IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 3, Issue 4, Aug-Sept, 2015
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Clickstream Analysis Using Big Data – Hadoop

Mrs. Charushila Shailendra Patil[1], Mr. Dinesh G Patil[2]

[1]Computer Science, RSSP's Maharashtra College of science and commerce, Pune, Maharashtra, India.
[2]Lead Consultant, CapGemini, Pune, Maharashtra, India

Abstract:As web is becoming a main channel for reaching customers and prospects, Clickstream data generated by websites has become another important enterprise data source, like other traditional business data sources, like store transactions, CRM data, call centre's logs etc. As simple as it sounds for recording every click a customer made, Clickstream data actually offers a wide range of opportunities for modelling user behaviour, gaining valuable customer insights.

Clickstream analysis commonly refers to analysing click data. Such analysis is typically done to extract insights into visitor behaviour on website especially social media or ecommerce websites to identify potential customers or to identify recommendations for existing customers. It results into making powerful decisions subsequently. Clickstream analysis can be used to figure out how users interact with online presence, what geographies and time zones is most of traffic coming from, what devices and Operating Systems are most commonly used to access site, what are the common paths users take before they do something that's of interest to you - usually referred to as a "goal" (e.g. register for the mailing list, sign up for an account or add an item to the shopping cart), and so on. You can relate such click data with marketing spend to do further analysis like return on investment of various marketing channels (organic search, paid search, social media spending, etc.). You can also, optionally, join click data with ODS (Operational Data Store) or CRM (Customer Relationship Management) data to do further analysis based on additional information about users.

This is definitely a data source which has been underutilized. However, benefits also come with a problem. Amazon records 5 Billion clicks a day and the whole US generates 400 Billion clicks, equivalent to 3.4 Petabytes a day. This immense volume has given enterprises and their IT professionals a big data problem before they can fully utilize this insight-rich data source.

To solve this big data problem, one of Big data technology like Hadoop can be used. **Hadoop** is an open-source software framework for storing and processing big data in a distributed fashion on large clusters of commodity hardware. Essentially, it accomplishes two tasks: massive data storage and faster processing.

Keywords—clickstream anylsis,pig,Hive,Hadoop,Big data.

I. INTRODUCTION

Clickstream analysis is an approach to examining the way users interact with web pages and software programs that relies on tracking the clicks the user makes. As users navigate, they use their mice to jump from page to page, to activate menus, and to engage in other activities. This information, the clickstream, can be used in a number of different ways to improve experiences for users, develop better marketing campaigns, and analyze productivity in a workplace or similar environment. The clickstream is important in web design, as well as Internet marketing and advertising. It is the focus of clickstream analysis and clickstream mining, both of take advantage of captured clickstreams to gain a fuller understanding of visitors' behavior. Clickstreams reveal both where users are clicking and where they aren't, and clickstream data can be combined with other types of analytics. The data from a clickstream analysis be used to reorganize site page layout, reconfigure an entire website, or sell advertising space based on a history of site and page performance.

"Big Data" as a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications.
In simple terms, ***"Big Data"*** consists of very large volumes of heterogeneous data that is being generated, often, at high speeds. These data sets cannot be managed and processed using traditional data management tools and applications at hand. Big Data requires the use of a new set of tools, applications and frameworks to process and manage the data.
Evolution of Data / Big Data
Data has always been around and there has always been a need for storage, processing, and management of data, since the beginning of human civilization and human societies. However, the amount and type of data captured, stored, processed, and managed depended then and even now on various factors including the necessity felt by humans, available tools/technologies for storage, processing, management, effort/cost, and ability to gain insights into the data, make decisions, and so on
Volume

Volume refers to the size of data that we are working with. With the advancement of technology and with the invention of social media, the amount of data is growing very rapidly. This data is spread across different places, in different formats, in large volumes ranging from Gigabytes to Terabytes, Petabytes, and even more. Today, the data is not only generated by humans, but large amounts of data is being generated by machines and it surpasses human generated data. This size aspect of data is referred to as ***Volume*** in the Big Data world.

Velocity

Velocity refers to the speed at which the data is being generated. Different applications have different latency requirements and in today's competitive world, decision makers want the necessary data/information in the least amount of time as possible. Generally, in near real time or real time in certain scenarios. In different fields and different areas of technology, we see data getting generated at different speeds. A few examples include trading/stock exchange data, tweets on Twitter, status updates/likes/shares on Facebook, and many others. This speed aspect of data generation is referred to as ***Velocity*** in the Big Data world.

