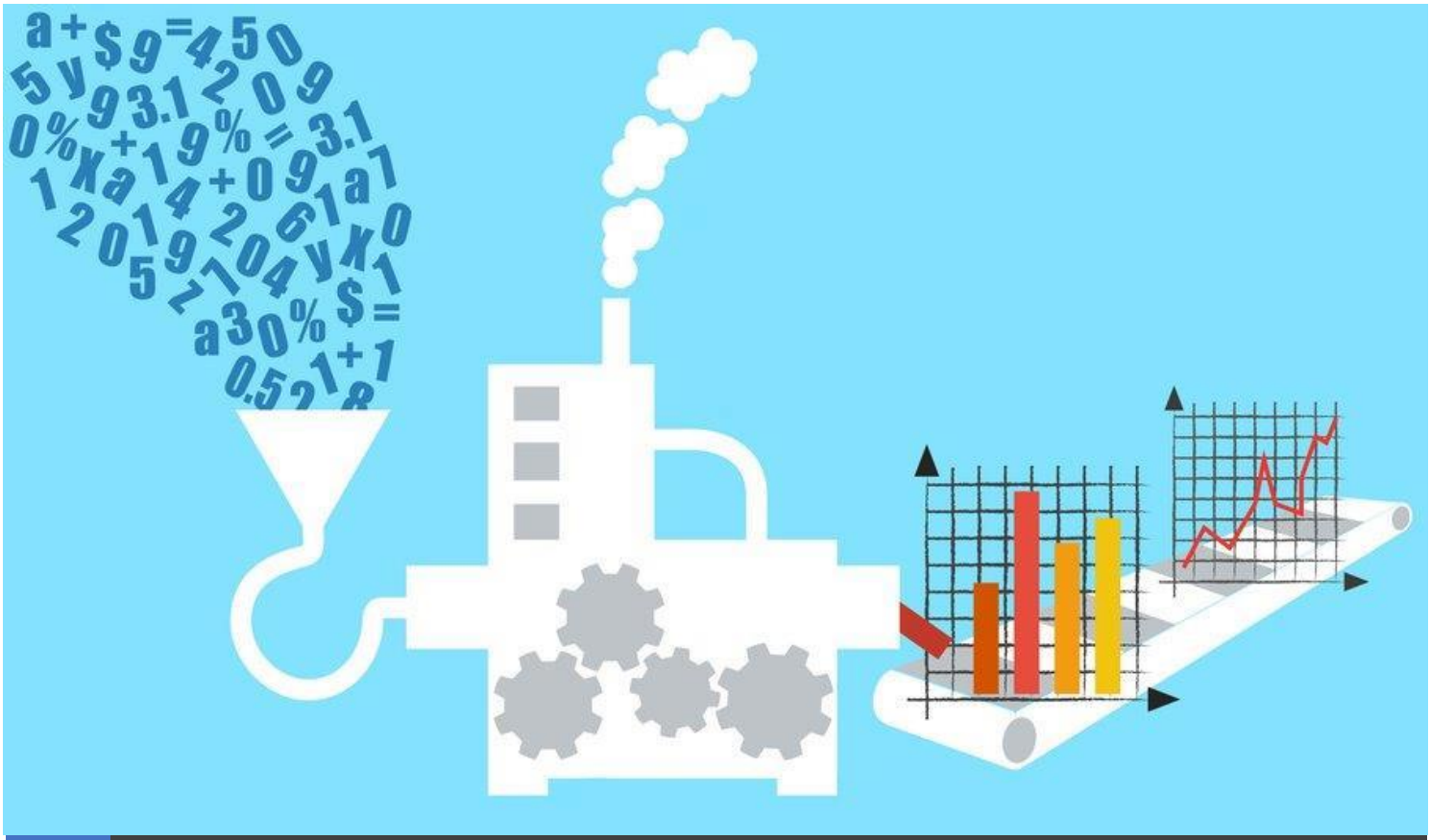


Big Data Ingestion Using Hadoop (trendzzz4u.com)

Faculty Advisor and Sponsor: Dr. Rohit Aggarwal

Team

Aditya Kannan

Ekta Jaiswal

Vikal Gupta

Table of Contents

Executive Summary

The purpose of this project is to build and enable a cloud platform for performing Clickstream Analytics. For this project, we have developed an ecommerce platform (www.trendzzz4u.com) which would provide the data. The industrial use case for this project is to make decisions and build marketing campaigns based on the results of the analysis.

Since the business expects exponential growth in terms of volume of unstructured data, there is a need to build highly scalable cloud platform. We have leveraged the power of Amazon Web Services (EC2) to configure the base infrastructure for our project. Using Cloudera Hadoop on AWS, we can perform all data transformation to track the website visitor data and convert it to structured format. The structured data would act as the primary data source for the marketing teams which can leverage data visualizations tools such as tableau and power bi to gain insights. To summarize this is an end to end implementation project which would add value in long term to any e-commerce company by targeting the right customers and using disruptive marketing strategies.

Project Overview

Objective

The objective of this project is to convert the unstructured data to relational data using Hadoop ecosystem. Our e-commerce platform (trendzzz4u.com) is an online retailer which offers array of products ranging from clothing, accessories and handbags.

The business teams of our company identify the need to track our website visitors so that they can push promotions and discounts at the right time. Through this project the company the plans to fill the void in their marketing campaigns by enabling timely decisions.

