Map Reduce Recommendation System For Web Log Analytics

R. Boomathi[1], S. Gowthami[2], A. Kanimozhi[3]
[1, 2, 3] Department of CSE
[1, 2, 3] Kongunadu college of engineereing and Technology

Abstract- *Recommendation systems are found in many applications and these systems usually provide the user with a list of map reduce based on preference and prediction. By combining existing datasets, hybrid recommendation systems can be developed that considers both the job status and job completion time. I can import the web log dataset of size in Terabytes, a big data analysis tool such as Hadoop is used. Hadoop is a software framework for distributed processing of large data sets. Hadoop uses MapReduce paradigm to perform distributed processing over clusters of computers to minimize the time involved in analyzing the web log features. The proposed system is trustworthy and fault tolerant when compared to the existing recommendation systems as it collects the data from the user to predict the analysis and interest the item to find the features. The system is also adaptive as it updates the list frequently and finds the updated interest of the user. Experimental results show that the proposed system is much perfect than the existing recommender systems.*

Keywords- Recommendation System, Hadoop, Big Data, MapReduce, Web log data.

I. INTRODUCTION

A data is a collection of details from web servers usually of unstructured form in the digital universe. A large quantity of the data accessible in the internet is generated either by individuals, groups or by the organization over a meticulous period of time. The volume of data becomes bigger day by day as the procedure of World Wide Web makes an interpenal part of human activities. Rise of these data leads to a novel technology such as big data that acts as a tool to method, control and direct very large dataset along with the storage space required. Big Data is large volume, large velocity and variety information assets that insist cost-effective, inventive forum of information processing for improved insight and decision making. Big data, a buzz word that can be handle peta bytes or terabytes of data in a reasonable amount of time. Big data is separate from large existing database which uses Hadoop framework for data rigorous scattered applications. Big Data analytics apply higher analytical techniques of big datasets to find out hidden patterns and other useful information. It is performed using software tools mainly for predictive analysis and data mining. The mounting number of technologies is used to aggregate, manipulate, manage and analyze big data. The basic flow is described in fig 1.

II. RELATED WORK

P. Bhatotia [4] present a system called Incoop, which permit existing MapReduce programs, not calculated for incremental processing, to execute visibly in an incremental manner. In Incoop, calculation can respond repeatedly and professionally to modifications to their input data by reusing middle results from previous runs, and incrementally inform the output according to the modify in the input.

Y. Bu, [6] present Pregelix, a large-scale graph analytics system that we began in 2011. Pregelix obtain a novel set-oriented, iterative dataflow approach to apply the user level Pregel programming model. It achieves so by treating the messages and vertex states in a Pregel calculation like tuples with a well-defined schema; it then employ database-style query evaluation techniques to execute the user's program.

B. Howe, [7] provides HaLoop, a customized version of the Hadoop MapReduce framework that is planned to serve these applications. HaLoop not only extends MapReduce with programming support for iterative applications, it also considerably improves their efficiency by making the task scheduler loop-aware and by totaling various caching mechanisms.

J. Ekanayake, [10] implements Twister framework which is an improved MapReduce runtime with an extensive programming model that supports iterative MapReduce computations compenently. It uses a publish/subscribe messaging infrastructure for communication and data transfers, and supports long running map/reduce tasks, which can be used in "configure once and use many times" approach.

