ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

Optimization of Map Reduce Function on Hadoop with Ishuffle

Aravind R[1], Suriya B[2], Rejinpaul N R[3]

U.G Students, Dept. of CSE, Velammal Institute of Technology, Chennai, Tamil Nadu, India [1,2]

Associate Professor, Dept. of CSE, Velammal Institute of Technology, Chennai, Tamil Nadu, India[3]

ABSTRACT: Hadoop is a popular implementation of the MapReduce framework for running data-intensive jobs on clusters of commodity servers. Shuffle, the all-to-all input data fetching phase between the map and reduce phase can significantly affect job performance. However, the shuffle phase and reduce phase are coupled together in Hadoop and the shuffle can only be performed by running the reduce tasks. This leaves the potential parallelism between multiple waves of a map and reduces unexploited and resource wastage in multi-tenant Hadoop clusters. Thus we decouple shuffle from reduce tasks and convert it into a platform service provided by Hadoop called iShuffle. We propose an E-commerce based web application that collects data from the existing service providers like Amazon, Flipkart and stores it in our backend. This large volume of data (usually in the form of tab separated values TSV)is then batch processed. That parsed data is then map reduced using iShuffle which provides results much faster than the existing system. It also provides User Based Recommendation for the new user as well as existing users. The application has the payment gateway for buying the suggested product from the best service provider. Thus it allows us to compare the products and buy the best one.

KEYWORDS: MapReduce, shuffle, Hadoop, Relevance clustering

I. INTRODUCTION

MapReduce is a notable programming model, created inside Google, for handling a lot of crude information, for instance, slithered records or web ask for logs. This data is usually so large that it must be distributed across thousands of machines in order to be processed in a reasonable time. The simplicity of programmability, programmed information administration and straightforward adaptation to non-critical failure has settled on MapReduce a good decision for extensive scale server farms clump preparing. Map, written by a user of the MapReduce library, takes an input pair and produces a set of intermediate key/value pairs. The library bunches together all middle esteems related with a similar moderate key and passes them to the decrease work through an all-outline all-diminish correspondence called Shuffle. Reduce, also written by the user, receives intermediate key along with a set of values from Map and merges together these values to produce the final output. Hadoop is an open-source execution of MapReduce which is being enhanced and grown frequently by programming designers/scientists and is kept up by Apache Software Foundation It is a Java-based programming framework that supports the processing of large datasets in a Parallel and distributed computing environment. It makes Use of the commodity hardware Hadoop is Highly Scalable and Fault Tolerant. Hadoop runs in a cluster and eliminates the use of a Supercomputer. Hadoop is the widely used big data processing engine with a simple master-slave setup. Along with the above example, Flicker, a public picture sharing site, which received 1.8 million photos per day, on average, from February to March 2012. Assuming the size of each photo is 2 megabytes (MB), this requires 3.6 terabytes (TB) storage every single day. Indeed, as an old saying states: “a picture is worth a thousand words,” the billions of pictures on Flicker are a treasure tank for us to explore the human society, social events, public affairs, disasters, and so on, only if we have the power to harness the enormous amount of data. The above cases show the ascent of Big Data applications where information gathering has developed enormously and is past the capacity of normally utilized programming instruments to catch, oversee, and process inside an "average

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

passed time." The most fundamental challenge for Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions By and large, the learning extraction process must be extremely productive and near continuous on the grounds that putting away all watched information is about infeasible Therefore, the extraordinary information volumes require a compelling information examination and expectation stage to accomplish quick reaction and ongoing grouping for such enormous information. for such Big Data.

II. LITERATURE REVIEW

In 2014, Faraz Ahmad, Srimat T.Chakradhar, Anand Ragunathan, and T. N. Vijaykumar published a paper about "ShuffleWatcher: Shuffle-aware scheduling in multi-tenant mapreduce clusters," Shuffle, We propose ShuffleWatcher, a new multi-tenant MapReduce scheduler that shapes and reduces Shuffle traffic to improve cluster performance while operating within specified fairness constraints. ShuffleWatcher employs three key techniques. Initially, it checks intra-work Map-Shuffle simultaneousness to shape Shuffle movement by deferring or prolonging a vocation's Shuffle in view of the system stack.Second, it exploits the reduced intra-job concurrency and the flexibility engendered by the replication of Map input data for fault tolerance to preferentially assign a job's Map tasks to localize the Map output to as few nodes as possible. Third, it abuses confined Map yield and deferred Shuffle to lessen the Shuffle movement by specially appointing a vocation's Reduce assignments to the hubs containing its Map yield.