Clickstream Analysis

Clickstream Analysis is one of the widely used application of Big Data. It is defined as the process of collecting, analyzing and reporting aggregated data about the website visitor. The path that a visitor takes while browsing the website is called the clickstream.

Clickstreams are records of users' interaction with a website or other compute application. Each row of the clickstream contains a timestamp and an indication of what the user did. Every click or other action is logged—hence the term "clickstream." In some circumstances, what the website does is also logged; this is useful when the website does different things for different users, such as post recommendations.

Clickstream data generally comes from one of two sources: (1) Logs from servers that originally served the website or (2) internet messages transmitted by JavaScript embedded in pages of the website that are received by a central server.

Many of e-commerce vendors have tons of clickstream data that they wish to analyze for brand, operational or other uses. A typical approach is to load the log data into a big data ecosystem such as Hadoop and then perform transformations using ELT (Extract Load and Transform). Because of the sheer scale and complexion of clickstream data, this operation tends to take longer time than usual.

Why this approach?

Clickstreams will tell the precise story about user behavior. With a sample record of 1,000 clickstreams we can find out where people are clicking and where they aren't. This will help any e-commerce company to setup their products and web pages.

For example, the top right corner has the best conversion, or using different text will get you more clicks. Perhaps the company can realize that lots of people are clicking to one specific web page. All this information combined is valuable data that goes beyond normal analytics.

Difference between Cookie based approach and Clickstream?

A cookie is a text file containing information about a user that a web server stores on a computer's hard drive. They're typically relied on for remembering specific user settings that have taken place on the specific device and in the specific browser in question – account log-in information, shopping cart items, location settings, etc. Traditionally e-commerce companies preferred cookie based marketing approach.

Major disadvantage of cookie data is it can be lost due to deletion, expiration, and blocking. Because many consumers are blocking cookies altogether with ad blocking software or private browsing. For instance, 11 percent of the global internet population is blocking ads, and therefore cookies.

To overcome the challenges in this approach companies have shifted to utilize clickstream data. The new approach fills few of the loopholes of the traditional system.

Data Engineering using Hadoop ecosystem

Data engineering is the process of building analytic data infrastructure, or internal data products, that supports the collection, cleansing, storage, and processing (in batch or real time) of data for answering business questions.

Examples can include:

- The construction of data pipelines that aggregate data from multiple sources
- The creation of pre-built tools that assist data scientists in the query process (e.g. UDFs or entire applications)
- Data engineers rely on Apache Hadoop ecosystem components such as Apache Spark, Apache Kafka, and Apache Flume as a foundation for this infrastructure.

For purpose of this project we have targeted learning and implementing Cloudera Distribution of Hadoop. Cloudera is a vendor which offers packaged solutions of Hadoop ecosystem components.

Business Requirements

To support e-commerce analytics, there are many vendors which offer ready to use solutions. Few of them which are available in market are Google Analytics, ELK and Open tracker. We finalized on using Hadoop firstly because we wanted to learn an implementation of end to end project. Also, clickstream analysis is the most popular use case in Hadoop.

Questions asked by the Marketing Department

The Marketing department of our company wants the following business questions to be answered to promote and push discount deals.

- Most popular website browsing time
- Most popular product category
- Weekly Distribution of Clicks per page
- Successful vs Unsuccessful Customer ratio

For each of the answers, they need adequate evidence so that they can tie results to the competitive strategies.

Technology Stack Used in this project

There were three separate components involved in the implementation of the project. Each component consists of a set of technologies used with a purpose.

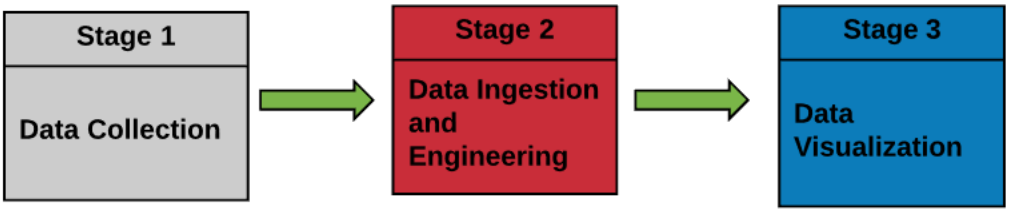
1) e-Commerce Website Setup –
    a. Magento eCommerce store on Go Daddy Server: - This component was used for building the online store.

2) Data Collection and Development Environment –
    a. Divolte.js: - To enable data collection
    b. Aws EC2: - To host and build cloud analytics platform
    c. Cloudera Hadoop: - The development environment for the project

3) Data Visualization –
    a. Microsoft Power BI and Tableau: - To generate reports from the transformed data.

Project Workflow

The workflow used to complete out project is defined below. Each of the stages has few sub-stages associated with it. We followed an agile SCRUM approach throughout the project lifecycle.



The detailed explanation for each of the stages is provided in the Results section of this document. At the end, the project we had completed four iterations of the above workflow.

Performing several iterations helped us to identify and streamline the best practices which are needed in big data projects.

Team Member and Roles

The size of the team is three. Below given is a snapshot of each of the members roles in the project.

