On-line advertisement click prediction - Project proposal

SAHITI KORRAPATI[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: sakorrap@iu.edu, S17-IR-2013*

S17-IR-2013, May 04, 2017

This project aims at predicting the most suitable advertisements to be displayed on the web pages. Advertisements are selected based on the relevance. Relevance factor is calculated by ranking each ad based on the likelihood of clicking the ad if displayed. Data is obtained as CSV files from Kaggle Data sets and is stored in Hadoop Data File system(HDFS). In this project, Ansible along with Cloud mesh is used to deploy cloud architecture and the necessary soft wares on Chameleon cloud. For data exploration and analysis, as the dataset is huge in the order of 100s of gigabytes Apache Pig on Hadoop is chosen. The Pig engine in can execute the data flows in parallel by making use of Hadoop MapReduce framework.

Keywords: Hadoop, Ad click Prediction, BigData, Ansible, Chameleon cloud, Apache Pig, HDFS, MapReduce

https://github.com/sakorrap/sp17-i524/tree/master/project/S17-IR-2013/report

1. INTRODUCTION

It has been analyzed that an average American spends about 23 hours per week surfing on-line [1]. This on-line user activity is being captured by companies to perform analyzes for advertisement, recommendations and many other purposes. This has given rise to the field of "Web Analytics" and one such application is Ad Click prediction.

Many measures are available to asses the ad performance. One popular measure to asses the immediate ad response is click-through rate (CTR) of the advertisement [2] which is defined as the ratio of number of clicks on an ad to the number of times the ad is shown, expressed as a percentage. In this project, I attempted to make use of CTR along with ranking each ad based on the level of relevance to the original web page. For calculating the CTR, historical data of user activity needed to be explored. [3].

The user activity data from web that is used for prediction is enormous. Every page view, advertisement and click is being tracked by web browsers, and that level of data for millions of users potentially generates enormous volumes. The current internet giants use big data technologies to handle such large volumes of data. The current project uses Apache pig along with Hadoop for analyzing and for prediction. Hadoop is a large scale data processing system which runs on parallal processing to handle huge volumes of data. Pig is a high level language which runs on top of Hadoop's HDFS infrastructure. Pig Latin scripts are simple to comprehend and SQL-like queries which can process large volumes of data.

Ansible along with Cloudmesh is used to deploy the software and chameleon cloud for running virtual machines. Ansible is a cloud automation tool which allows easy deployment and configuration of multiple servers in one step. Cloudmesh allows us to deploy and install Hadoop instances on a cluster with any number of nodes.

2. BACKGROUND

2.1. About Data

The data set in this paper is taken from kaggle datasets. It is released by Outbrain which has 2 Billion page views and 16,900,000 clicks of 700 Million unique users, across 560 sites [4].

The dataset contains a sample of users' page views and clicks, as observed on multiple publisher sites in the United States between 14-June-2016 and 28-June-2016. Each viewed page or clicked recommendation is further accompanied by some semantic attributes of those documents [4]. It contains numerous sets of content recommendations served to a specific user in a specific context. Each context (i.e. a set of recommendations) is given a display_id. In each such set, the user has clicked on at least one recommendation. Our task is to rank the recommendations in each group by decreasing predicted likelihood of being clicked [4].

Each user in the dataset is represented by a unique id (uuid). A person can view a document (document_id), which is simply a web page with content (e.g. a news article). On each document, a set of ads (ad_id) are displayed. Each ad belongs to a campaign (campaign_id) run by an advertiser (advertiser_id). Figure 1

shows the fields in our dataset. Metadata about the document is also provided, such as which entities are mentioned, a taxonomy of categories, the topics mentioned, and the publisher [4].

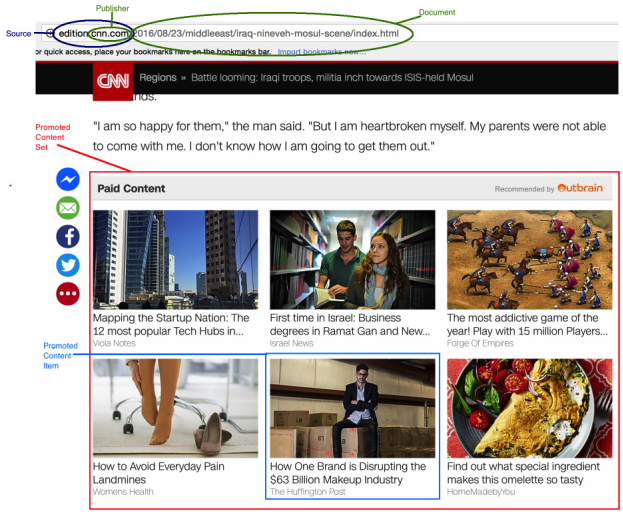

Fig. 1. Displaying Source, Publisher, Document, Promoted content set and items [4]

