www.ioirp.com
International Journal of Innovative Research in Computer Science and Engineering (IJIRCSE)
ISSN: 2394-6364, Volume – 2, Issue – 2. February 2016

I^2 MAP REDUCE: MAP REDUCE-BASED FRAMEWORK FOR INCREMENTAL BIG DATA MINING

Jhansi Rani S

Department of Computer Science and Engineering
K. Ramakrishnan College of Technology, Samayapuram
Tiruchirapalli, India
Email id: subu.janu2010@gmail.com

***Abstract*— Seeing that new data and updates are continuously arriving, the results of data mining applications turn out to be stale and obsolete over time. Incremental processing is a talented move towards to refreshing mining results. It utilizes previously saved states to avoid the expense of re-computation from scratch. Suggest i²MapReduce, a work of fiction incremental processing extension to MapReduce, the most widely used structure for mining big data. Compared with the state-of-the-art work on Incoop, i²MapReduce performs key-value pair level incremental processing somewhat than task level re-computation, supports not only one-step computation but also more sophisticated iterative computation, which is widely used in data mining applications, and incorporates a set of tale techniques to reduce I/O overhead for accessing preserved fine-grain computation states. Evaluate i² MapReduce using a one-step algorithm and four iterative algorithms with diverse computation description.**

***Keywords*— Cloud data, Map Reduce framework, Incremental processing, Job level task, Fine grained results**

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 interdisciplinary 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 intensive distributed applications. 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. Incoop detects changes to the inputs and enables the automatic update of the outputs by employing an efficient, fine-grained result re-use mechanism[1].

As computer systems create and collect growing amounts of data, analyze it becomes a basic part of improving the services provided by Internet companies. In this context, the Map Reduce structure offers techniques for suitable, distributed processing of data by enable a simple programming model that remove the burden of apply a complex logic or infrastructure for parallelization, data transfer, scalability, fault tolerance and scheduling. A vital property of the workloads method by Map Reduce applications is that they are often incremental by nature; i.e., Map Reduce jobs often run frequently with small changes in their input. In the architecture, implementation, and evaluation of a vital Map Reduce framework, named I^2 map reduce framework, for incremental computations. I^2 map reduce notice changes to the inputs and allow the automatic update of the outputs by employing an efficient, fine-grained result re-use mechanism.

II. INCREMENTAL AND ITERATIVE MAPREDUCE(I^2 MAPREDUCE)

Map Reduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key [3]. IMapReduce a framework that supports iterative processing. IMapReduce allows users to specify the iterative operations with map and reduce functions, while supporting the iterative processing automatically without the need of users involvement [7].IncMR framework for incrementally processing new data of a large data set, which takes state as implicit input and combines it with new data. Map tasks are created according to new splits instead of entire splits while reduce tasks fetch their inputs including the state and the intermediate results of new map tasks from designate nodes or local nodes[6].In the proposed system, the I^2 MapReduce a novel incremental processing extension to MapReduce, the most widely used framework for mining big data.I^2 MapReduce key-value pair level incremental processing rather than task level re-computation, supports not only one-step computation but also more sophisticated iterative computation, which is widely used in data mining applications, and incorporates a set of novel techniques to reduce I/O overhead for accessing preserved fine-grain computation states

In existing system, analyze the web logs using k means clustering. Clustering is an unsupervised classification and widely used for mining web usages with main objective to grouping a known collection of unlabeled objects into evocative clusters. Then cluster the web logs using page rank algorithm which play a major role in making the user search Navigation easier in the results of a search engine, which helps in best utilization web resources by providing required information to the Navigator. The PageRank algorithm computes ranking scores of web pages based on the web graph structure for supporting web search. However, the web graph structure is constantly evolving web pages and hyper-links are created, deleted, and updated. As the underlying web graph evolves, the PageRank ranking results gradually become stale, potentially lowering the quality of web search. Therefore, it is desirable to refresh the PageRank computation regularly. Given the size of the input big data, it is often very expensive to rerun the entire computation from scratch. Incremental processing exploits the fact that the input data of two subsequent computations A and B are similar. Only a very small fraction of the input data has changed. The idea is to save states in computation A, re-use A's states in computation B, and perform re-computation only for states that are affected by the changed input data.