In 2013,Faraz Ahmad, Seyong Lee, Mithuna Thottethodi and T. N. Vijaykumar published a paper about"MapReduce with communication overlap (marco). MapReduce is a programming model from Google for cluster-based computing in domains such as search engines, machine learning, and data mining MapReduce gives programmed information administration and adaptation to non-critical failure to enhance programmability of bunches. MapReduce's execution show incorporates an all-outline all-lessen correspondence, called the shuffle, over the system division Some MapReductions move large amounts of data (e.g., as much as the input data), stressing the bisection bandwidth and introducing significant runtime overhead. Optimizing such shuffle-heavy MapReductions is important because (1) they include key applications (e.g., inverted indexing for search engines and data clustering for machine learning) and (2) they run longer than shuffle-light MapReductions (e.g., 5x longer). In MapReduce, the offbeat idea of the rearrange brings about some cover between the rearrange and guide. Tragically, this cover is inadequate in rearrange overwhelming MapReductions.

In 2013, Rohan Gandhi, Di Xie, Y. Charlie Hu published a paper about, "How to Rebalance Load in Optimizing MapReduce On Heterogeneous Clusters" the key design challenge in this important optimization for MapReduce on heterogeneous clusters and make three contributions. We show that Tarazu estimates the target load distribution too early into MapReduce job execution, which results in the rebalanced load far from the optimal We articulate the delicate tradeoff between the estimation accuracy versus wasted work from delayed load adjustment, and propose a load rebalancing scheme that strikes a balance between the tradeoff. We implement our design in the task scheduler

In 2012,Faraz Ahmad, Seyong Lee, Mithuna Thottethodi and T. N. Vijaykumar published a paper about"Tarazu: Optimizing MapReduce on heterogeneous clusters," MapReduce's poor performance on heterogeneous clusters. Our first contribution is that the poor performance is due to two key factors: the non-intuitive effect that MapReduce's built-in load balancing results in excessive and bursty network communication during the Map phase, and the intuitive effect that the heterogeneity amplifies load imbalance in the Reduce computation. Our second contribution is Tarazu, a suite of optimizations to improve MapReduce performance on heterogeneous clusters. Tarazu consists of Communication-Aware Load Balancing of Map computation across the nodes, Communication-Aware Scheduling of Map computation to avoid bursty network traffic and Predictive Load Balancing of Reduce computation across the nodes.

In 2011,R. C. Chiang and H. H. Huang published a paper about "TRACON: Impedance mindful planning for information serious applications in virtualized situations," In this work, TRACON, a novel Task and Resource Allocation Control structure that mitigates the obstruction impacts from simultaneous information concentrated

