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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

Wrapper Approach for Range-Aggregate Queries in Big Data Applications

Siddheshwar D. Kanthekar, Prof. Ismail Mohammed

Department of Computer Engineering, Alard College of Engineering and Management, Pune, India

ABSTRACT: aggregate function on all tuples within given query ranges. Existing approaches to range-aggregate queries are insufficient to quickly provide accurate results in big data environments. In this paper, we propose FastRAQ a fast approach to range-aggregate queries in big data environments. FastRAQ approach on the Linux platform, and evaluate its performance with about 10 billion data records. Experimental results demonstrate that FastRAQ can solve the 1: n format Range aggregate queries problem, i.e., there is one aggregation column and n index columns in a record. We plan to investigate how our solution can be extended to the case of m : n format problem, i.e., there are m aggregation columns and n index columns in a same record. In this range aggregation, insertions of dynamic environments. With the geometric aggregation queries expressions.

KEYWORDS: big data, range-aggregate query, multidimensional histogram, balanced partition.

I. INTRODUCTION

Today Big data is the most demanded topic. The world is moving faster and the phrase becomes true 'World becomes a Village'. Every individual human wants to access network for staying connected with the world. These users may access a lot of data related to Geographical areas, political issues, neural network, health information and many more. There is another thing related to Big data is social sites and media. Social sites like Google for Gmail and most preferably for the search engine, Facebook, whatsApp are hit every day by billions of people around the world. These sites improve knowledge of human social networking, mathematicians, physicians and many more science fields by exchange of information in very small amount of time [1]. All these people search valuable information in just one click [1].

Most of query optimization algorithms are used graphs to analyze and operate effectively. The pattern matching algorithm is part of graph analysis. Distributed and live data can handle with this algorithm. The main importance of pattern matching algorithm is finding the patterns that are related to the outgoing or incoming data. Most time the DAG are used for query optimization. DAG is directed acyclic graph which does not have any cycle means better way a tree, so finding data will not end in Deadlock fashion. The pattern matching algorithm is mostly known to detect the attacks and prevent the attack, but here we are using it for finding the related queries. In existing system the approximate answering approach that acquires accurate estimations quickly for range-aggregate queries in big data environments. Which divide big data into independent partitions with a balanced partitioning algorithm, and then generates a local estimation sketch for each partition. When a range-aggregate query request arrives, RAQ obtains the result directly by summarizing local estimates from all partitions. This existing work used hive database for performance which is not that much fine with semi-structure dataset. So here we want to discover the count records, patterns and much more preprocessing has done before enter into the hive database. The proposed work is extended the existing system which has fast RAQ approach in big data environment. This system work with hive database which is not that much configure with semi-structure dataset. Which traverse the file record from 1 to n each iteration and also preprocessing is require on dataset. In contributed work we use MongoDB database with hadoop new version for better performance than existing system. In this approach the semi-structure data supported queries and configuration is available with the MongoDB database [6].