Variety

Variety refers to the different formats in which the data is being generated/stored. Different applications generate/store the data in different formats. In today's world, there are large volumes of unstructured data being generated apart from the structured data getting generated in enterprises. Until the advancements in Big Data technologies, the industry didn't have any powerful and reliable tools/technologies which can work with such voluminous unstructured data that we see today. In today's world, organizations not only need to rely on the structured data from enterprise databases/warehouses, they are also forced to consume lots of data that is being generated both inside and outside of the enterprise like clickstream data, social media, etc. to stay competitive. Apart from the traditional flat files, spreadsheets, relational databases etc., we have a lot of unstructured data stored in the form of images, audio files, video files, web logs, sensor data, and many others. This aspect of varied data formats is referred to as ***ariety*** in the Big Data world.

Hadoop:

Hadoop is an **open source framework** for processing, storing and analysing **massive amounts of distributed, unstructured data**. It was designed to **handle petabytes and Exabyte's of data** distributed over **multiple nodes in parallel**.

Rather than dealing with one, huge block of data using a single machine, Hadoop**breaks up big data** into multiple parts so each part can be **processed and analysed at the same time**

Hadoop is big and getting bigger. One recent estimate says that Hadoop currently had a market worth $1.5 billion dollars in 2012, and is projecting a compound annual growth rate of 54.7% between 2012 and 2018 for a global market size of $20.9 billion.

WHY DO WE NEED HADOOP?

Performing large scale computation is difficult –

- Hundreds of gigabytes of data constitute low end of Hadoop-scale
- Hadoop can handle hundreds of gigabytes to terabytes or petabytes
- Such data will not even fit on a single computer's hard drive
- Distributed file system and parallel processing to the rescue

Hadoop changes the economics and the dynamics of large scale computing

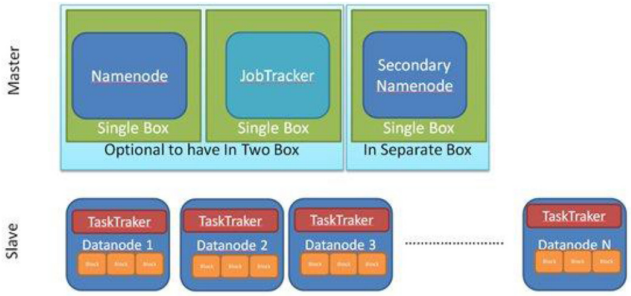
- **Scalable**: New machines can be added based on need
- **Cost effective**: Easily implementable on commodity servers
- **Flexible**: Can handle any type of data, structured and unstructured
- **Fault tolerant**: Reallocates new machines in case of hardware errors

HOW HADOOP WORKS?

- A client handles unstructured and semi-structured data from sources including log files, social media feeds and internal data stores
- HDFS breaks the data up into "parts," which are then loaded into multiple nodes running on commodity hardware
- Once the data is loaded into the cluster, it is ready to be analyzed via the MapReduce framework
- After the MapReduce job is completed, the client accesses these results, which can then be loaded into one of number of analytic environments for analysis

Hadoop Architecture

Master/slave architecture with 5 types of « deamons »: NameNode (NN), DataNode (DN), Secondary NameNode (SNN), JobTracker (JT) and TaskTracker (TT)



Hadoop Distributed File System (HDFS) component

HDFS is one primary components of Hadoop cluster and HDFS is designed to contain Master-Slave architecture

- **Master (NN and SNN)**

Gives orders to DN to perform I/O of data at lowest level
Monitors in real-time the location of « data bits »
SNN assists the NN in monitoring state of the HDFS cluster
SNN takes regular « snapshots » of system configuration

- **Slaves (DN)**

Reads/writes HDFS blocks in local HDD of slave machine
Executes orders from NN – Communicate with NN and other DN's

MapReduce component

MapReduce is also primary component of Hadoop and it also contains Master-Slave architecture

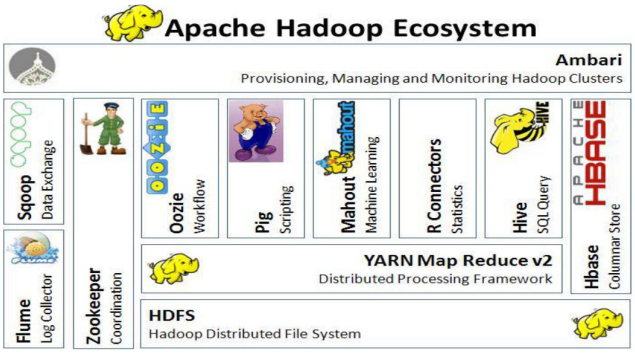
- **Master (JT)**

Interface with client's applications
Generates and monitors execution plans

- **Slaves (TT)**

Responsible for executing individual tasks ordered by the JobTracker
One single TT per node
Communicates with JT and other TTs