2.2. Ansible and Cloudmesh client toolkit

Ansible is an opensource cluster management tool which has the ability to maintain a fully immutable server architecture and design. It is a simple automation engine that automates cloud provisioning, configuration management, application deployment, intra-service orchestration [5]. It doesn't use agents or custom security infrastructure, so it's easy to deploy by using "YAML-language" (YAML, in the form of Ansible Playbooks).

Using Ansible, users can define any number of IP Addresses in hosts file, and create custom groups for specific tasks. These group names can be referred in the YAML file to perform specific set of tasks in each of the virtual machines. Ansible also has options to define specific roles for each task and make YAML files dynamic using global variables.

With Cloud mesh, the deployment becomes even simpler and easier. Cloud mesh client toolkit, a lightweight client interface to access heterogeneous clouds, clusters, and workstations, available as API, commandline client and commandline shell. Their quick start user manual has everything that is needed to start using the tool [6].

The current project uses Cloudmesh's "cm" command to perform multiple tasks including deployment of cluster, installing and configuring Hadoop and Pig, enabling the cloud nodes to be able to ssh with each other and assigning floating IP addresses to each node. The configuration of the cluster can be defined in Cloudmesh's YAML file and accordingly the cluster will be created with the given OS and type.

2.3. Hadoop in Big-data

Hadoop is a distributed platform designed to help manage and analyze massive data-sets that are too big or costly to put in relational databases. Rather than storing and processing the whole data in one high-end computer, data can be spread across many smaller nodes. By spreading the data across multiple nodes, the storage as well computational speed can be scaled by parallel processing [7].

Hadoop framework is developed for distributed processing of large data sets across clusters of computers using simple programming models. It can scale up from single servers to thousands of machines, each offering local computation and storage. The library is designed to deliver a highly available service on top of a cluster of computers, each of which may be prone to failures [8].

2.4. HDFS and MapReduce

Hadoop Distributed File System (HDFS) and the MapReduce parallel processing framework are two primary components at the core of Apache Hadoop. Figure 2 shows the high level architecture of Hadoop in regards to HDFS and MapReduce

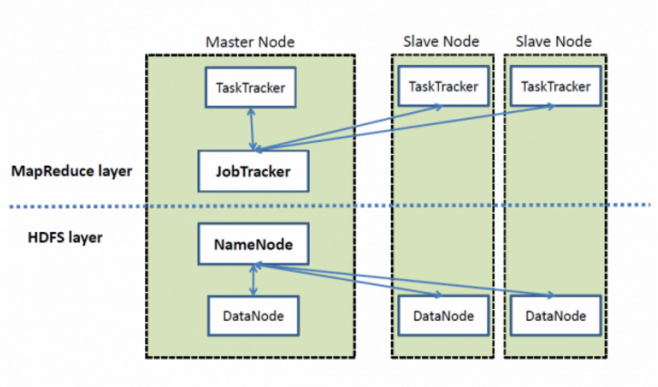

Fig. 2. High Level Architecture of Hadoop [9]

HDFS is a distributed, scalable, and portable file-system for Hadoop framework. Files are broken into blocks and spread across nodes in the cluster in HDFS. The blocks are also replicated on different nodes so that even if one of the nodes fails another one of the live nodes still has a copy of the data. There is a NameNode and DataNodes. The NameNode maintains the references on the file split up in blocks across nodes in the cluster. While reading, client process contacts the NameNode for this metadata and ask the corresponding DataNodes for those blocks for reading [7]. MapReduce is for processing files stored in a distributed environment like HDFS. A typical MapReduce application has two functions, a Mapper and a Reducer. Mappers and Reducers run as tasks on nodes in the cluster. Two processes JobTracker and TaskTracker manages MapReduce framework. The JobTracker is the master process that coordinates the Map and Reduce tasks sent across the cluster.