Ekta Jaiswal (u1064358) – Data Engineer

- Performed ELT (Extract Load and Transform) using Apache Pig
- Coded Python script to enable data transformation
- Assisted the team with technical knowledge in data warehousing.

Vikal Gupta (u1081193) – Software Developer and Reporting Analyst

- Gained broad understanding of e-commerce platforms such as Magento, Prestashop and Wordpress.
- Responsible for setup of e-commerce store.
- Developed reports and dashboards in Tableau

Aditya Gopala Kannan (u1082183) – Software Engineer and Project Manager

- Responsible for infrastructure setup on Aws EC2.
- Installation of Cloudera Hadoop and Divolte tracker on Aws EC2.
- Coded parser in Apache Hive.
- Getting the team together for monthly meetings, to track deadlines and communication with sponsor.

For any additional long standing technical issue or activity, the team used to get together and resolve it collectively.

Results

Project Stages Explanation

To complete the project, following were the stages our team worked parallelly to achieve the desired results. The project architecture is a combination of all the stages given below.

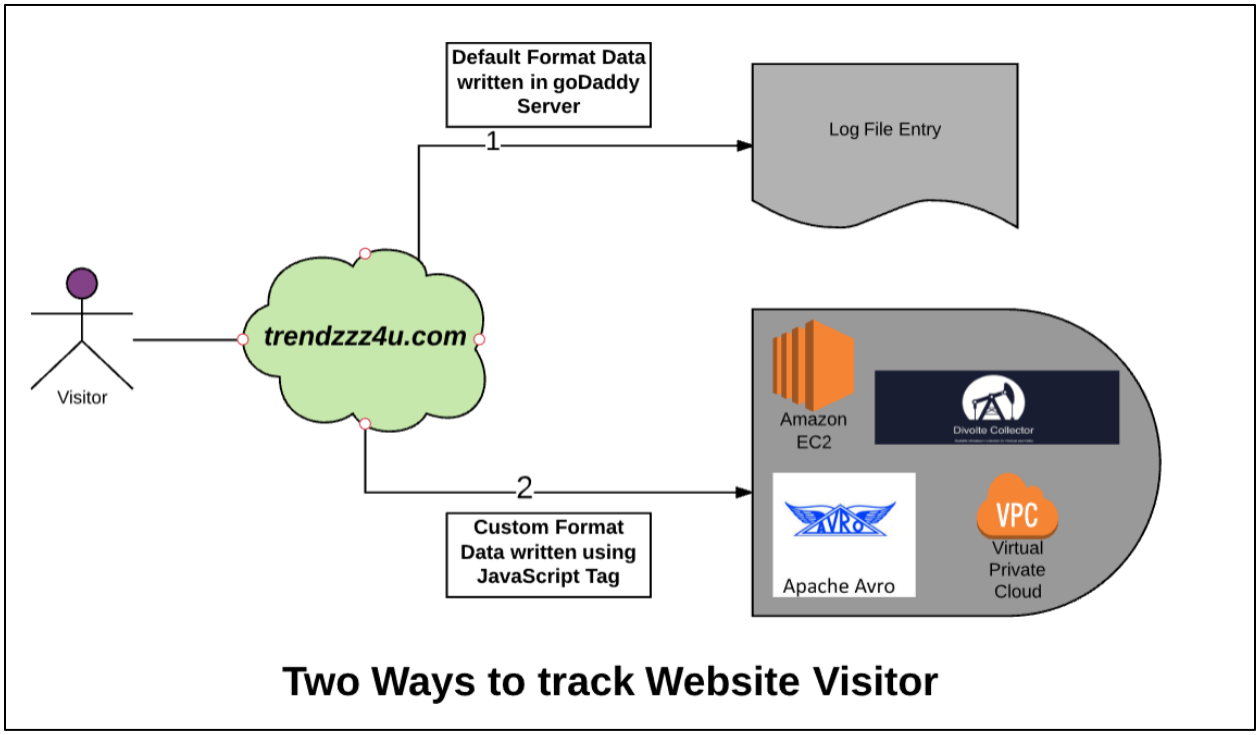
Stage 1 –Data Collection

This was the basic step to kick start the project. It involved setting up the architecture for data collection for the project. Following were the sub categories involved in each of the phase.

When a visitor navigates our website, for each click we tracked the data in two different ways given below

a) Data Collected in Log File
b) Custom Data Collection using Divolte tracker.

The below infographic depicts the above statements visually.



