Extraction of User Profile from Library Data Set Using HADOOP

Prof.Bilkis Chandargi, Kunal Oswal, Saloni Mapara, Asmita Deshmukh, Richa Runwal,

***Abstract*:-We present the details of a large scale user profiling framework that we developed here on Apache Hadoop. We address the problem of extracting and maintaining a very large number of user profiles extracted from large scale data. In this work, a user profiles is often used to classify a given user into pre-defined user segments or to capture the online behavior of the user including the user's private interests and preferences. A user profiles can be explicitly defined by the user himself. User Profiling is usually defined as the process of implicitly learning a user profiles from data associated with the user. The Data extracted in stored form of the xlsx, pdf, docx format in certain Data-marts or organization is also extracted to get user information and personalize the user's behavior accordingly. Data sources for user profiling include among others the user's browsing sessions or even other user profiles using collaborative filtering techniques.**

***Index Terms*— Extraction, HADOOP, Software engineering, Framework, HDFS.**

I. INTRODUCTION

Hadoop is a rapidly evolving ecosystem of components for implementing the Google Map Reduce algorithms in a scalable fashion on commodity hardware. Hadoop enables users to store and process large volumes of data and analyze it in ways not previously possible with less scalable solutions or standard SQL-based approaches. As an evolving technology solution, Hadoop design considerations are new to most users and not common knowledge.

Hadoop is a highly scalable compute and storage platform. While most users will not initially deploy servers numbered as hundreds or thousands.

1.2. Brief Description:

There is large number of data saved regularly and searched. Such data is difficult to store and calculate. Hadoop is one such software that helps to solve this problem. In big companies there are many employees. It is essential to track all the information about these employees so in this project we create a software using hadoop that extracts all the records off the employee and cluster them and store it. Large data can be stored using hadoop. We can keep the record of the user about their comments, the site they visit or any information about particular subject. This is the way we can also get to know about the habits of the user and its interests.

1.3. Problem Definition:

There is large number of data saved regularly and searched. Such data is difficult to store and calculate. Hadoop is one such software that helps to solve this problem. In big companies there are many employees. It is essential to track all the information about these employees so in this project we create a software using hadoop that extracts all the records off the employee and cluster them and store it. Large data can be stored using hadoop. We can keep the record of the user about their comments, the site they visit or any information about particular subject. This is the way we can also get to know about the habits of the user and its interests.

1.4. Applying software engineering approach:

The system will be developed keeping in mind the basic approaches followed in the waterfall model. Initially requirement analysis will help in exploring the existing systems. Then the scope of the system to be built will be analyzed. Then the design will be converted to code which will give the final application. Test cases will be generated and each unit of the system will be tested for efficiency and reliability. Thus the final error free application will be deployed.

II. SYSTEM FEATURES

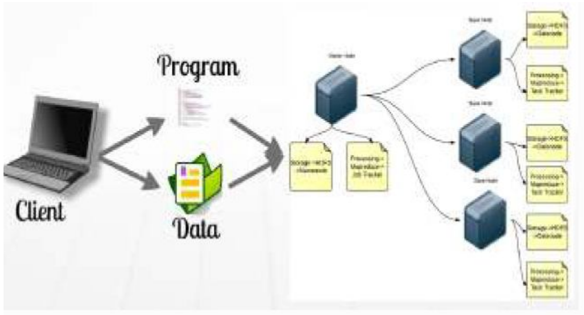

Figure (II): System Feature of the project

Manuscript received March, 23, 2014.