2.5. Apache Pig

Pig platform is designed to work with large data sets for analysis. The Pig dialect is called Pig Latin, and the Pig Latin commands get compiled into MapReduce jobs that can be run on a suitable platform, like Hadoop. Figure 3 illustrates how it makes use of MapReduce framework.

Pig is a high level language which is similar to SQL in syntax. The structure of Pig is designed such that the Pig scripts are made amenable for parallel processing. Pig scripts are automatically optimized so that the programmers can focus on semantics rather than efficiency [10]. Pig also has options to create user defined functions to perform complex tasks using Pig.

2.6. Chameleon cloud

Chameleon Cloud is a cloud platform offered for research purposes free of charge. It is maintained by the Chameleon commu-

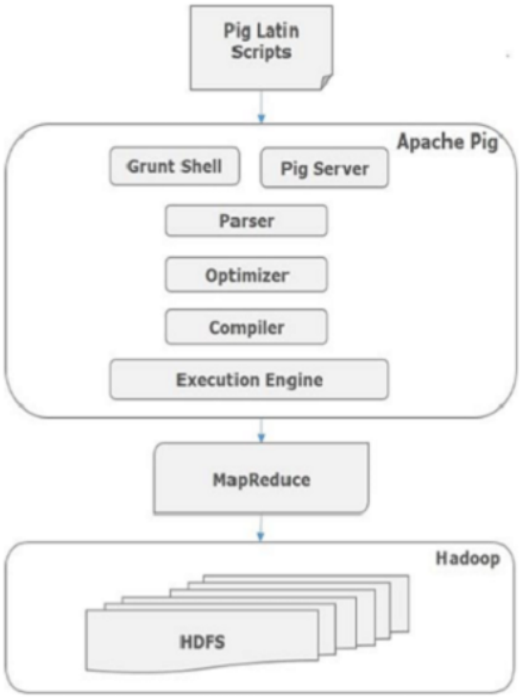

Fig. 3. Architecture of Apache Pig [9]

nity, and is funded exclusively for use by students and faculty by the National Science Foundation. Chameleon Cloud is deployed at University of Chicago and Texas Advanced Computing Center, and consists 650 multi-core cloud nodes, 5PB of total disk space, and leverage 100 Gbps connection between the sites.

The current project uses a cluster created on Chameleon cloud instance with three virtual machines on which Hadoop operates. Ubuntu 14.04 was installed on these machines with 20 GB disk space and 2 GB ram each.

2.7. Data Analysis

Data was analyzed using Pig in the MapReduce mode. The datasets clicks_train and events were joined to get a list of documents to make the predictions. The above list was joined to documents_events, documents_categories and documents_topics to generate a master dataset of all the required documents, their topics, categories and entities. The dataset promoted_content was joined with documents_events, documents_categories and documents_topics to generate a dataset which has all advertisement ids, and their related document entitiy, topic and category. A master join between the two datasets on the event, topic and category ID was performed to get matching document IDs. On this data, relevancy score for each match, by multiplying the confidence levels. The final dataset is grouped by document ID and select top 6 matching ads based on relevancy score. The top 6 ads are the recommendations to be shown on each page. This recommendation engine is purely based on the matching document properties.

3. SETUP AND CONFIGURATION

This project was setup using Ansible and Cloudmesh Client. Cloudmesh client was used to install and deploy Hadoop clusters on Chameleon Cloud. Ansible was then configured to perform a set of tasks on the namenode of the server. File movement was handled using Ansible playbooks. The final Pig script was run on the namenode Hadoop server using automated Ansible scripts.

Server	Ubuntu14.04
VCPU	2
Ram	4 GB
Disk	40 GB

Table 1. Virtual Machine Configuration

The configuration for the cluster deployed is given in Table 1. The project uses three virtual machines defined on Chameleon Cloud. One of them acted as namenode while the rest acted as datanodes.

4. WORK FLOW

The deployment of clusters was done using Cloudmesh client's "cm deploy" command. This automatically created a three node Hadoop cluster and enabled cross SSH between the nodes so that they will be able to communicate with each other. Next step was to use Ansible to transfer the data files and Pig script to cloud. Once this data transfer was done, Ansible scripts were used to move the files from cloud to hdfs dfs repository. The pig script was run using Ansible from local system and timed for benchmarking. The final step was to transfer the files back from hdfs to local system using Ansible.

