A Hadoop-driven Data Analysis System

Ashish Jindal
Rutgers University
Piscataway, NJ, USA
Email: ashish.jindal@rutgers.edu

Yikun Xian
Rutgers University
Piscataway, NJ, USA
Email: siriusxyk@gmail.com

Sanjivi Muttena
Rutgers University
Piscataway, NJ, USA
Email: sanjivi.muttena@rutgers.edu

***Abstract*—**The complexity of modern analytics needs is outstripping the available computing power of legacy systems. Distributed system like Hadoop compliments this requirement for storing and analyzing huge sets of information by providing a platform for parallel processing of large data sets stored over multiple machines. This project aims to setup a Hadoop infrastructure and demonstrate its power for big data analysis. Some machine learning models will also be deployed in this system so that developers can directly call corresponding APIs to execute training and testing models.

***Index Terms*—**Hadoop-driven, Data Analysis, Infrastructure

I. Project Description

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. Hadoop stores data as it comes in - structured or unstructured - saving time on configuring data for relational databases. In our project we will store a large dataset in Hadoop file system and use it for analysis of events from various sources like news, twitter etc. Our project falls in the category *Scalable Algorithms Infrastructure*. The main stumbling points for this project are identifying a large dataset for Hadoop, configuring Hadoop for multiple clusters and building a data analytics system over Hadoop.

The project has four stages: requirement gathering, design, infrastructure implementation, and user interface.

A. Stage 1 - The Requirement Gathering Stage

Following are the deliverables for this stage:

a) General Description: This project is about setting up an infrastructure for big data analytics using Hadoop and demonstrate a data analytics application using event data from various sources like news and blogs, etc. The first part is to construct a distributed computing platform based on Hadoop and its high-level applications like Mahout. The system provides some well-encapsulated APIs for training and testing general models. The second part include a real application on event analysis driven by Hadoop. It mainly demonstrate the feasibility of architecture to apply Hadoop into existing web-based application.

b) User Type 1: People interested in analyzing data that can't be stored on a single machine.

- User Interaction Modes: Hive query language.
- Real World Scenarios:
  - Scenario 1 Description: Processing of large server logs.
  - System Data Input for Scenario 1: Log files.
  - Input Data Types for Scenario 1: Structured log files.
  - System Data Output for Scenario 1: Information based on query.
  - Scenario 2 Description: Text mining.
  - System Data Input for Scenario 2: Big sources of textual information.
  - Input Data Types for Scenario 2: Text.
  - System Data Output for Scenario 2: Information based on query.

c) Project Timeline and Division of Labor: We will start with studying about Hadoop and map reduce systems and then set up a Hadoop system on local host. When we are successful in doing that we will find a small dataset, large enough to fit on a single system and start test analysis over it using Hadoop. Then we will repeat the same process by setting up Hadoop in pseudo distributed mode. Parallely two team members will start building the demo analytics application over Hadoop. In the end we will setup Hadoop on multiple clusters and deploy our application over it.

- Week 1 and Week 2: Work on requirement gathering and design of system using Hadoop. Setup Hadoop on local host as a single cluster and perform dummy analysis to test its functionality.
- Week 3 and week 4: Setup Hadoop infrastructure over multiple clusters and create an analytics application demonstrating big data analysis using Hadoop.
- Week 5: Test the application and prepare the necessary documentation & project report.

B. Stage 2 - The Design Stage

The project is divided into three parts. Infrastructure module involves Hadoop deployment, data storage management and integration with data analysis models and web application. Analytic module is responsible for processing large data set based on Hadoop clusters and provides interfaces for access to models. UI module is constructed along with a web application to read and visualize data from backend. The general architecture consisting of these three modules is shown in figure **??**.