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

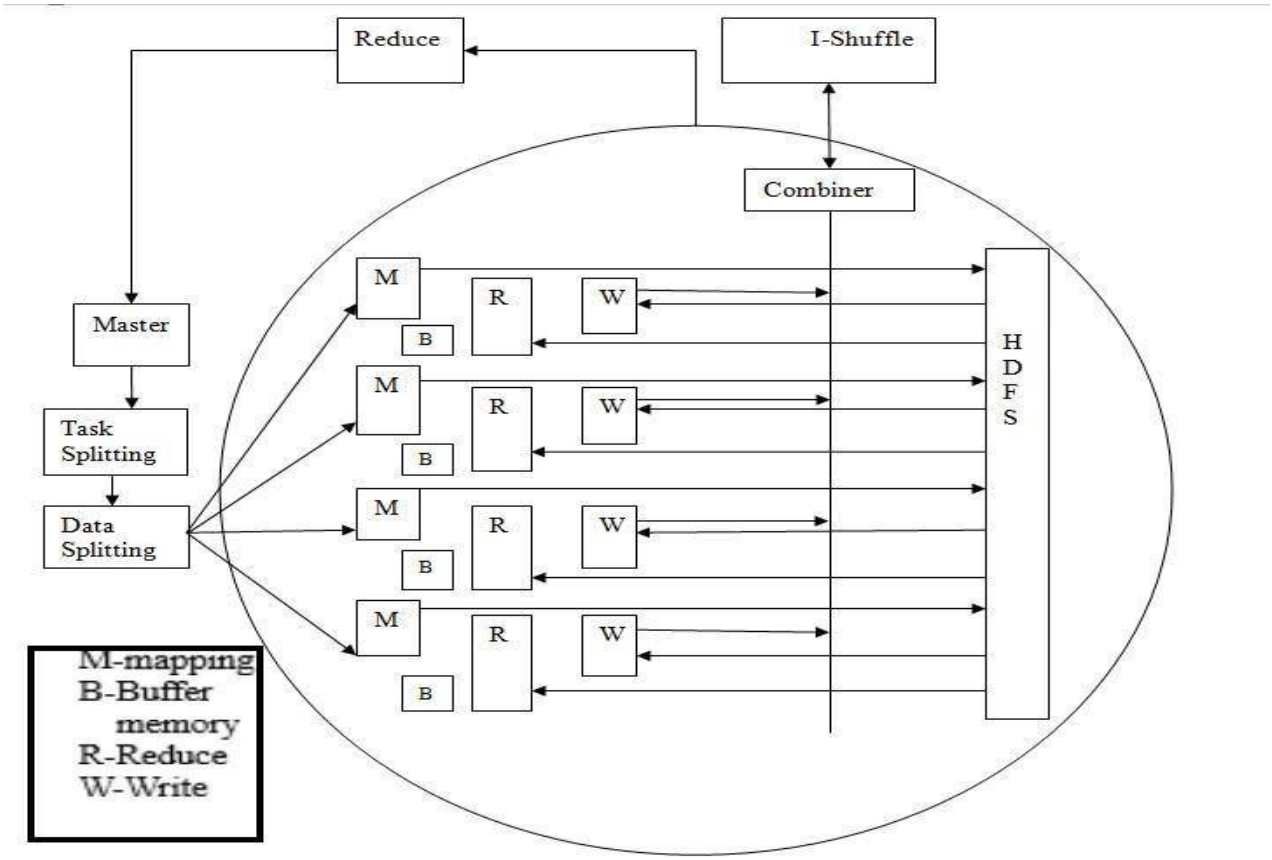
Vol. 7, Special Issue 2, March 2018

applications and extraordinarily enhances the general framework execution. TRACON uses displaying and control procedures from measurable machine learning and comprises of three noteworthy parts: the obstruction forecast demonstrate that derives application execution from asset utilization saw from various VMs, the impedance mindful scheduler that is intended to use the model for powerful asset administration, and the undertaking and asset screen that gathers application qualities at the runtime for show adaption. We recreate TRACON with a wide assortment of information escalated applications including bioinformatics, information mining, video preparing, email and web servers, and so on.

III. PROPOSED SYSTEM

To decouple rearrange from diminish assignments and change over it into a stage benefit gave by Hadoop. We show iShuffle, a client straightforward rearrange benefit that master effectively pushes outline information to hubs by means of a novel rearrange on-compose activity and adaptably plans diminish undertakings considering workload adjust. It covers the information rearranging of any diminish undertaking with the guide stage, addresses the information skew in decrease assignments, and empowers effective lessen planning. We executed iShuffle on a Hadoop bunch and assessed its advantages utilizing benchmark employments from the Purdue MapReduce Benchmark Suite (PUMA) and the HiBench with E-Commerce datasets gathered from genuine applications. We looked at the execution of iShuffle running both rearrange substantial and rearrange light workloads with that of stock Hadoop.

SYSTEM DESIGN



ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

A. BIG DATA ENVIRONMENT

Sample web applications were built so that the users can compare their products with different service providers. The applications use sample datasets that have been crawled in Amazon previously. Similar datasets were prepared for other applications too using the meta-model that has been crawled earlier. Each data set was loaded independently in various web applications. Features and other specifications have been loaded differently for each application based on the service providers requirement. These applications have been deployed in web servers so that the application is up and running. Web services have been written on each web application so that any third party can communicate with secure authentication.

B. OUR GATEWAY APPLICATION AND BATCH PROCESSING OVER THE TSV DATA

Now Our Gateway Application is built which gives users with Recommendations and Comparisons between the Products in the Market. Generally, the Resources provided by Various Web Servers are in TSV (Tab Separated Values) Format and should be Batch Processed before Proceeding. For that, we use our own API for TSV Manipulation. The TSV files were parsed for data. These data's are used for further processing (i.e. For Recommendation and comparison).

C. WEB CRAWLING FOR RESOURCES AND MAPREDUCE

The Users can register and can login to view Various Products Available in Market. This is done by writing a Web Service Client Process for each Service provider. It can connect to the Various Web Applications Web Service and can pull all the needed data to our backend. A huge amount of data got accumulated now. Web crawling looks for web services provided by various web applications. The Crawled Resources are then reduced by MapReduce Framework and converted into a single object. This Reduced Object Contains all the necessary information for providing comparison and Recommendations. Map Reduce – It is a combination of a map task and multiple Reduce Tasks which is used for clubbing all the slave process outcomes.