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

II. LITERATURE SURVEY

Feng Li [1] proposed a Map-Reduce Framework for supporting real-time OLAP system. The open source distributed key/value system; they called it as HBase and Stramed Map-Reduce as Streaming for incremental updating. They proposed an R-store for Map-Reduce support on Real-time OLAP. They evaluate their performance results on the basis of TPC-H data. Jiewen Huang [2] and colleagues introduce query optimization techniques based on distributed graph pattern matching. They proposed two Frameworks that are from the linear and bushy plan is considered in System-R style dynamic programming algorithm and cycle detection algorithm for reduce intermediate result size. The computations reuse technique for eliminating redundant sub queries and traffic reduction. Characterization of point pattern matching is done by the local descriptor called Line Graph spectral context. This work is done by Jun Tang [3] and his associates by doing an analysis of spectral methods and aiming to introduce a robust
for positional jitter and outlier. Multiview spectral embedded technique is used for finding the similarities between descriptor by comparing their low dimensional embedding.
Kosaku Kimura [4] and fellows aimed to reduce the cost of data transmission between components that are processing nodes and interconnection service. Multi-query unification technique generates unified components for DFD. Unification methods are used nesting, clause assembly for collecting the queries and assemble into a single query for reduction of execution time. Results are calculated on the virtual DFD by applying two-stage unification on DSP using Esper and CDP using Hive. Better performance is of DSP using Esper. For Big data analytics, i.e. high-level dataflow system an extensible and language independent framework m2r2 is described in Vasiliki Kalavri [5]. This prototype implementation is done on the Pig dataflow system and results handled automatically in catching, common sub query matching not only rewriting but also garbage collection. Evaluation is done using the TPC-H benchmark for pig and report reduction in query execution time by 65% on average. Xiaochun Yun [6] proposed FastRAQ- big data query execution in a range-aggregate queries approach. A balanced partition algorithm is used first to divide big data into independent partitions, then local estimation sketch generated for each partition. FastRAQ gave result by summarizing local estimation from all partitions. The Linux platform is helpful for implementing FastRAQ and performance evaluated on billions of data records. According to the authors, FastRAQ can give good starting points for real-time big data. It solves the 1: n format range-aggregate query problem, but m:n formatted problem still outside there.
In analyzing the data patterns plays an important role in that each incoming data is analyzed. For a set of patterns for a set of objects in order to determine all possible matches method used is Rete Match Algorithm [7]. It maintains state information of objects which are matched and partially match until the object is present in the memory. There is another pattern matching algorithm also like exact pattern matching which usage searching of related patterns in giving text. Knuth-Morris-Pratt is another algorithm which is also on searching for patterns using Java techniques. RE pattern matching and grep algorithms are on regular expressions and they give more than one result for related pattern. High performance computing (HPC) experienced explosive
growth of data in recent days. Saba Sehrish [8] introducing MRAP (MapReduce with access patterns) techniques for demonstration of results with good percentage of throughput. MapReduce tool can be used for data analysis and reorganizing the HPC storage semantic and data-intensive systems. Running multiple MapReduce phase cause more overhead so authors provide data-centric scheduler to improve performance of MapReduce on HPC. This paper propose a sensor-integrated radio frequency identification (RFID) data repository-implementation model using MongoDB, the most popular big data-savvy document-oriented database system now. First, devise a data repository schema that can effectively integrate and store the heterogeneous IoT data sources, such as RFID, sensor, and GPS, by extending the event data types in electronic product code information services standard, a de facto standard for the information exchange services for RFID-based traceability. Second, we propose an effective shard key to maximize query speed and uniform data distribution over data servers.

III. SYSTEM ARCHITECTURE

A. Proposed Work.

In proposed work this paper extended the existing system as range-aggregate queries in big data environment. Existing system used with the hive database which not suitable or comfortable with semi-structure dataset. To work with semi structure dataset with hive database data cleaning, preprocessing and need to convert in structure form and

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

then apply aggregate function on structure database. So we overcome that problem in the proposed work using semi structure dataset supported MongoDB database [6][7]. Which does not required converting in structured dataset. The semi-structure dataset such as xml file. MongoDB store a semi-structure data in the form of tree structure and execute queries on this tree structure dataset. MongoDB can execute queries on these tree structure data. Here need not to open xml file again and traverse data records one by one or 1 to n records available in xml file. Proficient question taking care of on Big information utilizing Balance segment calculation and Exact example coordinating calculation. This paper is performs segment on the information sets that are coming online or progressively. The segment information is put away into a database as indicated by bunch and applying file system on regarded information. The example coordinating system utilized for coordinating question prerequisites with the gotten information. Related information is gotten and serve to the client. Another contribution in this paper is we provide data anonymization in this system which give privacy to the sensitive data which is need to hide from specific user. In previous system does not work on privacy techniques so the sensitive data was display to user. So here we overcome this privacy using data anonymization terminology. Here we used this techniques only display the result to the end user or vary with valid user. Preserve the privacy issue using data anonymization in this system.