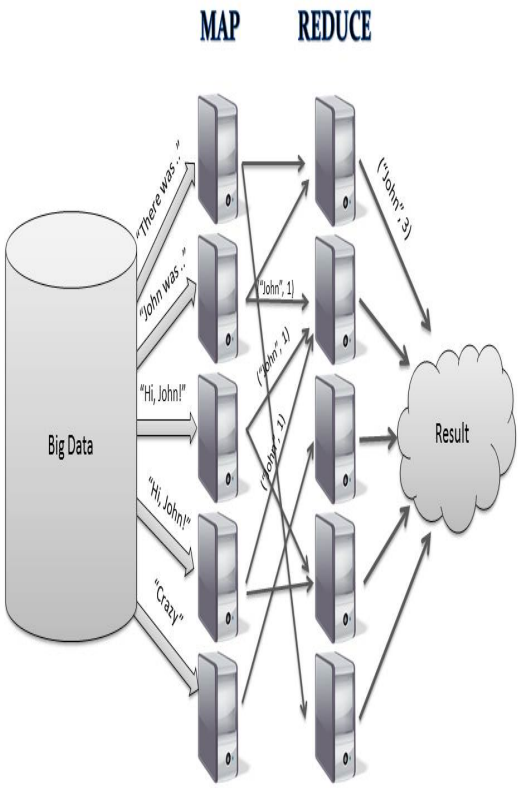

Fig 1: Map reduce function

S. Ewen, [1] propose a method to mix incremental iterations, a form of work-set iterations, through parallel data flows. After presentation how to mix bulk iterations into a dataflow system and its optimizer, we current an extension to the programming model for incremental iterations.

III. WEB LOG ANALYZING USING I^2 MAP REDUCE APPROACH

Website personalization is the process of customizing the content and formation of a website for specifically needs. Steps of personalization as

a) The collection of web data
b) Modeling and categorization of these data.
c) Analysis the collected data
d) Determination of the actions that should be performed.

A bipartite graph is a graph whose vertices can be partitioned into two subsets V_1 and V_2 such that no edge has both endpoints in the same subset, and every possible edge that could connect vertices in different subsets is part of the graph. That is, it is a bipartite graph (V_1, V_2, E) such that for every two vertices $v_1 \in V_1$ and $v_2 \in V_2$, v_1v_2 is an edge in E. A complete bipartite graph with partitions of size $|V_1|=m$ and $|V_2|=n$, is denoted $K_{m,n}$ every two graphs with the same notation are isomorphic. An alternative and equivalent form of this theorem is that the size of the maximum independent set plus the size of the maximum matching is equal to the number of vertices. In any graph without isolated vertices the size of the minimum edge cover plus the size of a maximum matching equals the number of vertices. Combining this equality with König's theorem leads to the facts that, in bipartite graphs, the size of the minimum edge cover is equal to the size of the maximum independent set, and the size of the minimum edge cover plus the size of the minimum vertex cover is equal to the number of vertices.

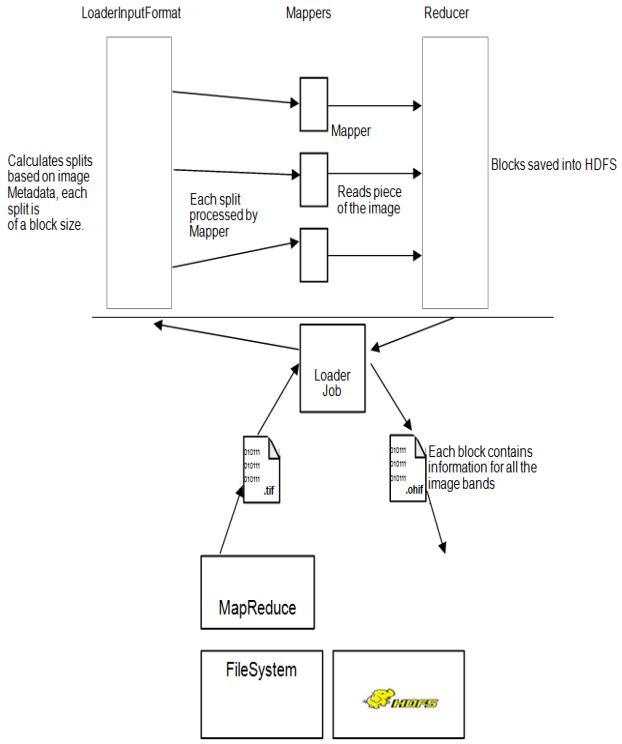

Fig 2: Job sequences

Servers store following information for every request. IP address, Date/time stamp, Status of request, Referring URL, Status of request, Type of user agent used software manufacturer and version no, Type of operation system, Network location and IP address: can include country, city or any other geographic data as well as the host name, Time of visit, Page visited, Time spent on each page of the website, Referring site statistics: can include the website you can through to reach this website and search engine query that brought there.