HADOOP ECOSYSTEM



Various Components

Ambari	Deployment, configuration and monitoring
Flume	Collection and import of log and event data
HBase	Column-oriented database scaling to billions of rows
HCatalog	Schema and data type sharing over Pig, Hive and MapReduce
HDFS	Distributed redundant file system for Hadoop
Hive	Data warehouse with SQL-like access
Mahout	Library of machine learning and data mining algorithms
MapReduce	Parallel computation on server clusters
Pig	High-level programming language for Hadoop computations
Oozie	Orchestration and workflow management
Sqoop	Imports data from relational databases
Whirr	Cloud-agnostic deployment of clusters
Zookeeper	Configuration management and coordination

Improving programmability

Pig and Hive are two solutions that makes Hadoop programming easier to non-Java programmers

- **Pig**

A programming language that simplifies some of the common tasks
Pig's built-in operations are simpler than MapReduce functions

- **Hive**

Enables Hadoop to operate as a data warehouse
SQL-like queries for exploring large amounts of more structured data

Improving data access

To provide better access of data and to connect with external databases; HBase, Sqoop, and Flume components runs on top of HDFS layer

- **HBase**

A column-oriented database to host billions of rows of data for rapid access
MapReduce, Hive and Pig can be used on HBase

- **Sqoop**

Imports data from relational databases into HDFS

- **Flume**

Import streaming flows of log data directly into HDFS

Coordination and Workflow

Zookeeper and Oozie provides coordination and naming services across Hadoop cluster

- **Zookeeper**

Handles member synchronization of the cluster
Know where to access services, and how they should be configured

- **Oozie**

Helps in handling complex pipelines of data transformations
Provides features to manage the workflow and dependencies

Management and Deployment

IBM and Microsoft have contributed to the development of Hadoop components related to monitoring and administration

- **Ambari**

Helps system administrators to deploy and configure Hadoop
Further has features related to Cluster upgradation and Monitoring services

- **Whirr**

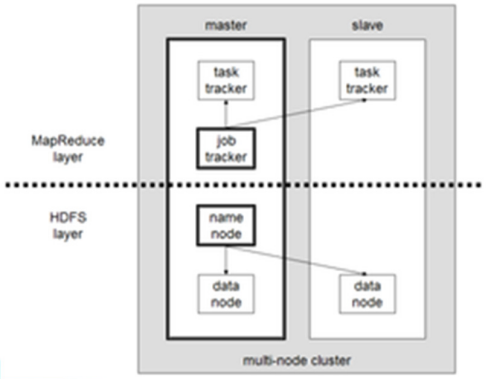
Highly complimentary component
Offers services for running Hadoop on cloud platforms
Currently supports the Amazon EC2 services

Apache Hadoop is a set of algorithms (an open-sourcesoftware framework) for distributed storage and distributed processing of very large data sets (Big Data) on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines, or racks of machines) are commonplace and thus should be automatically handled in software by the framework.

The core of Apache Hadoop consists of a storage part (Hadoop Distributed File System (HDFS)) and a processing part (MapReduce). Hadoop splits files into large blocks (default 64MB or 128MB) and distributes the blocks amongst the nodes in the cluster. To process the data, Hadoop Map/Reduce transfers code (specifically Jar files) to nodes that have the required data, which the nodes then process in parallel. This approach takes advantage of data locality[3] to allow the data to be processed faster and more efficiently via distributed processing than by using a more conventional supercomputer architecture that relies on a parallel file system where computation and data are connected via high-speed networking.

Hadoop consists of the *Hadoop Common* package, which provides filesystem and OS level abstractions, a MapReduce engine (either MapReduce/MR1 or YARN/MR2) and the Hadoop Distributed File System (HDFS). The Hadoop Common package contains the necessary Java ARchive (JAR) files and scripts needed to start Hadoop. The package also provides source code, documentation, and a contribution section that includes projects from the Hadoop Community.

For effective scheduling of work, every Hadoop-compatible file system should provide location awareness: the name of the rack (more precisely, of the network switch) where a worker node is. Hadoop applications can use this information to run work on the node where the data is, and, failing that, on the same rack/switch, reducing backbone traffic. HDFS uses this method when replicating data to try to keep different copies of the data on different racks. The goal is to reduce the impact of a rack power outage or switch failure, so that even if these events occur, the data may still be readable.



