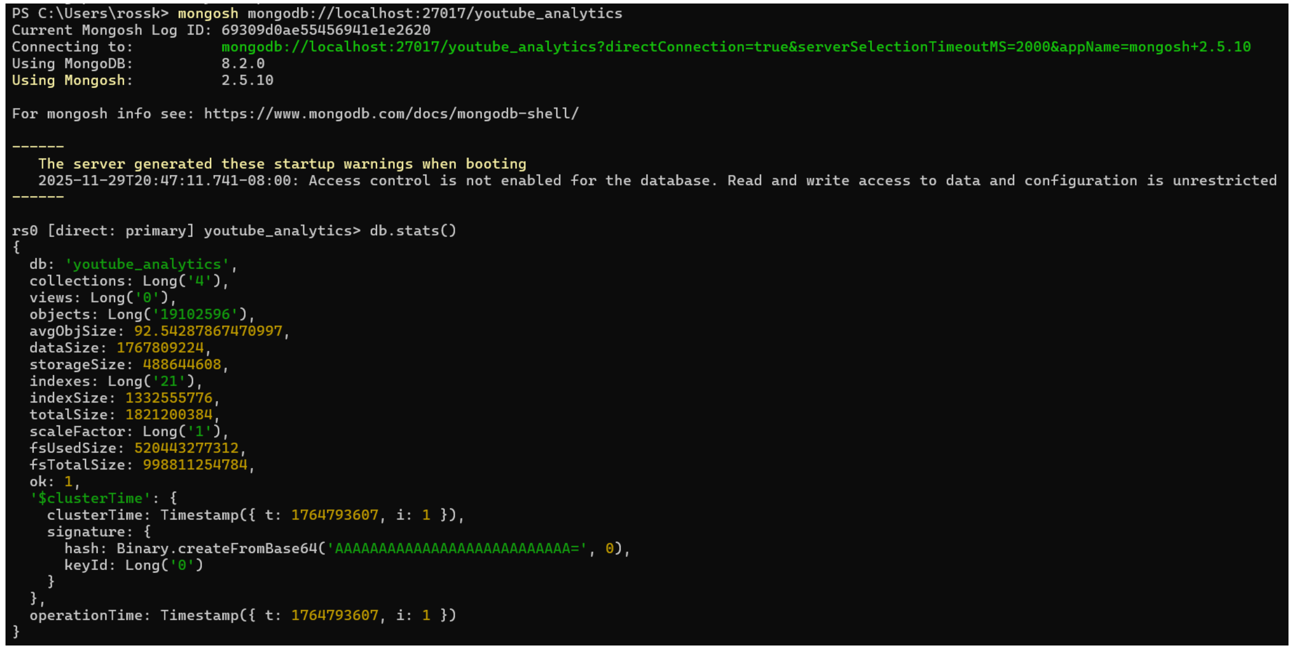
- *Keep in mind this is limited to the small sample of the dataset we used:
    - Music, entertainment are the top category of all time.
    - GRANHERMANO8 has most videos – Spanish Reality TV show, followed by CBS
    - Education was surprisingly one of the top categories, even though online schooling was likely not that popular in 2007-2008
    - Highest viewed video is over 6 million views, compared to current day (December 2025) data of 16 Billion views. https://en.wikipedia.org/wiki/List_of_most-viewed_YouTube_videos

Other Insights

- The dataset we worked with was old (2007-2008)
  - Many videos are unavailable – either violated terms or taken down by uploader
  - Would’ve been more beneficial to do our own crawling with current data
- “una” video category from the crawl dataset?

Lessons Learned

- Probably not do as much indexing as we did (unless we are going full-scale), because it made the dataset expand
- If we had time and resources, actually use Docker containers and deploy the app, with Spark nodes on different machines



```
PS C:\Users\rossk> mongosh mongodb://localhost:27017/youtube_analytics
Current Mongosh Log ID: 69309d0ae55456941e1e2620
Connecting to:          mongodb://localhost:27017/youtube_analytics?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.10
Using MongoDB:          8.2.0
Using Mongosh:          2.5.10

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-11-29T20:47:11.741-08:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

rs0 [direct: primary] youtube_analytics> db.stats()
{
  db: 'youtube_analytics',
  collections: Long('4'),
  views: Long('0'),
  objects: Long('19102596'),
  avgObjSize: 92.54287867470997,
  dataSize: 1767809224,
  storageSize: 488644608,
  indexes: Long('21'),
  indexSize: 1332555776,
  totalSize: 1821200384,
  scaleFactor: Long('1'),
  fsUsedSize: 520443277312,
  fsTotalSize: 998811254784,
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1764793607, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1764793607, i: 1 })
}
```

Thank you

- Questions?