D. PICKING PRODUCTS FROM RECOMMENDATIONS AND PURCHASE

The Recommendations were given based on the QOS, Availability, Delivery, Offers, Price, and Specifications of the particular product. The Users can pick any product so that our application provides with a most Genuine Recommendation and a set of Comparisons. The Users are provided with neat and clean indexes so that he can pick the best provider for a particular product. The picked products were added to Cart and can be purchased later. The User Cart is equipped with Case-Based Recommender Systems. It uses case-based reasoning (CBR) to identify and recommend the items that seem more suitable for completing a user's buying experience provided that he or she has already selected some items. The system models complete transactions as cases and recommended items come from the evaluation of those transactions. Because the cases aren't restricted to the user who purchased them, the developed system can generate accurate item recommendations for joint item selections, both for new and existing users. Having analyzed the previous transactions and identified the concepts within which concrete items appear, the given part of a new transaction is matched over the existing. Ones to find the adequate solution. i.e. the best way to fill this basket. When the User initiates Transaction our Gateway will connect to the Banking Web Services directly on behalf of the Service Provider and Completes the transaction securely with help of OTP sent to their mail id given on User Registration. A Bank Account is needed for Complete the Transaction which can be created earlier through our Banking Application. The Process will be back to our Application as soon as the Transaction is over and the Purchased products will be reflected on the Bag List. i.e. Purchased Items List.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

PROPOSED ALGORITHM

Clustering is widely used data mining model that partitions data points into a set of groups, each of which is called a cluster. With the emerging growth of computational biology and e-commerce applications, high dimensional data becomes very common. Thus, mining high-dimensional data is more needed. Relevant clustering algorithm work can be done in three steps. First step elimination of irrelevant features from the dataset; the relevant features are selected by the features having the value greater than the predefined threshold. In the second step selected relevant features are used to generate the graph, divide the features using graph theoretic
method, and then clusters are formed by using Minimum Spanning Tree. In the third step find the subsets features that are more related to the target class is selected.