VI. MAP REDUCE RECOMMENDATION SYSTEM FOR WEB LOG DATA

4.1 Recommending Map reduce system

The idea of this system is to develop a recommendation engine that can recommend mapper and reducer to the users with increased accuracy by analyzing the process of the user and recommending the process. A hybrid recommender system is developed that gets its input from the user in the form of datasets. This list and the profile of the user are the key terms used to predict the interest of the user. The data set considered is a large set of web logs which is a big data. In order to analyze the features of the data set that is so large, I go for a tool named Hadoop. MapReduce programs have been written to find the feature. Preprocessing tasks are also performed in order to eliminate the missing values and to generate the tasks for each job. Recommender system framework is defined as follows:

Check job process for each Map and Reduce

- (1) Running state: The state when a compute node is working;
- (2) Idle state: If there are no tasks arriving at a compute node, the node goes through an idle period to avoid frequent switches from the deep sleep state. The threshold of idle period is T
- (3) Sleep state: After the idle period of T , if there are no incoming tasks, the compute node goes into sleep state.
- (4) Recovering state: When a task arrives at the compute node under sleep state, the compute node needs to recover and then start to execute the task.

Dynamic scheduling is that the task arrival is uncertain at run time and allocating resources are tedious as several tasks arrive at the same time. In case of dynamic scheduling information of the task components/task is not known before hand. Thus execution time of the task may not be known and the allocation of tasks is done on fly as the application executes Project focus on scheduling periodic and independent real-time tasks. Dynamic approach to create virtual clusters to deal with the conflict between parallel and serial tasks.

Tasks are dynamically available for scheduling over time by the scheduler. It is more flexible than static scheduling, to be able of determining run time in advance. It is more critical to include load balance as a main factor to obtain stable, accurate and efficient scheduler algorithm. A new recommendation method that take into consideration the make span conservation and energy reduction. The scheduling is devised with measure to identify the degree of task computation efficiency relative to the application completion time. The degree is used as a utilization value to identify a level of virtue for executing task on processor, and implied an effectively energy consumption of that processor. In this approach, the task load is adjusted automatically without running time prediction. Implement recommendation scheduling strategy to provide best tradeoff in task scheduling. Extend this work to analyze the processing speed for task completion. Dynamic scheduling algorithms for this scheduling mechanism have been introduced to generate scheduling with the shortest average execution time of tasks

Map Reduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A Map Reduce program is composed of a Map() procedure that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a Reduce() procedure that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies). The "Map Reduce System" (also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance.

Mined data:

In this module, we can implement pattern matching approach to construct the profile based on user id and keyword. User id and user name are used to construct the profile. The user profile is incrementally developed over time and it is stored for use in later sessions. The information exploited for constructing the profile usually comes from various sources, so it relies on different aspects of the user. On the other hand, in ephemeral preferences, the information used to construct each user profile is only gathered during the current session, and it is immediately exploited for executing some adaptive process aimed at personalizing the current interaction. In other words, since each user profile is computed based on term weights in a Web page the user browsed and the browsed pages are different according to each user, the profile is constructed in the form of a user-terms. This approach allows us to construct a more appropriate user profile and perform a fine-grained search that is better adapted to each user’s preferences.

Performance evaluation:

In this module, we can evaluate the performance of the system using time and accuracy metrics. The proposed approach presents an incremental data processing model

which is compatible with the Map Reduce model and its runtime. It supports Map Reduce-based applications without any modification. A part of this various things may be of help to the system administrator like, analysis of errors helps to know the problems while accessing the website, analysis of references to website during special event will help administrator to know and balance load, analysis of navigational patterns and duration will help the administrator with the knowledge about how to decrease the duration of the user by providing layout change, decrease in duration helps to usage of less bandwidth.