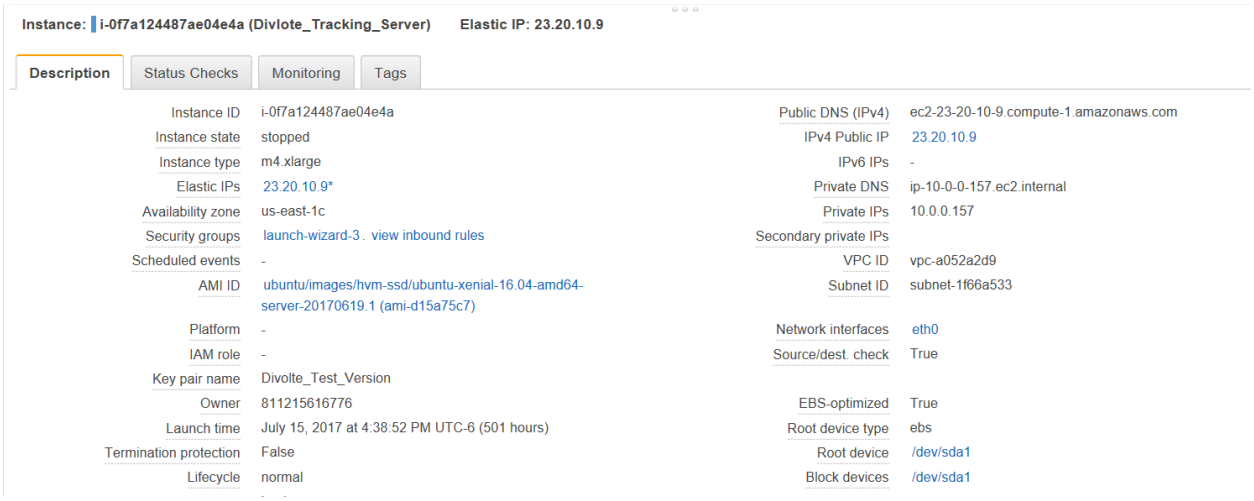
a) Data Collected in Log File: - Whenever any user visits any website there is a corresponding entry created in the log file of the webserver. This file does not have any schema defined for it. Below given is the example of the sample file generated at our go daddy server.

```
157.55.39.231 - - [30/Apr/2017:02:16:07 -0700] "GET / HTTP/1.1" 302 - "-" "Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/
bingbot.htm)"
157.55.39.231 - - [30/Apr/2017:02:16:18 -0700] "GET /?SID=uvgl9to1im3k9t9fie7kh11hn3 HTTP/1.1" 200 8667 "-" "Mozilla/5.0 (compatible;
bingbot/2.0; +http://www.bing.com/bingbot.htm)"
66.249.66.125 - - [30/Apr/2017:04:24:39 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 229 "http://www.trendzzz4u.com/women/
tops-women/jackets-women.html?material=38&size=167&style_general=126" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/
bot.html)"
66.249.66.127 - - [30/Apr/2017:04:24:41 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 230 "http://www.trendzzz4u.com/women/
tops-women/jackets-women.html?material=38&size=167&style_general=126" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/
bot.html)"
66.249.66.127 - - [30/Apr/2017:05:14:06 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 270 "http://www.trendzzz4u.com/men/
tops-men.html?climate=203&color=52" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.66.127 - - [30/Apr/2017:07:29:23 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 228 "http://www.trendzzz4u.com/promotions/
pants-all.html?climate=205&color=58&pattern=195&style_bottom=109" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/
```

b) Avro File generated by Divolte tracker: - Divolte. Js is an open source community project owned by Apache. Using this project, we can track custom events on any web page by adding a JavaScript tag.

Following were the steps involved in activating the Divolte tracker:

- Setting up Ubuntu Linux Server on Amazon Web Services (EC2). This acted as the data collection server.



- Installing Apache Web Server on Linux Server to host websites.
- Installing Divolte.js on the hosted Linux Server on Amazon EC2
- Adding custom js tag on the Magento ecommerce store.

<script src="http://ec2-23-20-10-9.compute-1.amazonaws.com:8290/divolte.js" defer async></srcipt>

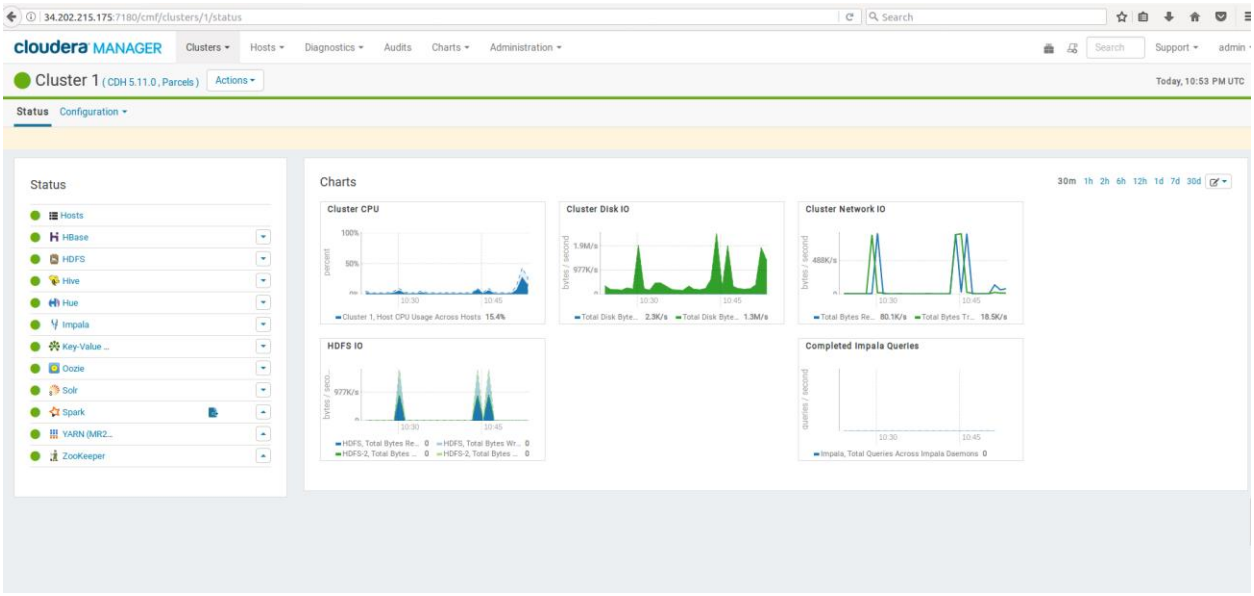
- Collecting data at the AWS Linux server for every click on the e-commerce website.