IV. RESULTS

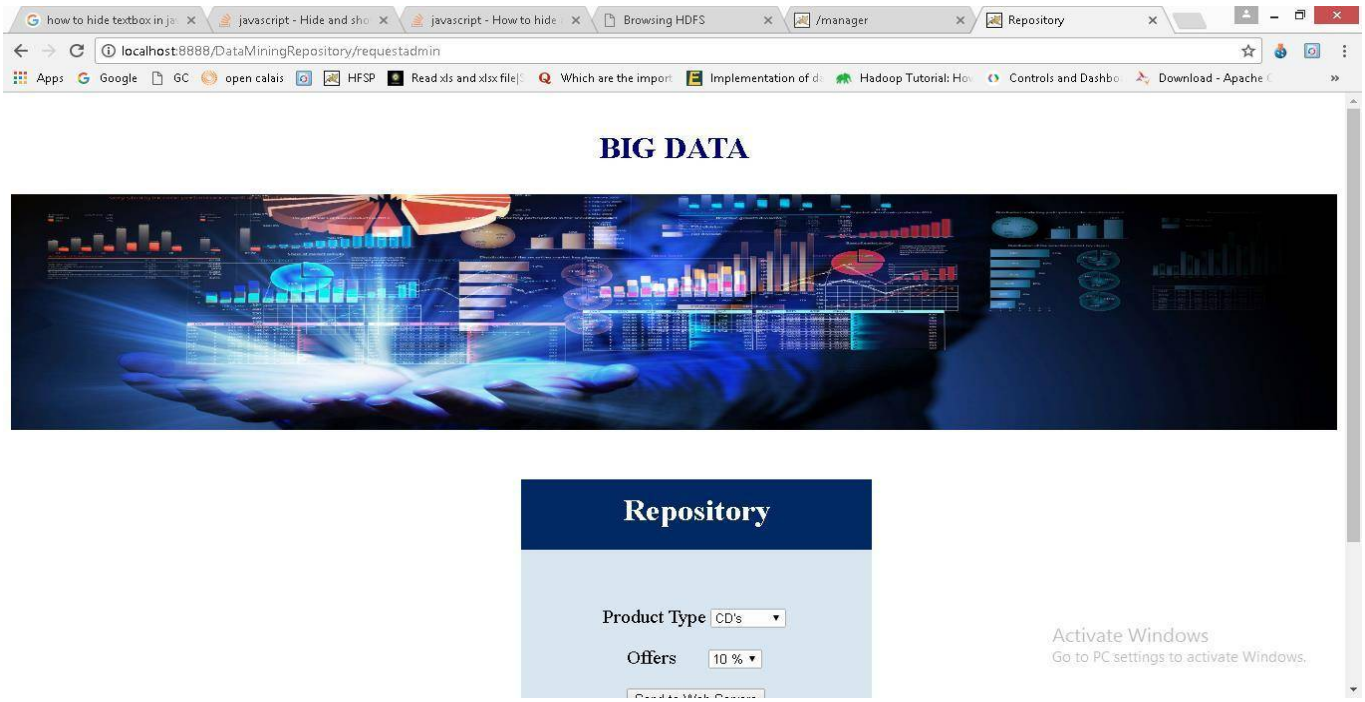


Fig.1
In Fig.1, It involves distribution of structured data to the slave nodes

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

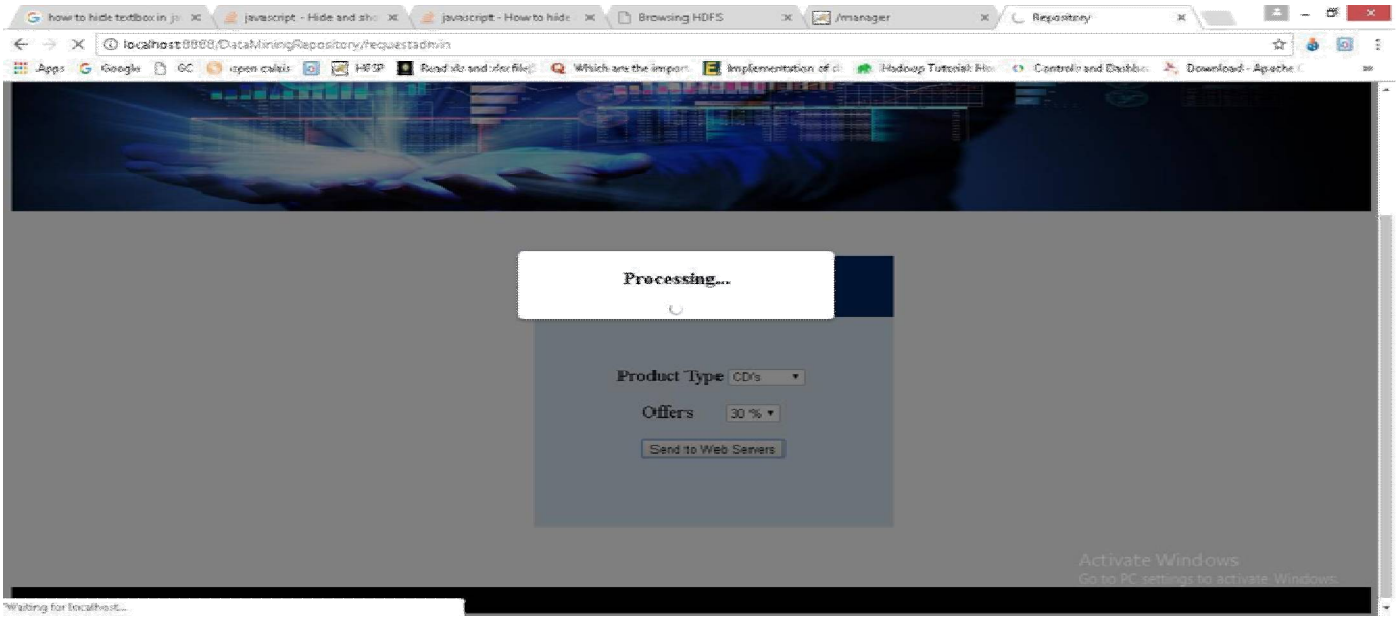


Fig.2

In Fig.2, The raw unstructured data is modified to structured data and then distributed. The products are distributed along with a certain discount percentage. The discount percentage is added to mimic the workflow of an e-commerce website.