B. Architecture

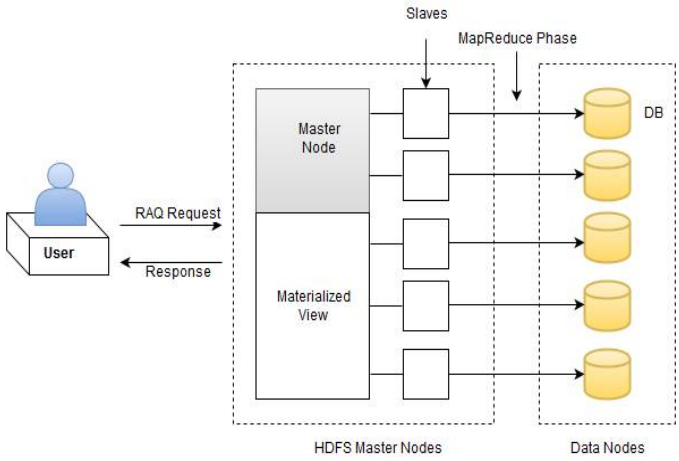


Figure 2 : System Architecture Parallel Processing

System architecture consists of Master node, Materialized view, Slaves. In figure 2 system architecture no. of files or dataset is dividing into different group with regard to their attribute values of interest. No of available server is managing the different group to process the data. User fire a range-aggregate queries to the system then master node apply that operation on different group concurrently. Each group can process the request on different group make a local result or map a data on local dataset. Each slave process data at different data nodes. Result set is revert to the materialized view if result set is accurate respect to the query then it is directly display to the user. If the data is not fine as per quires such as redundancy, abstracts result, more specialization is require so we apply same query on gathered data which is located in materialized view and gathered data consist of the local result generated from the all different group with available server. Such refine data display to the user. We consider a data privacy issue here with the display of data to the different accessibility of users. So we provide data anonymization privacy technique when data display to the user. It maintain the privacy of user personal information, sensitive data etc. suppose for example patient age attribute is display to the respective user so it display as range like 20-25 age. This is one the data anonymization technique. As per type of data apply different techniques to display data at user side to maintain privacy.

Proposed Algorithm

Algorithm 1

Input: Query Q with complex params, Dataset D with different partitions.

Q : select sum(Agg column) other Col name where li1<ColNamei<li2 opr lj1<ColNamej<lj2.

Output: S: range-aggregate query result.

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

Step 1: Generate Q from end user.
Step 2: Submitted to job manager first, then calculate the cardinality of range of columns with col1,col2 using local histogram approach.
Step 3: Generate the mapper as <M1,M2…..Mn> base on data nodes.
Step 4: Assign same Q to each M.
Step 5: Execute the each on each partition of D.
Step 6: Start the processing parallel on each D.
Step 7: Generate the materialize view as MV from each mapper.
Step 8: apply reducer on MV and store in S.
Step 9: return S.

Algorithm 2
Definition 1: (Attribute detachment and Columns)
In quality partition, D(database) comprises of a few subsets, such that every ascribe has a place with precisely one subset. Every subset of characteristics is known as a segment. In particular, let there be C sections C1; C2; . . . Cc, then U(c)i=1,C=D; and for any $1 \leq i1 \neq i2 \leq c$, $Ci \cap Ci2 = \Theta$. For effortlessness of examination, we consider stand out touchy characteristic S. In the event that the information contain numerous touchy traits, one can either think of them as independently or consider their joint circulation. Precisely one of the c sections contains S. Without loss of all inclusive statement, let the section that contains S be the last segment C. This segment is likewise called the touchy section. Every single other segment {C1, C2… … Cc-1} contain just QI qualities.