1) Infrastructure Module:

2) Analytic Module:

a) Short Project Description:

This part focuses on data management and analysis based on Hadoop infrastructure. Data sources includes Wikipedia

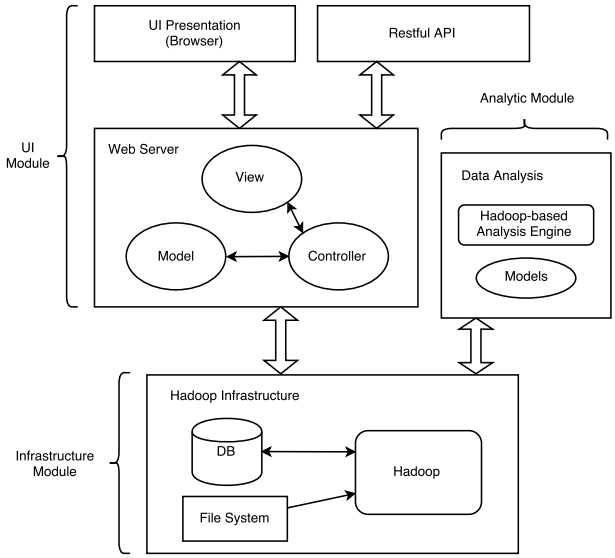

Fig. 1. System Architecture and Modules

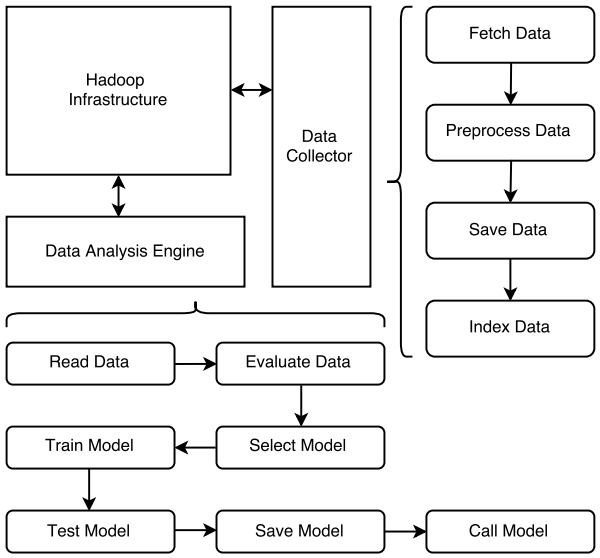

Fig. 2. Data Analysis Flow Diagram

[1] and news websites [2] all in the form of text information. Hadoop platform is used to store these large data sets and support data analysis accessed by interfaces described in infrastructure module. Specifically, analysis part in this project mainly provides encapsulated interfaces for training, testing and calling models including ranking algorithm, similarity measure algorithm and topic model. In this project, the event is generally defined as a sequence of text information belongs to similar topics. Therefore, ranking is designed for evaluating topic trend based on Wikipedia links and flow. Topic model is used to reduce dimension of textual information and transform into a vector so that can be easily measured similarity.

b) Flow Diagram:

As figure **??** shows, the module provides two functions, data crawler and data analysis engine. Data crawler manages data input and output by interacting with data sources and Hadoop infrastructure. Data analysis engine based on Hadoop computing resources provides strong modeling function to analyze textual data.

c) High Level Pseudo Code System Description: The corresponding pseudo code for work flow of data management is shown as follows. The first one describes a basic procedure to retrieve data, clean data and store data.

1) Extract textual data from links
2) Batch clean and filter data
3) Save data to Hadoop platform
4) Index data for future search

The second one gives a general illustration of modeling data for any algorithms used in this project including ranking, similarity measure and topic models.

1) Read data from Hadoop
2) Perform basic statistics on dataset
3) Select suitable model on requirement
4) Train model on Hadoop
5) Test model by evaluation index
6) Save model including parameters to database
7) Return model interface

d) Algorithms and Data Structures:

Two major algorithms in the project are ranking and similarity measure with topic.

Firstly, algorithm **??** describes basic procedure to rank web links. The input is graph data file where each line represents a connection between two web links, so the whole data set can be regarded as a directed graph.

Algorithm 1 Ranking

```
1: function RANK(links)
2:     Extract features from links
3:     for each link i in links do
4:         Represent link i as a vector
5:         Calculate weight w_i for link i
6:     Normalize w_i for every link i
7:     Sort links by weight w
8:     return sorted links
```

Secondly, algorithm **??** describes how to compare similarity between web pages with respect to their topics. The input data set is a collection of textual documents and the output is a set of clusters where documents with similar topics are in the same cluster. Data here is mainly represented by vectors and matrix.