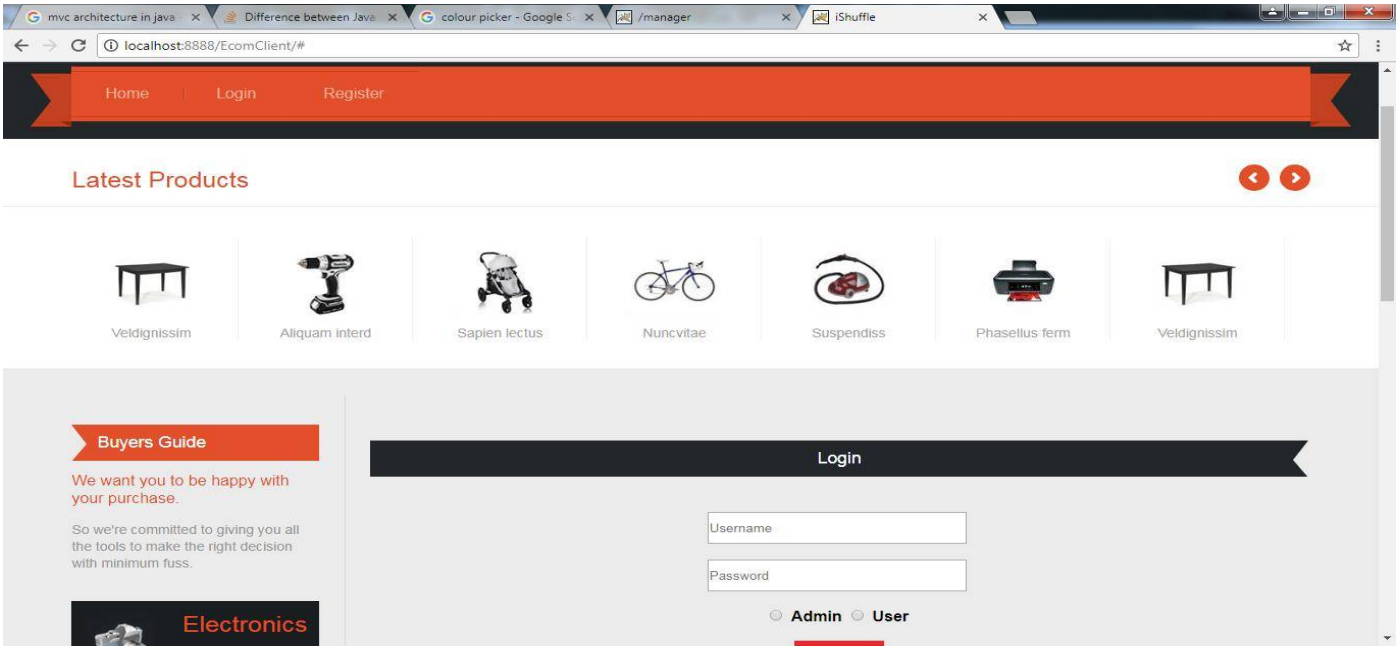


Fig.3

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

In Fig.3,it involves sign in process to the e-commerce website, by giving proper credentials.

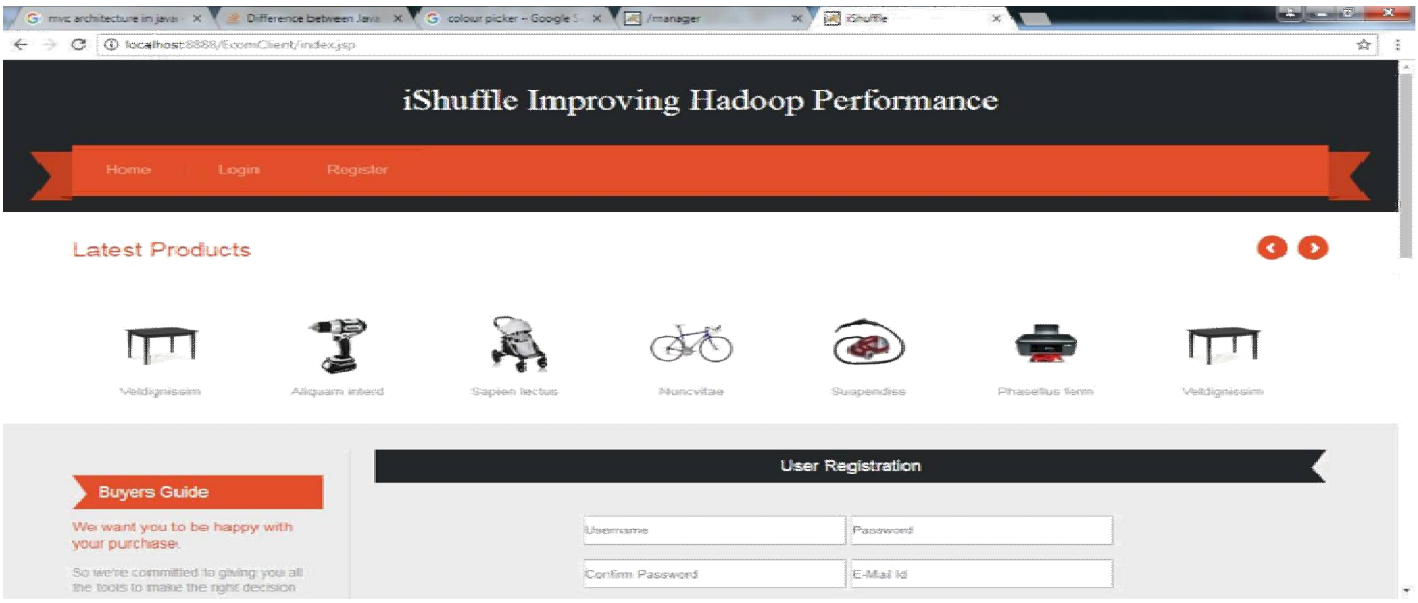


Fig.4

In Fig.4, it involves admin login, to gain access to the necesssary product details. By logging in, the distributed data can be processed, by selecting each type of product and their corresponding attributes. This is where reduce property is applied when the tasks are completed 50%.