III. INCREMENTAL AND ITERATIVE MAPREDUCE (I^2 MAPREDUCE) WORKING

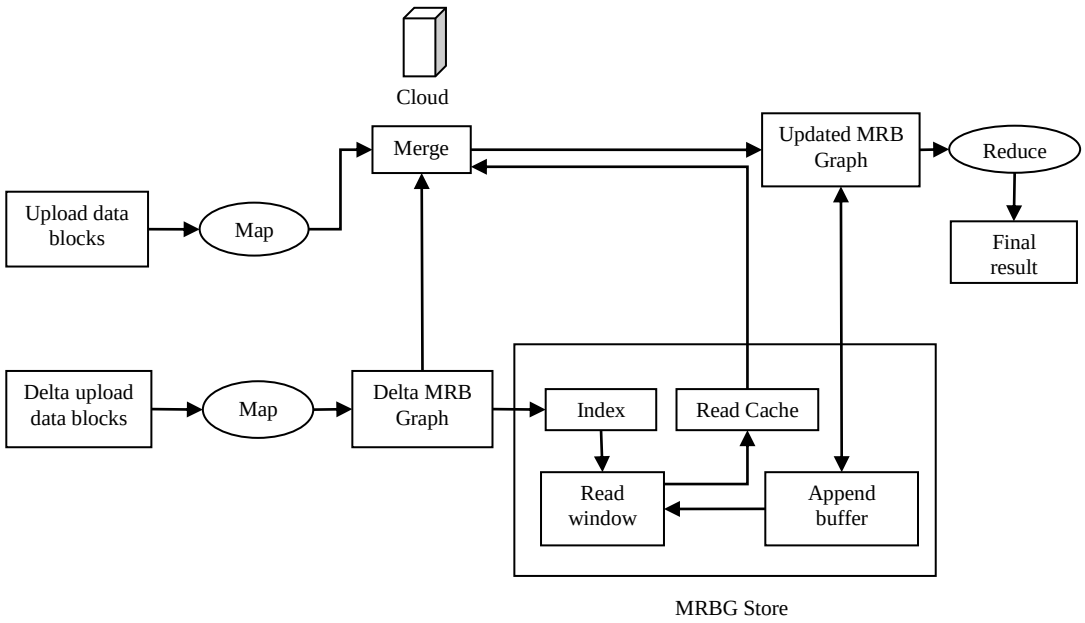

Figure 1.1 System Architecture

Figure 1.1 describes the system architecture, i^2 MapReduce exploits a fine-grain kv-pair level re-computation that are more advantageous. Incremental processing for iterative application, Proposes a timely

dataflow paradigm that allows stateful computation and arbitrary nested iterations. To support incremental iterative computation, programmers have to completely rewrite their MapReduce programs. In comparison, extend the widely used MapReduce model for incremental iterative computation. Existing Map-Reduce programs can be slightly changed to run on i^2 MapReduce for incremental processing .Implement i^2MapReduce by modifying Hadoop 1.0.3. Evaluate i^2MapReduce using a one-step algorithm (A-Priori) and four iterative algorithms (PageRank, SSSP, K-means, GIM-V) with diverse computation characteristics. Experimental results on Sample E- Commerce Application show significant performance improvements of i^2MapReduce compared to both plain and iterative MapReduce performing re-computation. For example, for the iterative PageRank computation with 10 percent data changed, i^2MapReduce improves the run time of re-computation on plain MapReduce.