Kunal Oswal, Information Technology, University of Pune/ Trinity College of engineering/ KJ's Institute, Pune,India, Mobile No: 9604979366, (e-mail: kunaloswal137@gmail.com).
Saloni Mapara, Information Technology, University of Pune/ Trinity College of engineering/ KJ's Institute, Pune, India, Mobile No: 8551843399, (e-mail: maparasaloni3@gmail.com).
Asmita Deshmukh, Information Technology, University of Pune/ Trinity College of engineering/ KJ's Institute, Pune, India, Mobile No: 9730816204, (e-mail: asmitadeshmukh24@gmail.com).
Richa Runwal, Information Technology, University of Pune/ Trinity College of engineering/ KJ's Institute, Pune,India, Mobile No: 9922453908, (e-mail: runwalricha91@gmail.com).
Bilkis Chandargi, Information Technology, Assistant Professor at Trinity College of engineering/ KJ's Institute, Pune, India, Mobile No: 9923923123, (e-mail: bilkisbagmaru@gmail.com).

Working of Hadoop:

Client provides two things- 1.The data that has to be analyzed and 2.A program which will have functionality to analyze data.

Data is broken into files splits of 64 Mb or 128 Mb and blocks are moved to different nodes. Once all the blocks are moved the Hadoop framework passes on the program to each node. Job tracker then starts scheduling the program on individual node. Once all nodes are done the output is return back to the Data node (master) and displayed on local host.

III. HADOOP FRAMEWORK:

A.HDFS-Hadoop distributed file system:

The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. Each node in a Hadoop instance typically has a single namenode; a cluster of datanodes form the HDFS cluster. The situation is typical because each node does not require a datanode to be present. Each datanode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses the TCP/IP layer for communication. Clients use Remote procedure call (RPC) to communicate between each other.

HDFS stores large files (typically in the range of gigabytes to terabytes) across multiple machines. It achieves reliability by replicating the data across multiple hosts, and hence theoretically does not require RAID storage on hosts (but to increase I/O performance some RAID configurations are still useful). With the default replication value, 3, data is stored on three nodes: two on the same rack, and one on a different rack. Data nodes can communicate to each other to rebalance data, to move copies around, and to keep the replication of data high. HDFS is not fully POSIX-compliant, because the requirements for a POSIX file-system differ from the target goals for a Hadoop application. The tradeoff of not having a fully POSIX-compliant file-system is increased performance for data throughput and support for non-POSIX operations such as Append.

The HDFS file system includes a secondarynamenode, which misleads some people to thinkingthat when the primary namenode goes offline, the secondary namenode takes over. In fact, the secondary name node regularly connects with the primary name node and builds snapshots of the primary name node's directory information, which the system then saves to local or remote directories. These check pointed images can be used to restart a failed primary name node without having to replay the entire journal of file-system actions, then to edit the log to create an up-to-date directory structure. Because the name node is the single point for storage and management of metadata, it can become a bottleneck for supporting a huge number of files, especially a large number of small files. HDFS Federation, a new addition, aims to tackle this problem to a certain extent by allowing multiple name-spaces served by separate name nodes.

B. MapReduce Framework:

Hadoop Map Reduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (Thousands of nodes) of commodity hardware in a reliable, fault-tolerance.

A Map Reduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The Map Reduce framework consists of a single master Job Tracker and one or more slave Task Tracker per cluster-node. The master is responsible for scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks for minimum of 4 failure attempts.