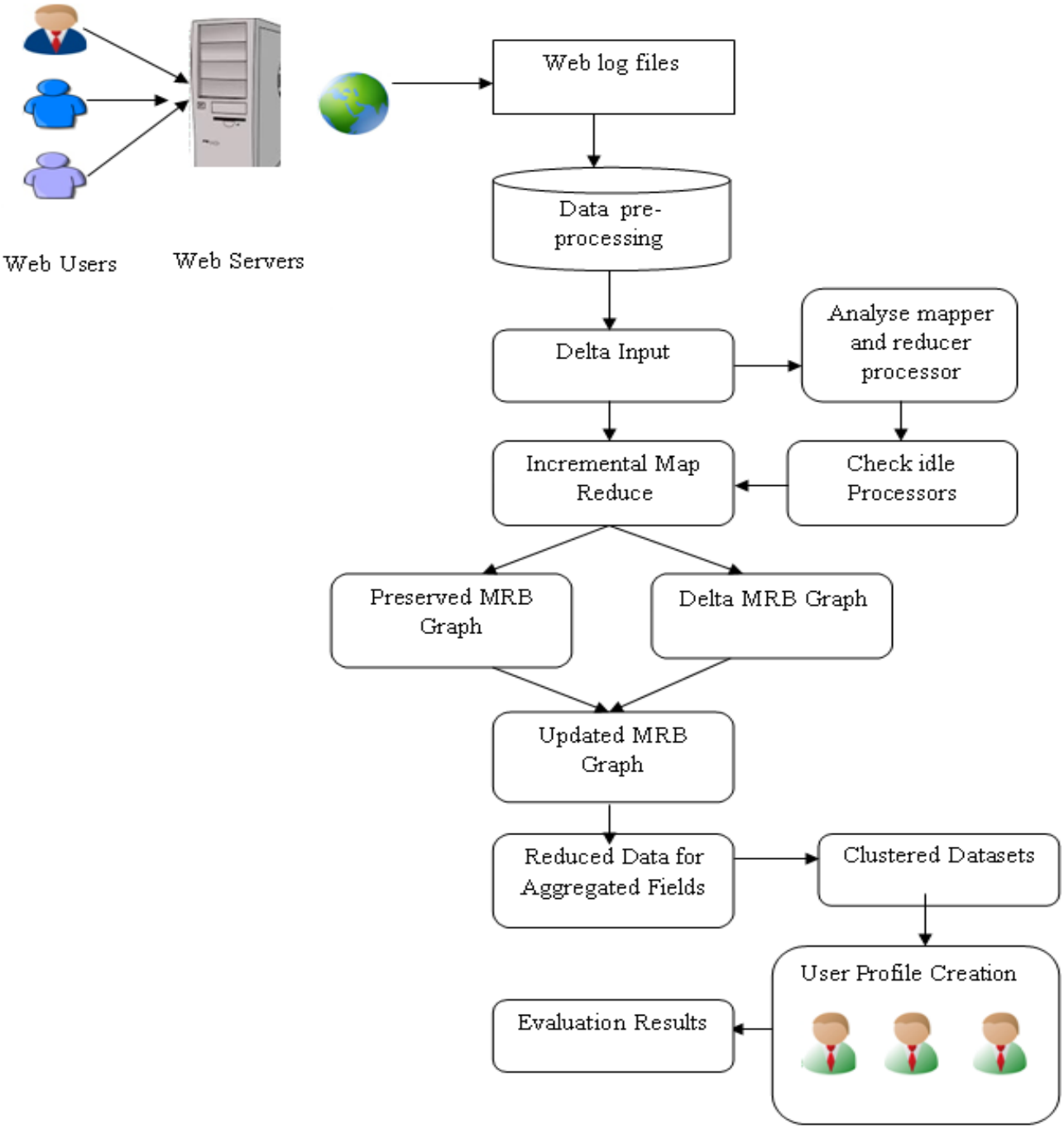

Fig 3: Map Reduce Framework

V. CONCLUSION

In this paper, I accomplished and optimized a recommendation system referring to the algorithm introduced by recommender system, based on the real-world web log dataset. The recommendation algorithm is essential a multiplication of the data with various jobs. It optimizes the multiplication adapted to Hadoop MapReduce. Finally, our recommendation program computes recommended web logs for different users. However, if I just use a single computer to