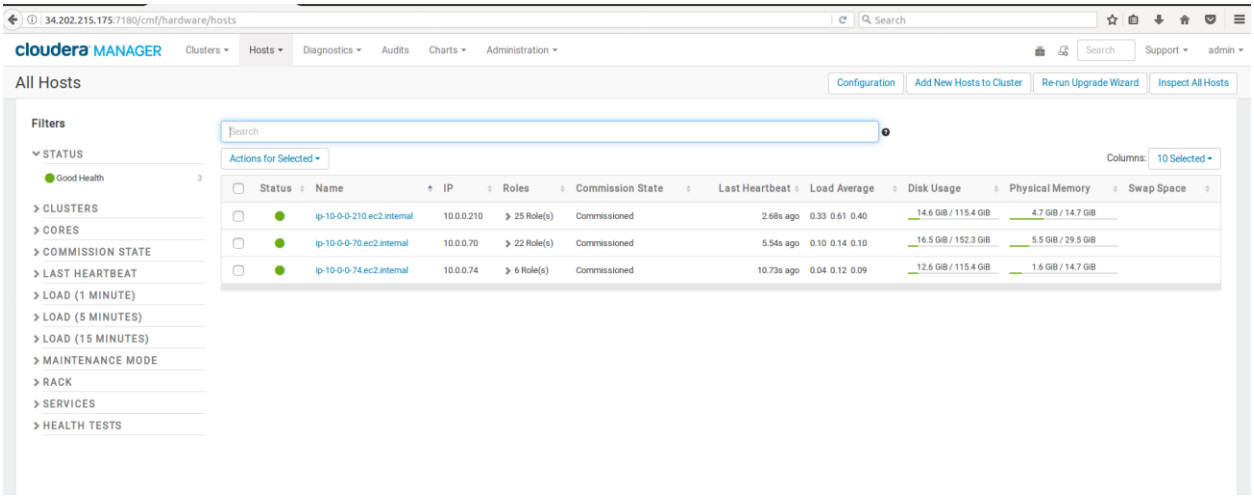
Additionally, to complete the setup it also involved setting up multi node Hadoop Cluster on AWS Ec2. This served as the development environment for the project.

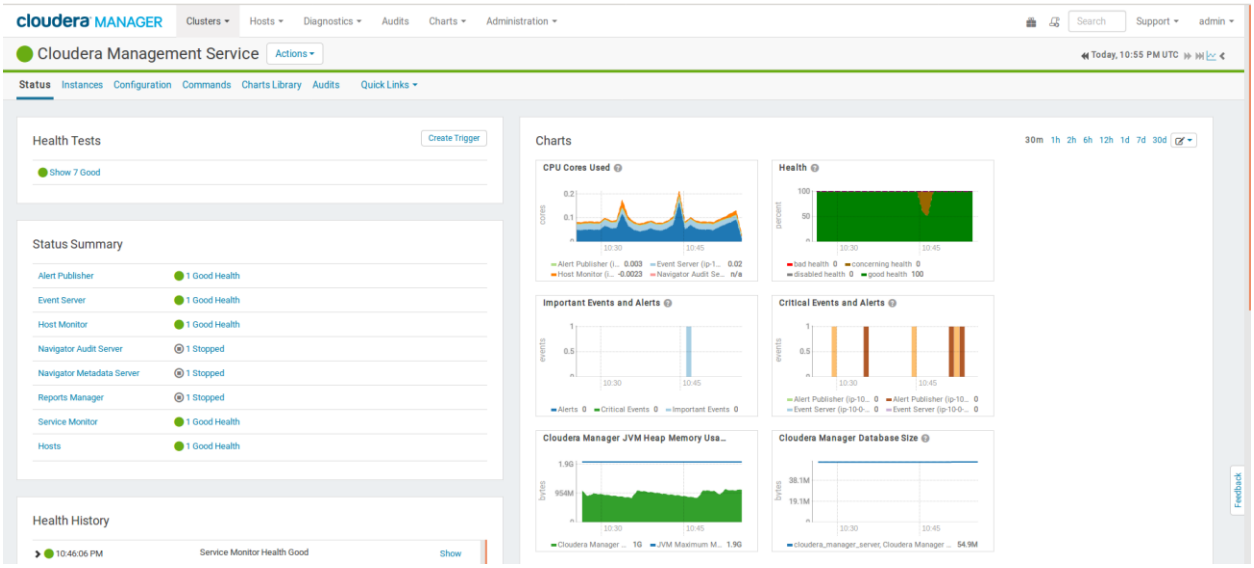
Amazon Web Services Control Panel for Instances:

Name	Instance ID	Instance Type	Availability Zo	Instance Sta	Statu	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
CL_ResourceManager	i-015c7a3d1a2f989bc	m3.2xlarge	us-east-1c	stopped		None	ec2-34-202-215-175.co...	34.202.215.175
CL_NodeManager1	i-09c2bd1126d4be171	m3.xlarge	us-east-1c	stopped		None	ec2-34-202-192-105.co...	34.202.192.105
CL_NodeManager2	i-0c2ead85dade5e25e	m3.xlarge	us-east-1c	stopped		None	ec2-34-194-89-0.comp...	34.194.89.0

Few snapshots of the Cloudera Hadoop environment. Cloudera Manager provides a web UI to access all the nodes in the cluster. This drastically reduces the time consumption to monitor and track the individual components of Hadoop ecosystem.