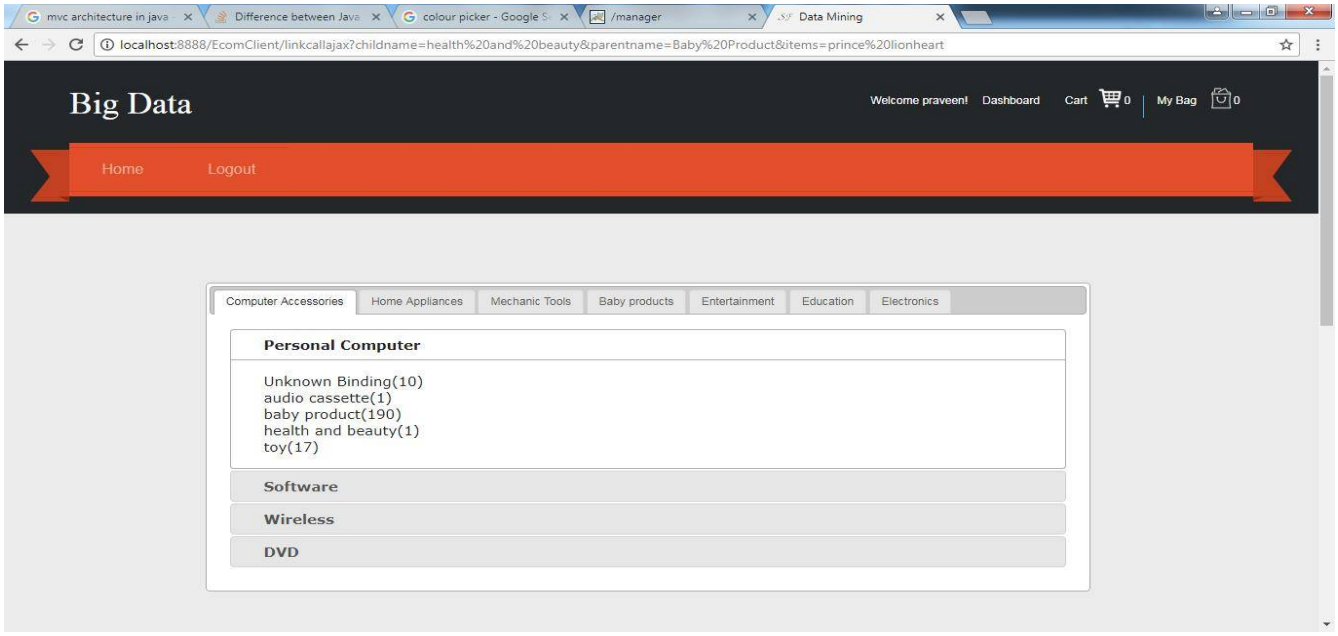


Fig.5

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

In Fig.5, this is the second stage of user login, to view the products listed, that were processed in the individual slave nodes.