3) UI Module:

a) Short Project Description:

The third phase of the project involves designing the UI for the user to access the trending data. The overall UI can

[1] https://dumps.wikimedia.org/
[2] https://developers.google.com/news-search/

Algorithm 2 Measure similarity

function RANK(*documents*)
  for each d in *documents* **do**
    Represent d as a vector by word counting
  $result$=TOPIC-MODEL(documents in vectors)
  Extract topic distribution of each document from result
  $clusters$=CLUSTERING(documents in topics)
  return *clusters*

be broken down into three segments. The first segment is the home page. The user will have two options. The first option would be to go for the current trending topics irrespective of the category. There will also be another search bar that allows the user to search for a specific topic. The search bar will also provide suggestions as the user types. These suggestions will be the topics that have been indexed in the database.

The second segment of the UI will be the listings of the topics that the user has requested. These can be either the current trending topics or the topic the user had searched. Once these listings have been generated, the user will have options to choose the source from which the data had been generated and go to that news source. The user also has options to change the source if multiple news sources are available. The final operation that the user can perform is to visualize the data stats on the topic.

The third segment of the UI is the data visualization. These stats will be the data obtained from the Wikipedia page of that topic. It would in essence visualize the number of page views over a period of time and would also provide a short description of the event whenever there is a sudden increase in the amount of traffic to the page. If location data is available, we also plan to show the regions in which the topic in question had generated interest.

b) Flow Diagram:

The figure **??** provides a general overview of the overall working of the UI. Each colored nodes represent webpages and the colorless nodes represent the actions the user can perform on them. The green nodes represent external webpages and the yellow nodes represent the webpages on the UI. The transition between the pages occur when the user clicks or interacts with the option in the colorless boxes.

c) High Level Pseudo Code System Description:

Algorithm 3 UI pseudocode

procedure HOMEPAGE
  $initialpage \leftarrow homepage$
  if $getinputfrombutton = true$ **then**
    $navigate \leftarrow Display("trending")$;
  else if $getinputfromtextbox = true$ **then**
    $navigate \leftarrow Display(textbox.text)$;

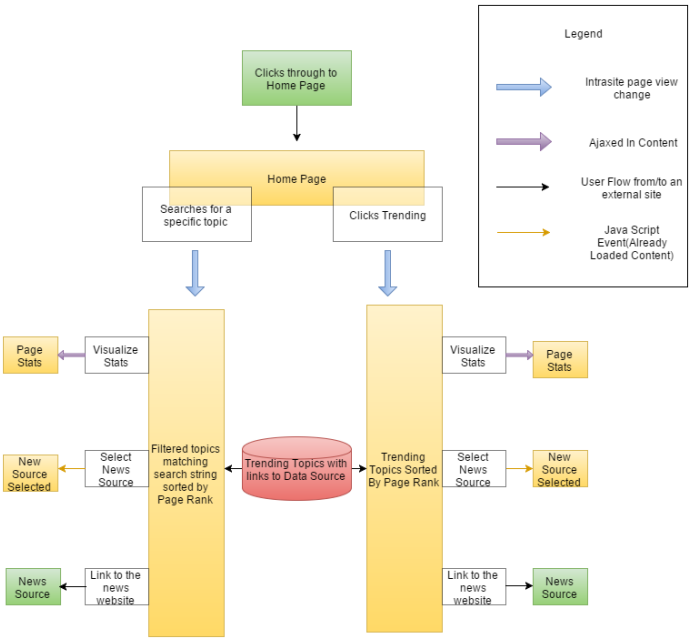

Fig. 3. UI Flow Diagram

Algorithm 4 Display

procedure DISPLAY(contenttodisplay)
  if $content = trending$ **then**
    Display topics sorted by page rank;
  else
    Search Topics that match search string;
    Sort them by Page Rank;
  if $user.choice = visualize$ **then**
    Display Visualization for Page Traffic
  else if $user.choice = gotosource$ **then**
    $navigate \leftarrow source$;