execute the recommendation program, it may take a very long time to finish it. In addition, a single machine has limited memory, storage space and computation capability, I have to partition the dataset into many pieces before I can handle them. This will make the processing of data extremely long and inefficient. Hadoop MapReduce provides us a great solution to process the dataset of very large scale.

ACKNOWLEDGEMENT

The authors wish to thank the reviewers for their valuable feedback that resulted in an improved paper and we would also like to thank our guide for assisting with code development.

REFERENCES

[1] A Ramachandran "Individualized Travel Recommendation By Mining People Ascribes And Travel Logs Types From Community Imparted Pictures". (IEEE-2013)

[2] Bhatotia P., Wieder A., Rodrigues R., Acar U.R. and Pasquin R.(2011),'Incoop: Mapreduce for incremental computations', In Proc. of SOCC '11.

[3] Brian McFee, Luke Barrington and Gert Lanckriet, "Learning Content Similarity for Music Recommendation" IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20, No. 8, 2012.

[4] Bu Y., Borkar V., Jia J., Carey M.J., and Condie T.(2015),'Pregelix: Big(ger) graph analytics on a dataflow engine',PVLDB, 8(2):161–172.

[5] Bu Y., Howe B., Balazinska M. and Ernst M.D.(2010),'Haloop: efficient iterative data processing on large clusters.' PVLDB, 3(1-2):285–296.

[6] Ekanayake J., Li H., Zhang B., Gunarathne T., Bae S.H., Qiu J. and Fox G.(2010), 'Twister: a runtime for iterative mapreduce',In Proc. of MAPREDUCE '10.

[7] Emmanouil Vozalis, Konstantinos G. Margaritis, " Analysis of Recommender Systems‟ Algorithms", conference proceeding of IEEE.

[8] Ewen S., Tzoumas K., Kaufmann M. and Markl V.(2012), 'Spinning fast iterative data flows', PVLDB, 5(11):1268–1279.

[9] Fay Chang, Jeffrey Dean, "Bigtable: A Distributed Storage System For Structured Data". (IEEE-2013)

[10] Logothetis D., Olston C., Reed B., Webb K.C. and Yocum K.(2010),'Stateful bulk processing for incremental analytics', In Proc. of SOCC '10.

[11] Low Y., Bickson D., Gonzalez J., Guestrin C., KyrolaA.andHellerstein J.M. (2012), 'Distributed graphlab: a framework for machine learning and data mining in the cloud', PVLDB, 5(8):716–727.

[12] Malewicz G., Austern M.H., Bik A.J., Dehnert J.C., Horn I., Leiser N. and Czajkowski G.(2010),'Pregel: a system for large-scale graph processing', In Proc. Of SIGMOD '10.

[13] Mihaylov S.R., Ives Z.G. and Guha S.(2012),'Rex: recursive, delta-based data-centric computation', PVLDB, 5(11):1280–1291.

[14] Murray D.G., McSherry F., Isaacs R., Isard M., Barham P. and Abadi M. (2013), 'Naiad: A timely dataflow system', In Proc. of SOSP '13, pages 439–455.

[15] Paul C.Zikopolus and Chris Eaton, " Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data", thesis, 2013.

[16] Rafael Sotelo Jose, Joskowicz Alberto, "An Affordable and Inclusive System to Provide Contents to DTV Using Recommender System".(IEEE-2014)

[17] Yasha Sardey, Pranoti Deshmukh "A Mobile Application For Bus Information System And Location Tracking Using Client-Server Technology". (IEEE-2014)