Project Statement for Milestone 1

Konqueror

Ross Kugler, Huy (Harry) Ky, Ben Bordon

Project Report Topics:

1. Problem Statement and Project Goals:
   a. Give a formal description of the project. Include a description of the dataset and its links.

      Project: Youtube Analyzer

      Related Industry: Social Media

      Dataset: "Statistics and Social Network of YouTube Videos" by Xu Cheng and Cameron Dale and Jiangchuan Liu

      Dataset Description: a dataset containing several early large-scale crawls of YouTube's video network. The crawler used breadth-first search up to a max of 20 levels. Each video is a node; an edge from video A → B exists if B appeared in the top 20 "related videos" list of A. For the purposes of this project, we will only be using the data from the first few crawls of this dataset.

      Dataset Link: http://netsg.cs.sfu.ca/youtubedata/

      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. The analyzer provides following functions for users:

      • Network aggregation: efficiently report the following statistics of Youtube video network:

         - Degree distribution (including in-degree and out-degree); average degree, maximum and minimum degree

         - Categorized statistics: frequency of videos partitioned by a search condition: categorization, size of videos, view count, etc.

      • Search:

         - Top k queries: find top k categories in which the most number of videos are uploaded; top k rated videos; top k most popular videos.

- Range queries: find all videos in categories X with duration within a range [t1, t2]; find all videos with size in range [x,y].

- User identification in recommendation patterns: find all occurrence of a specified subgraph pattern connecting users and videos with specified search condition.

• Influence analysis:

- Use PageRank algorithms over the Youtube network to compute the scores efficiently. Intuitively, a video with high PageRank score means that the video is related to many videos in the graph, thus has a high influence.

- Effectively find top k most influence videos in Youtube network. Check the properties of these videos (# of views, # edges, category...). What can we find out? Present your findings.

Note:

- Where applicable, develop effective optimization techniques to speed up the algorithm you used, including indexing, compression, or summarization.

b. Why is the problem you want to address important? What s its application?

The project is solving the problem of efficiently storing, processing, and analyzing large-scale YouTube video network data to extract meaningful insights (trends, influence, user patterns), which cannot be done effectively with traditional data management tools. For example, we can determine trends among the top k most influential videos in the YouTube network (e.g. "They are all tech videos").

This problem is important because YouTube is a massive, socially and commercially influential network, and being able to analyze it efficiently enables applications in recommendation systems, trend discovery, marketing, research, and content moderation.

c. Specify the goal you want to achieve (an end-to-end application with a graphical user interface, and/or a research-based evaluation of existing and new algorithms).

Primary Goal — End-to-End Application with GUI

If we have enough time, we can build a small REST API (Flask/FastAPI) + minimal dashboard to trigger jobs and show tables/plots. Otherwise, we will just create CLI + Jupyter/Databricks notebooks that call Spark jobs and query NoSQL.

Team Description:

a. Who are the team members? What knowledge and skills do the team have from previous courses, projects, or internships?

We are all software engineering majors. We have all taken CPTS451, so we have experience with relational databases, SQL, PostGreSQL. We have done several projects utilizing databases, like Microsoft Azure SQL Server and DB, Azure Storage Account Tables, and SQLite. We are experienced in several coding languages, including Python, C#, C++, Java, etc.

Ben took WebDev and has experience with Firebase which is a baas (backend as a service).

b. What will be each team member's roles and responsibilities in the current project?

Ross Kugler – Liaison, Developer, research Hadoop MapReduce

Huy Ky – Main Database Manager, research NoSQL Database (MongoDB)

Ben Bordon – Developer, Documentation, research Spark GraphX/GraphFrames

2. Dataset Type and Data Model:

a. A dataset can be described with more than one data model. Evaluate the dataset against **each** of the following data models, reporting its statistics:

i. Key-Value: Can the data be represented with a key-value data model? If so, how many key-value pairs represent the dataset? How many unique keys? What are the data types for keys and values? Are these basic data types or data structures?

Yes, the data can be represented with a key-value data model.
Key-value 1: key = video_id (string), value = structured record for the video (json)
Key-value 2: key = user_id?, value = a nested list of video ids associated with that user id (array of strings)

ii. Graph: Can the data be represented with a graph data model? If so, how many nodes and edges represent the dataset? How many attributes are there for the nodes/edges? Is it labelled? Directed?

Yes. The graph data model is a great way to represent this data.
Number of nodes: One per video. There is a rough range of 40,000 – 150,000 videos per crawl.
Number of edges: maximum of 20 ("If a video *b* is in the related video list (first 20 only) of a video *a*")
Number of attributes: 9 (uploader, age, category, length, views, rate, ratings, comments, related IDs)

Labelled: Yes (node = video id, edge = related video)
Directed: Yes (a->b if b is in a's related video list)

iii. Document: Can the data be represented with a document data model? If so, how many documents represent your dataset? How many elements / sub-elements does each document has? What are the attributes?