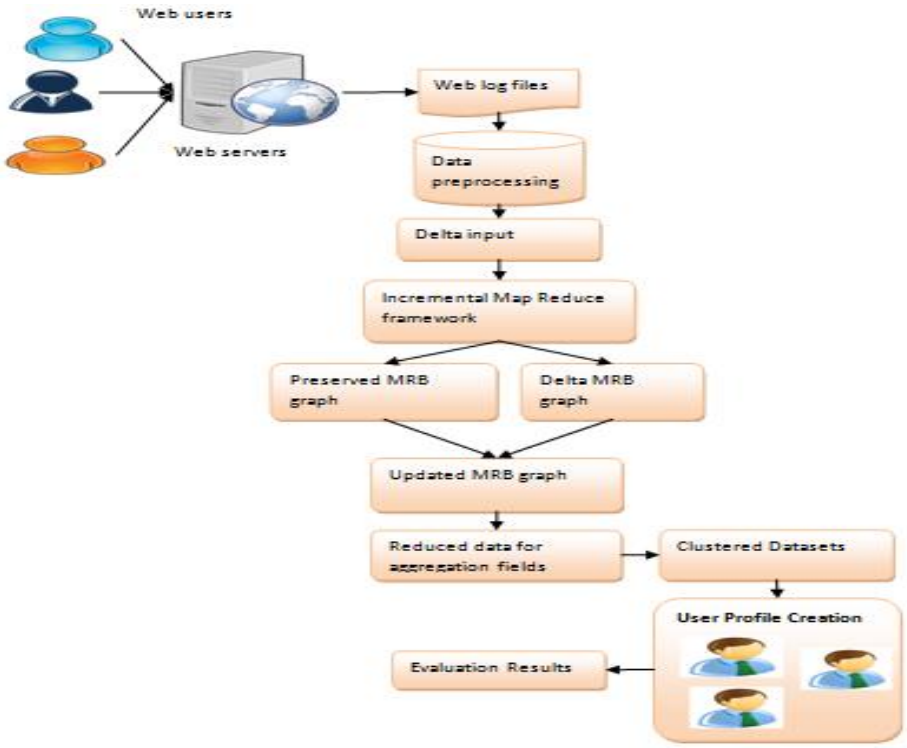

Figure: 1.2 workflow execution

Figure 1.2 defines the overall workflow of I^2 MapReduce. The following steps will demonstrate the working of I^2 MapReduce. . In the first iterative, delta input is produce delta structure data. The preserved MRB Graph reproduce the last iteration in job Ai−1. Only the Map and Reduce example that are precious by the delta input are re-computed. The output of the major Reduce is the delta state data. Apart from the computation, i^2MapReduce revive the MRB Graph with the newly calculate intermediate states. We denote the state as updated MRB Graph. In the j-th iteration, the structure data ruins the same as in the (j − 1)-th iteration, but the loop-variant state data has been updated. Using the preserved MRB Graph j−1, i^2MapReduce recomputed only the Map and Reduce instances that are affected by the input change. 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

IV. COMPONENTS

The various components used here are:(A) Collecting Data block, (B) Iterative Computation, (C)

Fine-grain incremental processing , (D) i²MapReduce Re-computation. The working of each of the components is elaborated in the following section:

A. Collecting Data block

Huge amount of digital data is being accumulated in e-commerce, social network, finance, health care, education environment. It has become increasingly popular to mine such big data in order to gain insights to help business decisions or to provide better personalized, higher quality services. Given the size of the input big data, it is often very expensive to rerun the entire computation from scratch.

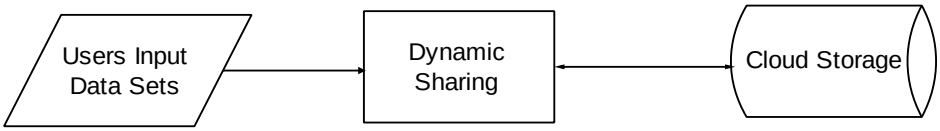

Figure 1.3 Data Block Collection

Divides a file into equal-sized blocks and stores the blocks across a cluster of machines. The Map Reduce system runs a Job Tracker process on a master node to monitor the job progress, and a set of Task Tracker processes on worker nodes to perform the actual Map and Reduce tasks.

B. Iterative Computation

Incremental iterative processing is substantially more challenging than incremental one-step processing because even a small number of updates may propagate to affect a large portion of intermediate states after a number of iterations. To reuse the converged state from the previous computation and employ a change propagation control (CPC) mechanism. Also enhance the MRBG-Store to better support the access patterns incremental iterative processing.