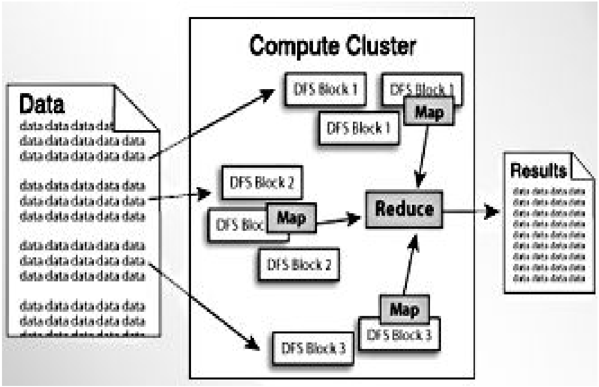

Figure (III): The Hadoop clustering of blocks of data

IV. SYSTEM ARCHITECTURE:

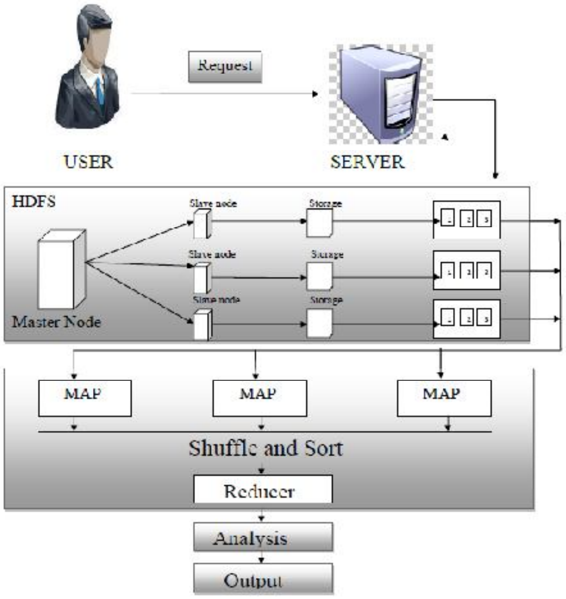

Figure (IV): The Architecture of extraction of data using Hadoop

Hadoop consists of the Hadoop Common package, which provides file system and OS level abstractions, a Map Reduce engine (either Map Reduce/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 small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a Job Tracker, Task Tracker, Name Node and Data Node. A slave or worker node acts as both a Data Node and Task Tracker, 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 start-up and shutdown scripts require Secure Shell (Ssh) to be set up between nodes in the cluster.

In a larger cluster, the HDFS is managed through a dedicated Name Node server to host the file system index, and a secondary Name Node that can generate snapshots of the name node’s memory structures, thus preventing file-system corruption and reducing loss of data. Similarly, a standalone Job Tracker server can manage job scheduling. In clusters where the Hadoop Map Reduce engine is deployed against an alternate file system, the NameNode, secondary NameNode and DataNode architecture of HDFS is replaced by the file-system-specific equivalent.

V.TECHNICAL SPECIFICATIONS:

A. Technology Details:

The system begins from the input component which enables the user to input his/her query. The system then fetches related links from original search engines, arranges them into a list according to some rules, and returns the top ranked results to the user. The user can see the results through a result displaying component, which displays the results in web pages. User also can interact with the system through this component. The user interaction module is then embedded into the user interface. In the interaction module, the user can choose some function to perform a specific task. In our system, the user can click a link if he/she finds the topic of the link is interesting.

The user can also mark the link as uninteresting after he/she reads the page. Otherwise, the system will think the user prefers this web page. Based on all the preferred pages, the system refines the user profile. The user can also choose to read abstracts. In this case, he/she may define the length of each abstract and the number of abstracts displayed. Since these two numbers will affect the speed of the abstracting task, user will learn to choose suitable numbers through trying. If the user does not assign these two numbers, the system will use default values.

VI. OUTPUT:

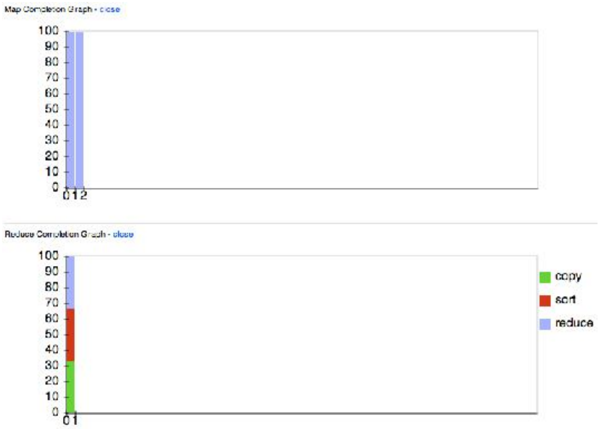

Figure (V): The Final Output in diagrammatic form

The diagram shows the output of the job done by mapper and reducer. In reducer, the three stages are shown that is copy i.e. replication of data next is sorting and then reducing process. This is how clustering of data is done using Hadoop. There is output file created where the actual analyzed data is created after map-reduce process. To check if the data node or server node is working or not one can check on local host. All information about the running jobs can be seen on local host and also the output i.e. the part file which is created.