Technical explanation given by Cloudera:

Cloudera Manager runs a central server ("the Cloudera Manager Server," which has also been called the "SCM Server" and the "CMF Server" in the past) which hosts the UI Web Server and the application logic for managing CDH. Everything related to installing CDH, configuring services, and starting and stopping services is managed by the Cloudera Manager Server.

The Cloudera Manager Agents are installed on every managed host. They are responsible for starting and stopping Linux processes, unpacking configurations, triggering various installation paths, and monitoring the host.

Stage 2 – Data Ingestion and Engineering

Pig Parser

The chief input source of data for our project are the logs generated by the Apache web server of this website- “trendzz4u.com”.

The Apache access logs stores information about events that occurred on your Apache web server. For instance, when someone visits our website, a log is recorded and stored to provide the web server administrator with information such as the IP address of the visitor, what pages they were viewing, status codes, browser used, etc.

Apache web servers also provide administrators with another type of log file called error logs. However, for our project, we specifically focused on the Apache access log file.

Every time a user visits the website, the server sends them the page and writes down some basic information about the person asking for it. That's one line for every single request/clicks from the user, human or bot that accesses your website.

This information generally includes what pages’ people are viewing, the success status of requests, and how long the request took to respond. Given below is a fragment of the server logs of the website- - “trendzz4u.com”.

```
157.55.39.231 - - [30/Apr/2017:02:16:07 -0700] "GET / HTTP/1.1" 302 - "-" "Mozilla/5.0 (compatible; bingbot/2.0;
+http://www.bing.com/bingbot.htm)"
157.55.39.231 - - [30/Apr/2017:02:16:18 -0700] "GET /?SID=uvgl9tolim3k9t9fie7khllhn3 HTTP/1.1" 200 8667 "-"
"Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)"
66.249.66.125 - - [30/Apr/2017:04:24:39 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 229
"http://www.trendzzz4u.com/women/tops-women/jackets-women.html?material=38&size=167&style_general=126" "Mozilla/5.0
(compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.66.127 - - [30/Apr/2017:04:24:41 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 230
"http://www.trendzzz4u.com/women/tops-women/jackets-women.html?material=38&size=167&style_general=126" "Mozilla/5.0
(compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.66.127 - - [30/Apr/2017:05:14:06 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 270
"http://www.trendzzz4u.com/men/tops-men.html?climate=203&color=52" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
66.249.66.127 - - [30/Apr/2017:07:29:23 -0700] "POST /swatches/ajax/media/ HTTP/1.1" 200 228
"http://www.trendzzz4u.com/promotions/pants-all.html?climate=205&color=58&pattern=195&style_bottom=109"
"Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"
66.249.66.127 - - [30/Apr/2017:08:39:20 -0700] "GET /robots.txt HTTP/1.1" 404 7550 "-" "Mozilla/5.0 (compatible;
```

Understanding the log structure:

Each line in the log file represents a single "hit" on a file on the web server, and is made up of a number of fields.

- IP Address: This is the IP of the customer or API visiting the website. Some web servers are set to automatically resolve IP addresses by conducting a who is lookup.
- Timestamp & Time Zone: This is the time the request was made (along with the time zone).
- Request Type: A GET request (which is what you see most of the time) is a person or bot asking for a page, a PUT request is someone sending information (e.g. sending a form).
- Page: The page being requested by the user.
- The protocol: This will always show HTTP, if you have a mixed HTTP/HTTPS website you'll need to get your developer to set-up tracking for this.
- Status Code: A number indicating the server's response type, e.g. 200 - request returned- OK, 301 - request redirected etc.
- Page size in bytes: The number of bytes transferred.
- User agent: The name of the API, or the name of the browser version a user is connected to access the website.

Tools that can be used to process logs:

- **Processing Using Apache Pig:** Pig has several built-in libraries that can help us load the apache web log files into pig and some cleanup operation on string values from crude log files. All the functionalities are available in the piggybank.jar mostly available under *pig/contrib/piggybank/java/* directory.

Steps to parse the log file are given below:

- Load Apache log files into HDFS location.
- Register piggybank jar file.
- Include pig commands to parse every record in the weblog and dump the result in another temporary file called 'Parsed2'.

Pig Script: {
A = LOAD '/home/training/Downloads/WebLog2' USING PigStorage ('-')
as(ip_addr:chararray,temp1:chararray,timestamp:chararray,time_zone:chararray,req_type:chararray,req_link:chararray,req_det:chararray,resp_code:int,bytes:int);
B = FOREACH A GENERATE
ip_addr,temp1,timestamp,time_zone,req_type,req_link,req_det,resp_code,bytes;
data = distinct B;
dump data;
STORE data INTO '/home/training/Downloads/Parsed2' using PigStorage('*');
}

Sample Log processed looks like below-


ParsedLog.xls

- This file is further analyzed to answer below questions based on parsed records.