Definition 2: (Tuple Partition and Buckets).
In tuple allotment, T comprises of a few subsets, such that each tuple fits in with precisely one subset. This tuples subset is known as a container. In particular, let there be b cans. B1,B2… … Bb then $U^{b}{}_{i}{}_{-1}B_i=T$ and for any $1 \leq i1 \neq i2 \leq b$, $Bi1 \cap Bi2 = \Theta$

Definition 3 (Slicing):
determined a small scale information table T, a cutting of T is given by a trait screen and a tuple segment. For instance, assume tables an and b are two cut tables. In Table a, the trait segment is {{Age}, {Gender}, {Zip code}, {Disease}} and the tuple board is {{t1; t2; t3; t4}, {t5; t6; t7; t8}}. In Table b, the property board is {{Age, Gender}, {Zip code, Disease}} and the tuple separator is {{t1; t2; t3; t4}, {t5; t6; t7; t8}}.

IV. RESULTS AND DISCUSSION

The system is built on hadoop 1.0 version and MongoDB database. For dataset we use semi-structure xml data which consist of several attribute. Currently we compare windows RAQ system with Linux platform. The proposed system gives the estimated results and we compare with different existing systems. On the theoretical results shows the satisfactory level accuracy. Here table 1 shows the proposed accuracy as well time complexity how it is better than existing approaches.

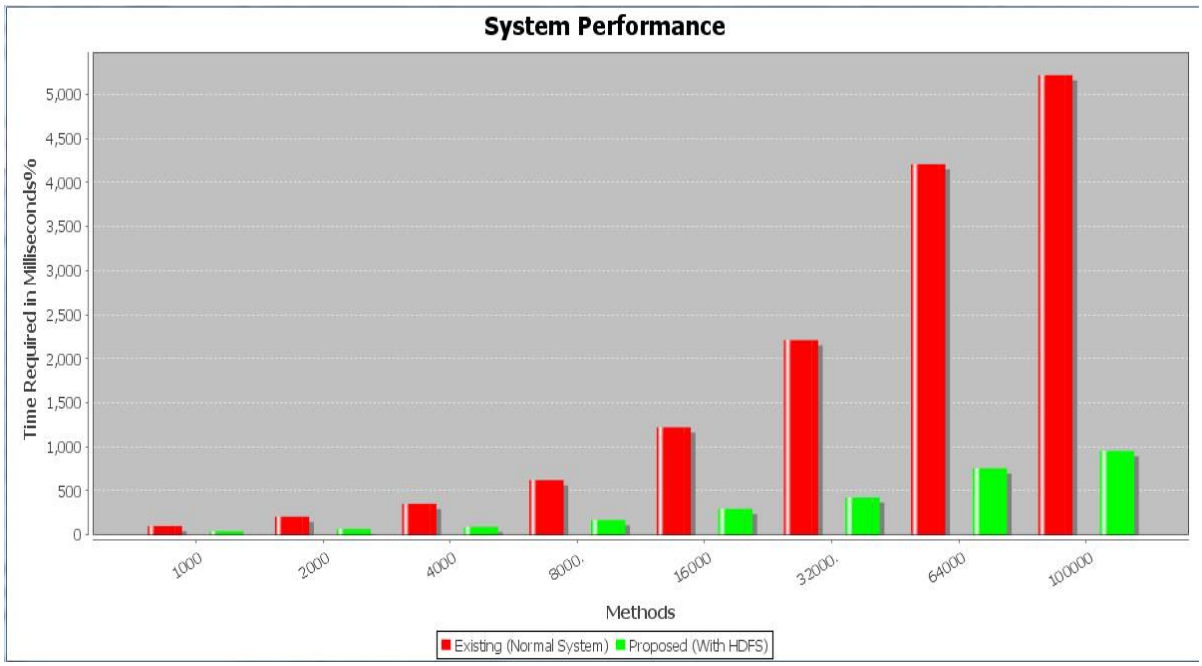
Below graph show the system performance of proposed system vs existing system retrieval time for total number of records.