The output file can also be downloaded. The data after the mapper and reducer job can be checked. This is how one can check output on local host. This output is the actual data from which interest of user can be found.

VII. ACKNOWLEDGEMENT

It is our pleasure to acknowledge sense of gratitude to all those who helped us in making this project a reality.

We thank our project guide **Prof.Bilkis Chandargi.** For helping us and providing all necessary information regarding our project.

We again thank to **Prof. Sarita Gaikwad.** (Head - Department of Information Technology) for providing us the required facilities and helping us while carrying out this project work.

Finally we wish to thank all our teachers and friends for their constructive comments, suggestions and criticism and all those directly or indirectly helped us in completing this project work.

VIII. CONCLUSION:

So this project will evidently make searching applications intelligent by mining the information and classifying and categorizing it automatically. This would mean identifying the patterns in the knowledge base using efficient data mining algorithms. Also the information being distributed many times over the cloud, it would also require ability to divide the work load over cloud and then merge the mined information. Thus, the project aim of increasing the efficiency of the help-desk persons as well as knowledge base administrators by providing them auto-generated knowledge classifications and cluster is achieved. Thus we made application which extracts user profile from large scale data. This we have achieved using Hadoop framework.

We proposed a scalable user profiling solution, implemented on top of Hadoop Map Reduce framework. Future work will include the extension of our framework with other profile models such as hierarchical or semantic models. We also intend to incorporate structured data sources into our framework. Thus, the project aim of increasing the efficiency of the help-desk persons as well as knowledge base administrators by providing them auto-generated knowledge classifications and cluster is achieved.

IX. REFERENCES:

[1] U. Cetintemel, M. J. Franklin, and C. L. Giles. Self-adaptive user profiles for large-scale data delivery. In ICDE, pages 622–633, 2000

[2] Y. Chen, D. Pavlov, and J. F. Canny. Large-scale behavioral targeting. In KDD '09, New York, NY, USA, 2009. ACM.

[3] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. In The Adaptive Web, volume 4321 of Lecture Notes in Computer Science. Berlin, Heidelberg 2007

[4] Yanagimoto and S. H. Omatu. User profile creation using genetic algorithm with kullbackleibler divergence. IEEJ Transactions on Electronics, Information and Systems,126:389–394, 2006.

[5] Michal Shmueli-Scheuer, Haggai Roitman, David Carmel, Yosi Mass, DavidKonopnicki

[6] IEEE,Extracting User Profiles from Large Scale Data, Michal Shmueli-Scheuer, Haggai Roitman, David Carmel, Yosi Mass, David,2009

[7] International Journal of Scientific & Engineering Research, Mapreduce Performance in Heterogeneous Environments: A Review, Salma Khalil, Sameh A.Salem, Salwa Nassar and Elsayed M.Saad, April -2013.

[8] International Journal of Scientific & Engineering Research, A SURVEY ON BIG DATA, Amegha.K, Sowmya.B, Apoorva M.P, July-2013

Kunal Oswal:
Qualification-B.E. in Information Technology

Saloni Mapara:
Qualification-B.E. in Information Technology
Paper Publication in 2013 to ICEISTCON, Topic: Cyborgs in Human Computer Interface.

Asmita Deshmukh:
Qualification-B.E. in Information Technology
Paper Publication in 2013 to IOAJ conference, Topic: Data Transforming device - REDTACTON

Richa Runwal:
Qualification-B.E. in Information Technology

Bilkis Chandargi:
Assistant Professor at Trinity College of Engineering
Qualification- B.E. in Information Technology