1) **/* Calculate the number of web pages a user visited based upon the IP Address*/**
   ip_data = GROUP data by ip_addr;
   ip_count = FOREACH ip_data GENERATE group AS timestamp,
   COUNT(data) as total_visits;
   dump ip_count;

2) **/* Statistics where requests were successful i.e. where response code =200 */**
   time_data = GROUP data BY timestamp;
   byte_count = FOREACH time_data GENERATE group AS timestamp,
   SUM(data.bytes) as total_bytes;
   dump byte_count;

Python Parser

We also wanted to try the processing of logs using python to parse the records further and break the request type field into more fields so that it will help us to perform detailed analysis about categories and sub categories of product links. For this purpose, we have developed a script- “parser.py”.

parser.py

With the help of this script we further broke the information in link fields to get more detailed information about individual categories and sub categories of product information that will help us to do visualization in Tableau.

Sample Log post processing looks like below-

ParsedLog2.xls

Hive Parser

To parse the. avro files we used hive SerDe to convert data in relational format. After the data transformation in, we exported the data to Power BI to generate report on the Customer Conversion Rate.

Hive_Parser.txt

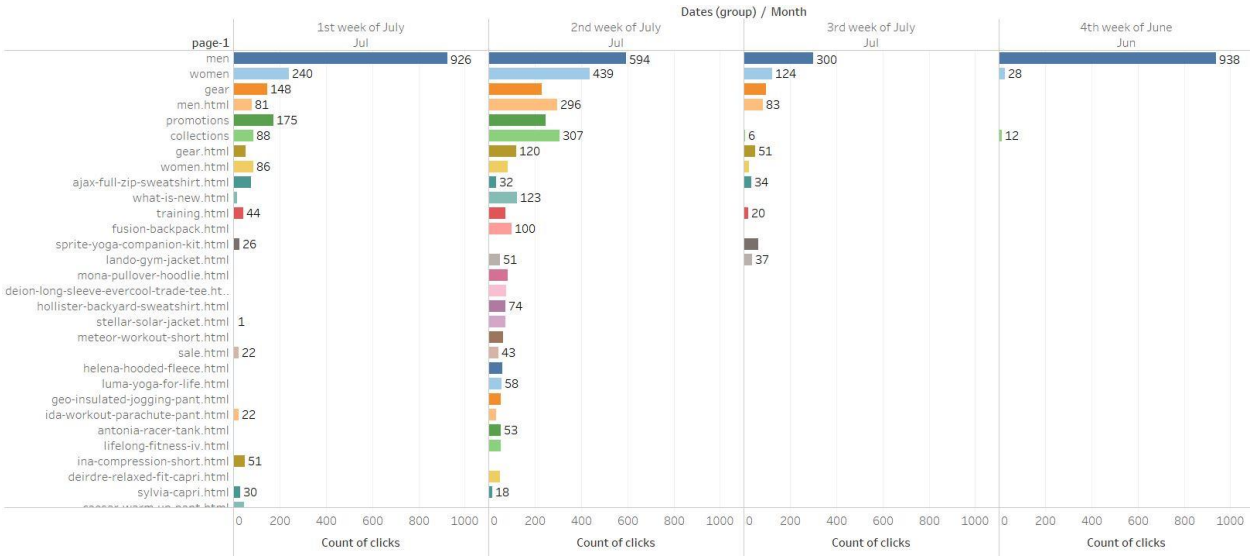
Stage 3 – Data Visualization

We built visualizations from the structured data using tools such as Tableau and Power BI. The main purpose of using multiple business intelligence tools was to develop reports using different custom visuals.

In addition to both tools we also explored d3.js which provides a way to develop custom visualizations using JavaScript framework.

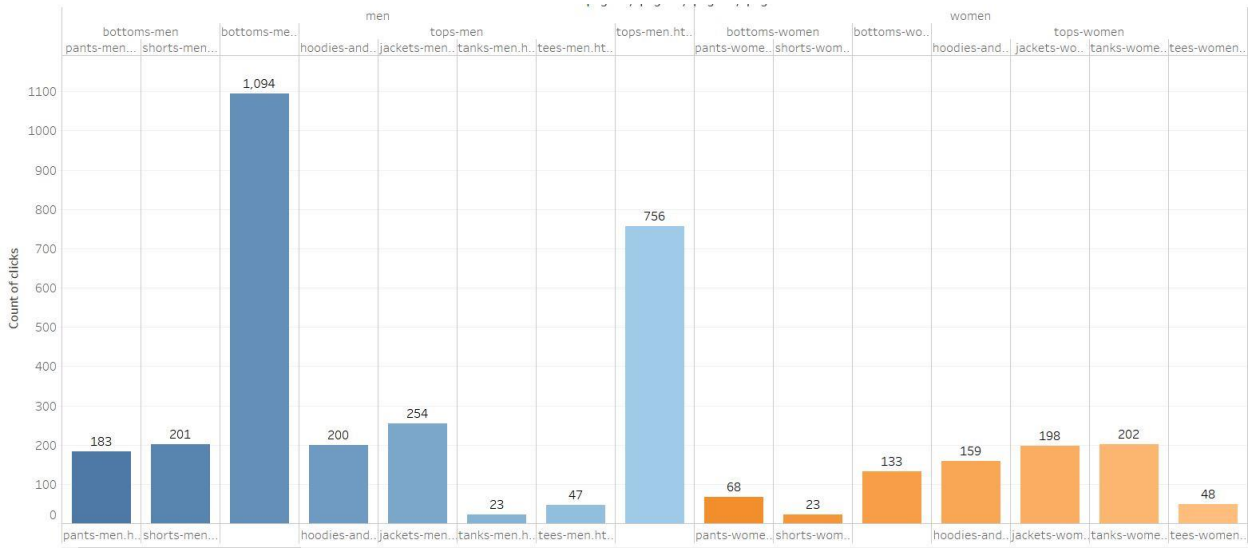
Report 1 – Weekly Distribution of Clicks Per Page