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016



This undertaking is actualizing the allotment calculation on PIG innovation with help of HADOOP stage what not writing computer programs is finished with the assistance of the Java dialect. After that venture need execution of calculations for preparing of total questions and applying a calculation on them. The venture has named hub i.e. expert hub and information hubs (slave Node). In this venture as a matter of first importance the segment of the information set is done which is uncontrolled as the Hadoop framework does not control information. So by utilization of the adjusted segment calculation, information is controlled and pieces are made in the to begin with phase of yield. At that point for mapping the information and using the time histogram is created. This is the second yield of the undertaking, which gives an advantage while coordinating contain with the client inquiry. Primary reason for the venture is taking care of information effectively for the total capacities which are terminated on one or more section on the huge information.

V. CONCLUSION AND FUTURE WORK

The present work is a wrapper approach of the existing system we plan to wrap different technology into single system to overcome the existing system limitation. Existing system work on files and directory in proposed system we consider a content of file. We also process the content of semi-structure data files. That purpose we form tree data structure using of MongoDB database support. MongoDB easily work with unstructured and semi-structured data. Most of the data is available in this form of data. So we implemented this system with MongoDB database which efficiently work with semi-structure data.

REFERENCES

[1] Wei Tan, M. Brian Blake and Iman Saleh, Schahram Dustdar, "Social-Network-Sourced Big Data Analytics," IEEE Internet Computing, September/October 2013.
[2] Xindong Wu, Xingquan Zhu, Gong-Qing Wu and Wei Ringing "Data Mining with Large Data," IEEE Trans on Information and Data Engineering, Vol. 26, No. 1, January 2014.
[3] G. Mishne, J. Dalton, Z. Li, A. Sharma, and J. Lin, "Fast data in the era of big data: Twitter's real-time related query suggestion architecture," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2013, pp. 1147–1158.
[4] K. S. Beyer, V. Ercegovac, R. Gemulla, A. Balmin, M. Y. Eltabakh, C.C. Kanne, F. Ozcan, and E. J. Shekita, "Jaql: A scripting language for large scale semi-structured data analysis," PVLDB, vol. 4, no. 12, pp. 12721283, 2011.

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

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

[5] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, "Haloop: efficient iterative data processing on large clusters," Proc. VLDB Endow., vol. 3, no. 1-2, pp. 285296, Sep. 2010.
[6] Kang, Y. Nano Information Technology Academy, Dongguk University, Seoul, Korea, Park, I. ; Rhee, J. ; Lee, Y., "MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data", Sensors Journal, IEEE , 10.1109/JSEN.2015.2483499, Sept 2015.
[7] Wenbin Jiang ; Services Comput. Technol. & Syst. Lab., Huazhong Univ. of Sci. & Technol., Wuhan, China ; Lei Zhang ; Weizhong Qiang ; Hai Jin, "MyStore: A High Available Distributed Storage System for Unstructured Data", IEEE 10.1109/HPCC.2012.39, June 2012
[8] Feng Li, M. Tamer Ozsu, Gang Chen and Beng Chin Ooi,"R-Store: A Scalable Distributed System for Supporting Real-time Analytics," IEEE ICDE Conference 2014.
[9] Jiwen Huang, KartikVenkatraman, Daniel J. Abadi,"Query Optimization of Distributed Pattern Matching," IEEE ICDE Conference, 2014.
[10] Hasan M. Jamil, "Mapping Abstract Queries to Big Data Web Resources for On-the-fly Data Integration and Information Retrieval," IEEE ICDE Workshops 2014.
[11] Xiaochun Yun, Guangjun Wu, Guangyan Zhang, Keqin Li, and Shupeng Wang, " FastRAQ: A Fast Approach to Range-Aggregate Queries in Big Data Environments," IEEE Transactions On Cloud Computing, Vol. 6, No. 1, January 2014.
[12] R. Sharathkumar and P. Gupta, "Range-aggregate proximity queries," IIIT Hyderabad, Telangana 500032, India, Tech. Rep. IIIT/ TR/2007/80, 2007.
[13] N. Pansare, V. Borkar, C. Jermaine, and T. Condie, "Online aggregation for large MapReduce jobs," Proc. VLDB Endowment, vol. 4, no. 11, pp. 1135–1145, 2011.