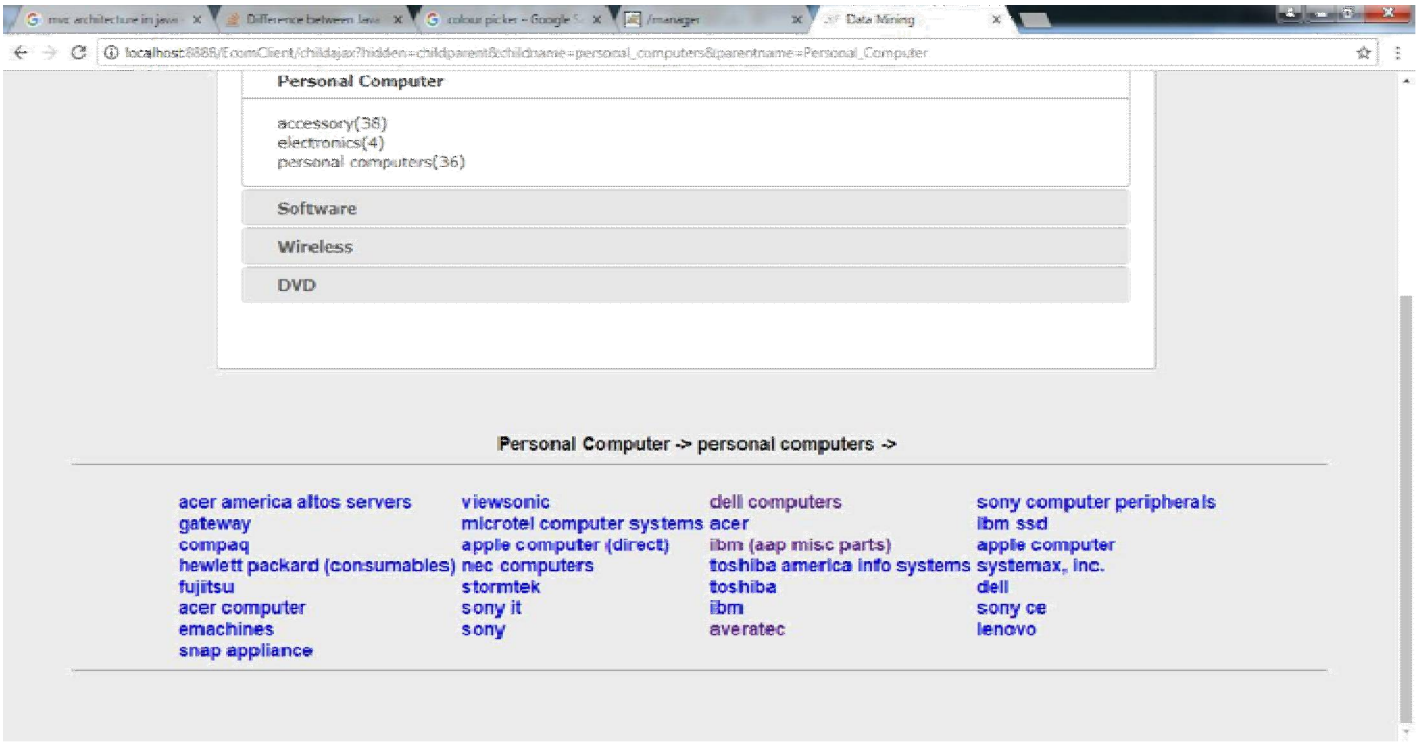


Fig.6

In Fig.6, The products and their corresponding attributes are listed, when a product is selected.

V. CONCLUSION

Thus the Broadcasted data's are received using Web services through SOAP protocol and batch processed in Hadoop. Map Reduce is done for batch job's output and the service level comparison is shown for the selected products. Recommendations are also given using Case based Recommendations and the transaction process is made. Hence we proposed a scalable efficient error prone system with Hadoop for big data from various providers through broadcasting technique.

REFERENCES

[1] Apache Hadoop Project. (2013). [Online]. Available: http://hadoop.apache.org
[2] HiBench. (2015). [Online]. Available: https://github.com/intelhadoop/HiBench
[3] PUMA: Purdue mapreduce benchmark suite. (2012). [Online].Available: http://web.ics.purdue.edu/_fahmad/benchmarks.htm
[4] Manual of Elasticsearch for Apache Hadoop. (2015). [Online].Available: https://www.elastic.co/guide/en/elasticsearch/hadoop/current/configuration-runtime.html
[5] SWIM: Statistical Workload Injection for MapReduce. (2012).[Online]. Available: https://github.com/SWIMProjectUCB/SWIM/wiki
[6] F. Ahmad, S. T. Chakradhar, R. Anand, and T. N. Vijaykumar, "ShuffleWatcher: Shuffle-aware scheduling in multi-tenant mapreduce clusters," in Proc. USENIX Conf. USENIX Annu. Techn.Conf., 2014, pp. 1–12.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Special Issue 2, March 2018

[7] F.Ahmad, S. T.Chakradhar, A. Raghunathan, and T.N.Vijaykumar, "Tarazu: Optimizing MapReduce on heterogeneous clusters," in Proc. Int Conf. Archit. Support Program. Languages Operating Syst., 2012, pp. 61–74.
[8] F. Ahmad, S. Lee, M. Thottethodi, and T. N. Vijaykumar, "MapReduce with communication overlap (marco)," J. Parallel Distrib.Comput., vol. 73, no. 5, pp. 608–620, May2013.
[9] G. Ananthanarayanan, et al., "Scarlett: Coping with skewed content popularity in MapReduce clusters," in Proc. ACM Eur. Conf. Comput. Syst., 2011, pp. 287–300.
[10] Y. Chen, S. Akspaugh, and R. Katz, "Interactive analytical processing of big data systems: A cross-industry study of mapredyce workloads," in Proc. VLDB Endowment, 2012, pp. 1802–1813.
[11] R. C. Chiang and H. H. Huang, "TRACON: Interference-aware scheduling for data-intensive applications in virtualized environments," in Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal. (SC), 2011, pp. 1–12.
[12] T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears, "MapReduce online," in Proc. 7th USENIX Conf. Netw. Syst. Des. Implementation, 2010, pp. 21–21.
[13] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in Proc. 6th Conf. Symp. Opearting S yst. Des.