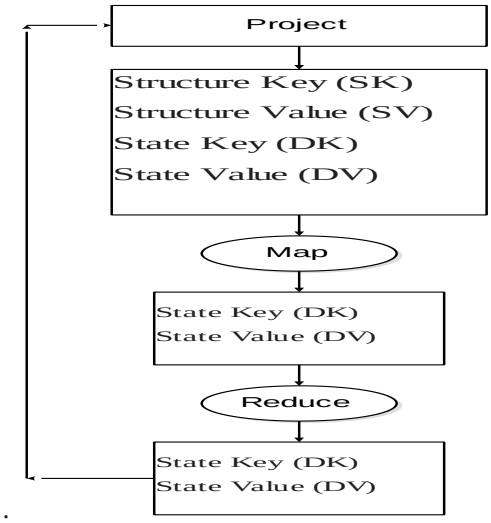

Figure 1.4 Iterative Computation

i²MapReduce is the first Map Reduce-based solution that efficiently supports incremental iterative computation . While users need to slightly modify their algorithms in order to take full advantage of i²MapReduce, such modification is modest compared to the effort to re-implement algorithms on a completely different programming paradigm.

C. Fine-grain Incremental Processing

i²MapReduce supports kv-pair level fine-grain incremental processing in order to minimize the amount of re-computation as much as possible. The kv-pair level data flow and data dependence in a MapReduce computation as a bipartite graph. A MRBG-Store is designed to preserve the fine-grain states in the MRBGraph and support efficient queries to retrieve fine-grain states for incremental processing.

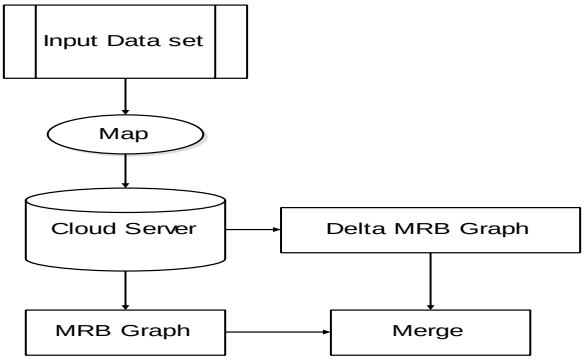

Figure 1.5 Fine-grain Incremental Processing

The Map function takes a kv-pair (K1; V1) as input and computes zero or more intermediate kv-pairs (K2; V2). Then all (K2; V2) is grouped by K2. The Reduce function takes a K2 and a list of V 2 as input and computes the final output kv-pairs (K3; V3). Every record corresponds to a vertex in the graph. K1 is vertex id i, V1 contains "j1:wi;j1 ; j2:wi;j2 ; ..." where j is a destination vertex and wi;j is the weight of the out-edge. Given such a record, the Map function outputs intermediate kv-pair hj;wi;ji for every j. The shuffling phase groups the edge weights by the destination vertex. Then the Reduce function computes for a vertex j the sum of all its in-edge weights as Pi wi;j.

D. i²Map Reduce Re-computation

i²Map Reduce expects delta input data that contains the newly inserted, deleted, or modified k v-pairs as the input to incremental processing. The engine merges the delta MRB Graph and the preserved MRB Graph to obtain the updated MRB Graph using the algorithm. For each datasets the engine deletes the corresponding saved edge state. For each Vertex, the engine first checks duplicates, and inserts the new edge if no duplicate exists, or else updates the old edge if duplicate exists it uniquely identifies a MRB Graph edge. Since an update in the Map input is represented as a deletion and an insertion, any modification to the intermediate edge state consists of a deletion followed by an insertion. For each affected K2, the merged list of V2 will be used as input to invoke the reduce function to generate the updated final results.