5. EXPERIMENTS AND RESULTS

Multiple experiments were conducted using different sizes of input data and the time results were used to benchmark Chameleon Client for this particular analysis. The data size was taken for predicting the most likely to be clicked for 100, 2000, 5000, 10000 webpages. Each document ID corresponds to a page that needs predictions and a list 6 Ad_id are predicted for each document_id based on relevance factor calculated. The results are presented in 4, **??**, **??**.

The graphs are plotted for time versus number of webpages to which predictions are made (# documents). It is evident from 4, the time required to predict is linearly proportional to number of webpages. This shows that the time complexity is O(n) where n is number of document ids.

6. CONCLUSION

Apache Hadoop technologies can be used to process large volumes of data parallely to make quick analyses. Apache Pig is an addon for Hadoop, which is a high level language for writing queries to process data. Apache Pig is efficient in query optimization and enables parallelism for its processing. The current project uses Apache Hadoop and Pig to process large datasets related to online ad prediction. Online advertisement is one space where large amounts of data is collected and it demands the use of Hadoop like parallel processing systems to get meaningful insights from the data.

As we can see from the results, Apache Pig was able to handle reasonably large sized files in short times. The processing time increases as the input query size increases. This is a small cluster of three nodes and it is able to process such large volumes of data efficiently. These nodes can be horizontally expanded to

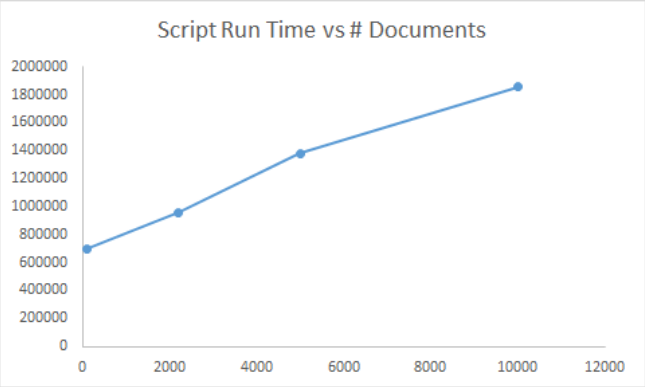

Fig. 4. Benchmarking Results: Time(in Milliseconds) taken for predicting Best 6 Ads for webpages

utilize the full extent of Hadoop's parallel processing capabilities in order to scale up.

ACKNOWLEDGEMENTS

The authors thank Professor Gregor Von Laszewski and all the AIs of big data class for the guidance and technical support.

REFERENCES

[1] D. Mielach, "Americans spend 23 hours per week online, texting," Web-page, accessed mar-13-2017. [Online]. Available: http://www.businessnewsdaily.com/4718-weekly-online-social-media-time.html
[2] marketingterms.com, "Clickthrough rate definition," Web-page, accessed mar-13-2017. [Online]. Available: http://www.marketingterms.com/dictionary/clickthrough_rate/
[3] Wikipedia, "Click-through rate," Webpage, accessed Mar-13-2017. [Online]. Available: https://en.wikipedia.org/wiki/Click-through_rate
[4] Outbrain, "Data introduction," Webpage, accessed Mar-13-2017. [Online]. Available: https://www.kaggle.com/c/outbrain-click-prediction/data
[5] Ansible, "How ansible works," Webpage, accessed Apr-24-2017. [Online]. Available: https://www.ansible.com/how-ansible-works
[6] G. von Laszewski, "Quickstart cloudmesh client," Webpage, accessed Apr-24-2017. [Online]. Available: http://cloudmesh-client.readthedocs.io/en/latest/quickstart.html
[7] Teradata Corporation, "Intro to hdfs and mapreduce," Webpage, accessed Apr-24-2017. [Online]. Available: https://www.thinkbiganalytics.com/2013/07/12/intro-hdfs-mapreduce/
[8] B. Proffitt, "Hadoop: What it is and how it works," Webpage, accessed Apr-24-2017. [Online]. Available: http://readwrite.com/2013/05/23/hadoop-what-it-is-and-how-it-works/
[9] S. P. Bappalige, "An introduction to apache hadoop for big data," Webpage, accessed Apr-24-2017. [Online]. Available: https://opensource.com/life/14/8/intro-apache-hadoop-big-data
[10] Apache Software Foundation, "Welcome to apache pig," Webpage, accessed Apr-24-2017. [Online]. Available: https://pig.apache.org/

AUTHOR BIOGRAPHIES

Sahiti Korrapati is pursuing her MSc in Data Science from Indiana University Bloomington