A multi-node Hadoop cluster

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or *worker node* acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications.Hadoop requires Java Runtime Environment (JRE) 1.6 or higher. The standard startup and shutdown scripts require that Secure Shell (ssh) be set up between nodes in the cluster.

In a larger cluster, the HDFS is managed through a dedicated NameNode server to host the file system index, and a secondary NameNode that can generate snapshots of the namenode's memory structures, thus preventing file-system corruption and reducing loss of data. Similarly, a standalone JobTracker server can manage job scheduling. In clusters where the HadoopMapReduce engine is deployed against an alternate file system, the NameNode, secondary NameNode, and DataNode architecture of HDFS are replaced by the file-system-specific equivalents.

References:

http://bigdatauniversity.com/

- "Data, data everywhere". *The Economist*. 25 February 2010. Retrieved 9 December 2012.
- Laney, Douglas. "3D Data Management: Controlling Data Volume, Velocity and Variety". Gartner. Retrieved 6 February 2001.
- Beyer, Mark. "Gartner Says Solving 'Big Data' Challenge Involves More Than Just Managing Volumes of Data". Gartner. Archived from the original on 10 July 2011. Retrieved 13 July 2011.
- Laney, Douglas. "The Importance of 'Big Data': A Definition". Gartner. Retrieved 21 June 2012.
- "What is Big Data?". Villanova University.
- http://www.bigdataparis.com/presentation/mercredi/PDelort.pdf?PHPSESSID=tv7k70pcr3egpi2r6fi3qbjtj6#page=4

- Billings S.A. "Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains". Wiley, 2013
- Delort P., Big data Paris 2013 http://www.andsi.fr/tag/dsi-big-data/
- Delort P., Big Data car Low-Density Data ? La faible densité en information comme facteur discriminant http://lecercle.lesechos.fr/entrepreneur/tendances-innovation/221169222/big-data-low-density-data-faible-densite-information-com
- Lee, Jay; Bagheri, Behrad; Kao, Hung-An (2014). "Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics". *IEEE Int. Conference on Industrial Informatics (INDIN) 2014*.
- Lee, Jay; Lapira, Edzel; Bagheri, Behrad; Kao, Hung-an. "Recent advances and trends in predictive manufacturing systems in big data environment". *Manufacturing Letters* **1** (1): 38–41. doi:10.1016/j.mfglet.2013.09.005.
- "LexisNexis To Buy Seisint For $775 Million". Washington Post. Retrieved 15 July 2004.
- "LexisNexis Parent Set to Buy ChoicePoint". Washington Post. Retrieved 22 February 2008.
- "Quantcast Opens Exabyte-Ready File System". www.datanami.com. Retrieved 1 October 2012.
- Bertolucci, Jeff "Hadoop: From Experiment To Leading Big Data Platform", "Information Week", 2013. Retrieved on 14 November 2013.
- Webster, John. "MapReduce: Simplified Data Processing on Large Clusters", "Search Storage", 2004. Retrieved on 25 March 2013.
- "Big Data Solution Offering". MIKE2.0. Retrieved 8 Dec 2013.
- "Big Data Definition". MIKE2.0. Retrieved 9 March 2013.
- Boja, C; Pocovnicu, A; Bătăgan, L. (2012). "Distributed Parallel Architecture for Big Data". *Informatica Economica* **16** (2): 116–127.
- http://www.imscenter.net or see references 26-28
- Manyika, James; Chui, Michael; Bughin, Jaques; Brown, Brad; Dobbs, Richard; Roxburgh, Charles; Byers, Angela Hung (May 2011). *Big Data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute.
- "Future Directions in Tensor-Based Computation and Modeling". May 2009.
- Lu, Haiping; Plataniotis, K.N.; Venetsanopoulos, A.N. (2011). "A Survey of Multilinear Subspace Learning for Tensor Data". *Pattern Recognition* **44** (7): 1540–1551. doi:10.1016/j.patcog.2011.01.004.
- Monash, Curt (30 April 2009). "eBay's two enormous data warehouses".
Monash, Curt (6 October 2010). "eBay followup — Greenplum out, Teradata > 10 petabytes, Hadoop has some value, and more".
- "Resources on how Topological Data Analysis is used to analyze big data". Ayasdi.
- CNET News (1 April 2011). "Storage area networks need not apply".
- "How New Analytic Systems will Impact Storage". September 2011.
- "What Is the Content of the World's Technologically Mediated Information and Communication Capacity: How Much Text, Image, Audio, and Video?", Martin Hilbert (2014), The Information Society; free access to the article through this link: martinhilbert.net/WhatsTheContent_Hilbert.pdf