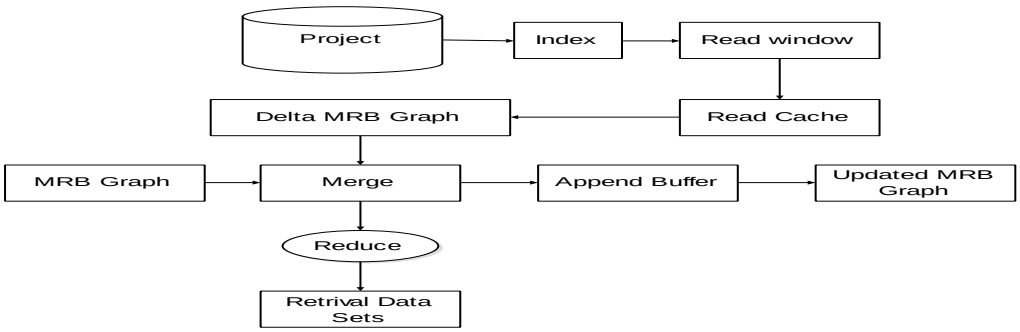

Figure 1.6 i²MapReduce Re-computation

i²MapReduce re-computes the reduce instance associated with each changed MRB Graph edge. For a changed edge, it queries the MRBG Store to retrieve the preserved states of the in-edges of the associated K2, and merge the preserved states with the newly computed edge changes.

V. CONCLUSION

In this paper, proposed a technique to observe the users and collect the information of users on the website and then provide the semantic data to the request of visitors. It will work better than the cookies and beacons etc. There is no need to enter user name and password every time. E-commerce analyzer will remember password and user name and also personalization, information, location memory and site understanding. The E-commerce analyzer optimize logs just track what you need to know about visitors without complex filtering. It will also reduce the size of log file. E-commerce analyzer will be used to collect the information across different domains and websites. The percentage of the total number of visitors who make a purchase on the site Log analyzer will enable them to better understand and respond to the interests of visitors to their sites. E-commerce analyzer will allow e-commerce sites to recognize visitor's generated form online and email advertising campaign.

I2MapReduce combines a fine-grain incremental engine, a general-purpose iterative model, and a set of effective techniques for incremental iterative computation. Real-machine experiments show that i²MapReduce can significantly reduce the run time for refreshing big data mining results compared to re-computation on both plain and iterative MapReduce .In Future, evaluate MapReduce computation using a one-step algorithm and within four iterative algorithms with diverse computation characteristics. The experimental applications results show the significant performance improvements in MapReduce by performing re-computation compared to i²MapReduce.

References

[1] Bhatotia P., Wieder A., Rodrigues R., Acar U. A., and Pasquin R. (2011),'Incoop: Mapreduce for incremental computations', *in Proc. 2nd ACM Symp. Cloud Comput.*, , pp. 7:1–7:14.

[2] Bu Y., Howe B., M. Balazinska M., and Ernst M. D. (2010), 'Haloop: Efficient iterative data processing on large clusters',*in Proc. VLDB Endowment, vol. 3,no. 1–2,* pp. 285–296.

[3] Dean J. and Ghemawat S. (2004), 'Mapreduce: Simplified data processing on large clusters',*in Proc. 6th Conf. Symp. Opear. Syst. Des. Implementation,* p. 10.

[4] Ekanayake J., Zhang H. Li, B., Gunarathne T., Bae S.-H., Qiu J., and Fox G. (2010), 'Twister: A runtime for iterative mapreduce', *in Proc. 19th ACM Symp. High Performance Distributed Comput.,* pp. 810–818.

[5] Jeorg T., Parvizi R., Yong H., and Dessloch S. (2011,),'Incremental recomputations in mapreduce," *in Proc. 3rd Int. Workshop Cloud Data Manage.,* pp. 7–14.

[6] Yan C., Yang X.,Yu Z., Li M., and Li X.(2012), 'IncMR: Incremental data processing based on mapreduce', *in Proc. IEEE 5th Int. Conf. Cloud Comput.,* pp. 534–541.

[7] Zhang Y., Gao Q., Gao L., and Wang C.(2012), 'imapreduce: A distributed computing framework for iterative computation', *J. Grid Comput., vol. 10, no. 1*, pp. 47–68.