The below report provides a snapshot of the comparison of clicks per page on a weekly basis. The insights what we can derive from this report is that mens.html was clicked the highest number of times in first week of July. Also, there is a gradual decrease in the clicks.



With such granularity, the business team can identify if there is a need to reorder/realign the products on the page so that the number of clicks is on a higher side end of each week.

Report 2 – Most Popular Product Category



The above visualization is a drill down report of the parent categories (Men's, Women's and Accessories) of our website. It shows the number of clicks on the sub category level. Using such information, the marketing department can place the ads in this section, as there would be more chances for the user to click on the advertisement.

Also, the least performing subcategory is tanks-men.html page. The business can rethink a strategy (Product Pricing/Product Combination) to make it more appealing.

Future Scope

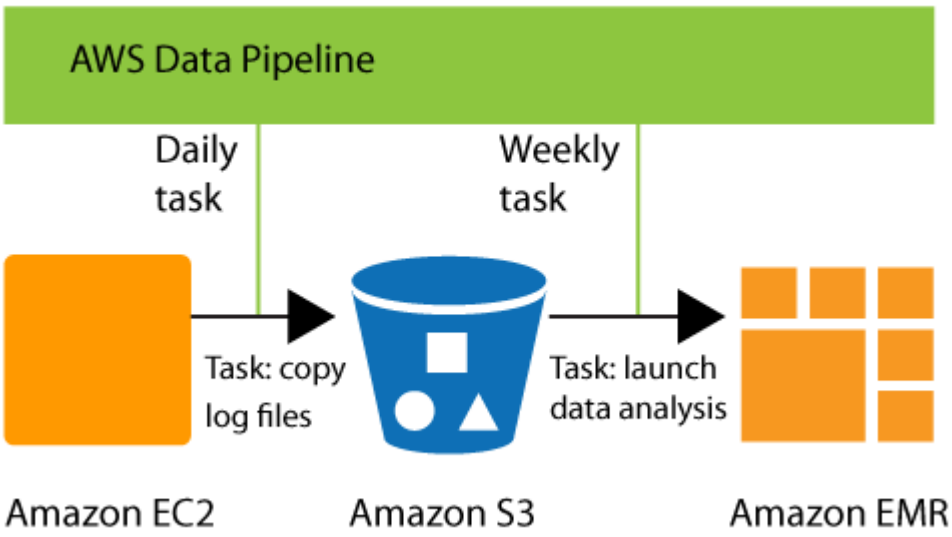
Data Pipeline

As per amazon, AWS Data Pipeline manages and streamlines data-driven workflows, which includes scheduling data movement and processing. The service is useful for customers who want to move data along a defined pipeline of sources, destinations and data-processing activities.

An activity is an action that AWS Data Pipeline performs, such as a SQL query or command-line script. A developer can associate an optional precondition to a data source or activity, which ensures that it meets specified conditions before running an activity. AWS Data Pipeline includes several standard activities and preconditions for services like Amazon DynamoDB and Amazon Simple Storage Service (S3).

As a part of future scope of this project we plan to incorporate this data pipeline between the data source and analysis server on amazon ec2.

For example, we can use AWS Data Pipeline to archive our web server's logs to Amazon Simple Storage Service (Amazon S3) each day and then run a weekly Amazon EMR (Amazon EMR) cluster over those logs to generate traffic reports.



Lessons Learned

Challenges Faced

Throughout the execution of the project we faced several challenges right from onset. Below mentioned are the major challenges we faced throughout the project.

- Setting up e-commerce store on Go Daddy
- Installation of Multi node cluster on AWS EC2
- Installing Divolte tracker on AWS EC2 for Data Collection
- Parsing unstructured data using Apache Pig, Python and Hive
- Learnings of Security Groups, Elastic IP Addresses on AWS EC2

To overcome these challenges, we used to schedule timely meetings with our sponsor Dr. Rohit, who would address all our problems.

Additionally, we used the following references which helped us greatly to complete the project.

References

http://community.cloudera.com/

https://stackoverflow.com/

http://docs.aws.amazon.com/

https://www.thinkbiganalytics.com/2013/10/28/applying-big-data-analytics-to-clickstream-data/

http://mashable.com/2009/08/25/clickstreams/#I6NgYA3R5gq3

https://www.behave.org/uncategorized/whats-difference-cookie-based-people-based-targeting/

http://hadooptutorial.info/cloudera-manager-installation-on-amazon-ec2/

https://www.youtube.com/watch?v=ePQB5hG9Y_4