Yes, each video can be described as document data. So for our purposes, only one type of document can represent the dataset (videos). For the related-ids element, there are up to 20 sub-element videos referenced.

Video Document Attributes:

```
{
  _id: "video_id",          // string
  uploader: "username",     // string
  age_days: int,            // int
  category: "string",       // string
  length_sec: int,          // int
  views: long,              // long
  rate: double,             // float
  ratings: long,            // long
  comments: long,           // long
  related: ["id1", ...],    // array<string> up to 20
}
```

Example video document:

```
{
  "_id": "6_g2i62q_fA",  // The video ID serves as the unique document key
  "uploader": "username123",
  "age_in_days": 600,
  "category": "Music",
  "length_seconds": 245,
  "views": 150000,
  "rate": 4.85,
  "ratings_count": 9500,
  "comments_count": 1200,
  "related_ids": [        // The list of related IDs fits perfectly in an array
    "_nS0L5k3y1c",
    "9_g2i62a_zB",
    "7_h3j73b_xV"
  ]
}
```

iv. Other: Can the data be represented by any other non-relational data model? If so, describe the data model and the applicable statistics on your dataset.

Some other non-relational data models that the data can be represented by:

Wide-column / Column-family (e.g., HBase/Cassandra):

- Row key: video_id
- Column families: meta (static fields), links (related IDs), ts (wide row of per-date updates)
- Stats: 1 row per video; up to 20 qualifiers in links; 7+ weekly columns in 2007 updates; 21 in 2008 updates.
- Excellent for time-series updates and fast scans by key prefix; integrates well with Hadoop jobs.

Search index (Lucene/Elasticsearch) as an auxiliary store:

- One indexed document per video for category, title (if added), uploader, length, views, etc.
- Powers top-k and range filters efficiently; not a primary store but a critical acceleration layer.

b. Describe the data model your team has chosen to represent the dataset? Justify why the data model is an appropriate one for the dataset. Note: You should be using a non-relational data model for this project.

Data model:

Document Store for MongoDB (storage)

Justification: MongoDB is well-suited for storing video metadata as documents and supports indexing that enables fast execution of top-k queries and range queries. Optimized for Apache Spark.

Property Graph for Spark GraphX/GraphFrames (analytics)

Justification: YouTube data is inherently graph-structured in which you could attach attributes to vertices and edges. Allows us to do degree distribution and PageRank and use Spark tools.

3. Tools:
   a. What database tools do you plan to use?
      MongoDB for document-based database.

   b. What data processing tools do you plan to use?
      Apache Spark and Hadoop MapReduce are our two data processing engines. We will experiment with their performance and the analytics they provide. They essentially do the same thing (data processing), but through different methods/approaches. Spark GraphX/GraphFrame for graph analytics.

   c. What cloud resources, if any, do you plan to use?
      - MongoDB Atlas for cloud database storage
      - Firebase to store the documents as collections

      (but it is important to note that we will try to implement the database and application locally first before deploying anything to cloud)

Appendix

Tool Notes (for our personal understanding):

What GraphFrames Will Do:

Model the YouTube network as a graph

- Vertices (nodes) → videos (with attributes: uploader, category, views, etc.)
- Edges → "related-to" relationships between videos (directed links)

Run graph algorithms efficiently on large data

- Degree stats → in-degree, out-degree, average/max/min degree of videos.
- PageRank → compute influence scores of videos and find the top-k most influential.
- Connected components / community detection → discover clusters of related videos.

Support graph queries & pattern matching

- Example: Find all users who uploaded videos that connect to a video in category X through two "related-to" hops. This is hard to do with plain SQL or MapReduce, but GraphFrames has a motif-finding API for this.

Integrate with Spark SQL and DataFrames

- You can easily combine graph analytics with tabular queries (e.g., filter only "Music" category videos, then run PageRank).
- Helps when you need both network analysis and data filtering/search together.

In your YouTube Analyzer project, Hadoop MapReduce will handle the batch-style, large-scale data processing tasks that are more tabular/aggregate in nature (not graph-specific).

What MapReduce Will Do:

Basic Aggregations

- Count videos per category, uploader, or rating bucket.
- Compute statistics like average views, max/min duration, etc.

Top-k Queries

- Find the top-k most viewed videos.
- Find the top-k categories with the most uploads.

Range Queries (with filtering)

- Select all videos within a duration range [t1, t2].
- Select all videos with size/views in a range [x, y].

Data Preparation / Cleaning

- Preprocess raw dataset into structured form (e.g., JSON, key-value pairs) before storing into MongoDB or using with Spark.

Why It's Useful in Your Project

MapReduce is efficient for bulk operations on very large datasets. Your project requirements explicitly list tasks like:

- "Top k queries"
- "Range queries"
- "Categorized statistics"

These map directly to MapReduce jobs.

It complements GraphFrames:

MapReduce → fast tabular analytics & summaries.
GraphFrames → advanced graph/network analytics.