International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-5, August 2015
ISSN: 2395-3470
www.ijseas.com

Reduce Traffic of Data on Network with Bloom filter

Ritu Jain[1], Mukesh Rawat[2]
Department of Computer Science & Engineering
Meerut Institute of Engineering &Technology, Meerut[1;2]
Email: jainritu60[1], mrawatmiet1976[2]@gmail.com

Abstract—In recent time, Data is exponentially increasing so there is problem in quick search. Time, Storage & Processing also main issues for big data so optimization can be done by implementing bloom filter in it. Hadoop is a bunch of technology & have capacity to store large amount of data on Data nodes. So various operations is required on data set. Bloom filter technique is probabilistic data model for getting data into array so that no need to travel all data in network. With implementation of this filter mapper can reduce the amount of data travel. This paper giving the concept to reduce traffic of data on network with efficient manner through probabilistic model of Bloom filter.

Keywords: Big Data, Hadoop, MapReduce, Bloom filter

1. INTRODUCTION

Today, there is demand of quick and accurate result. So need to store data in proper manner with power of easily retrieval. With explosion of data in recent scenario only traditional database is not enough to handle it. With high rate of changing data on web applications there is need of database which can perform to provide consistency as well as partition tolerance. Present database system work with vertical enhancement that give scale-in facility for system. That is not enough for huge database like Linkedin, facebook, Amazon etc. That huge amount of data need to have horizontal enhancement that give scale-out property. By this enhancement any number of node can be added with system.

For Big data there is use of MapReduce [1] programming model that perform operation on single large file so that there is no need to split data.
Companies like facebook, twitter, linkedin etc start using hadoop Hadoop Ecosystem include MapReduce, Apache Hive , PigLatin, Sqoop, flume, zookeeper and HBase . HBase is column oriented database. Its structure consist column family in which different columns are defined with unique row id.
In this paper, part1 describe the Introduction of the basics of MapReduce and Bloom filter.Part 2 gives the basic information of Hadoop components and detail description of HBase. Part 3 shows the proposed implementation on HBase with Bloom filter. Finally part 4 concludes the whole paper.

1.1 BASICS OF MAPREDUCE

MapReduce is a programming model for data processing. Hadoop can run MapReduce programs written in various languages. MapReduce works by breaking the processing into two phases: the map phase and the reduce phase. Each phase has key-value pairs as input and output, the types of which may be chosen by the programmer. The programmer also specifies two functions: the map function and the reduce function. MapReduce is framework that works in distributed environment with server and client infrastructure. Map' process generate intermediate result that need to be process further for resultant, 'reduce' phase start working preceded by shuffle and sort function. If there are 'P' no. of servers in cluster then shuffle phase has traffic $O(P^2)$flows [2].

International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-5, August 2015
ISSN: 2395-3470
www.ijseas.com

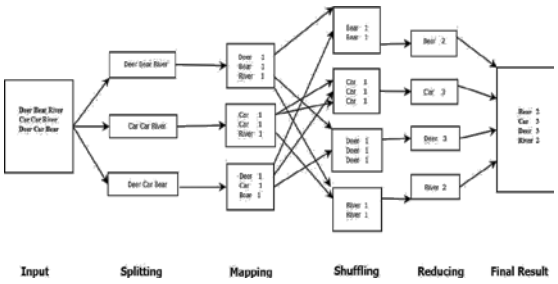

Fig. 1: Example of MapReduce

MapReduce consist several phases as shown in fig. 1 splitting, mapping, shuffling after that reducing. Map phase makes key value pair with default key value is page offset and passes these to reduce phase [1]. Reduce phase accept these value pair implement logic on that [1].

1.2 BASICS OF BLOOM FILTER

Bloom filter is probabilistic data structure which is space efficient with little error allowable when there is test performed. It's data structure is developed by Burton H. Bloom in 1970 [3]. Bloom filter stores elements in an array using hash functions. Let say set S =x_1; x_2; :::::; x_n, construct data structure to answer queries about presence of element existence like Is y in S ? This query does not provide direct result, it just provide idea about data's presence. It also deal with some allowable errors with Bloom filter.

1.2.1. CALCULATION OF BLOOM FILTER:

Size of bloom filter decides FP probability so it should be optimal choice for storing data. If it is less than required then FP probability may be increase or if it is more than optimal then searching will not get affected. In array all 0 represent not presence of element and all 1 indicate presence of element. If in searching any element remains 0 then it decide that element is not present in an array.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Hash each item x_j in S dataset k times. If $H_i(x_j) = a$, set B[a] = 1

0	1	0	0	1	0	1	0	0	1	1	1	0	1	1	0

To check if y is in S, check B at H_i (y). All k values must be 1

0	1	0	0	1	0	1	0	0	1	1	1	0	1	1	0

Possible to have a false positive, all k values are 1, but y is not in S.

0	1	0	0	1	0	1	0	0	1	1	1	0	1	1	0

Fig 2: Basic Bloom filter

For step wise derivation let assume there are 'm' bit array consider as bloom filter that consist 'n' elements of 'k' bits. Probability to set any bit of array '1' by hash function is '1/m' then probability not set it to 1 will be '1-1/m', if it is not set 1 by any n member of array then probability '$(1-1/m)^n$' since there are k bits in message then probability will become '$(1-1/m)^{nk}$'.

If element found in array i.e. 1 then it should be '$1- (1-1/m)^{nk}$'. For complete message found in array probability becomes 'f' i.e.

$$f = (1 - (1 - 1/m)^{nk})^k \approx (1- e^{\{-kn/m\}})^k \qquad (1)$$

Since item can also indicate FP in bloom filter so FP mostly depend with size of bloom filter. So that should be optimal as FP probability can obtain the minimum $(1/2)^k$ when optimum value of k is,

$$k = (m/n)\ln 2 \qquad (2)$$

Size of bit array m is can be choosen as [4]

$$m = -(n \ln p)/(\ln 2)^2 \qquad (3)$$

1.2.2 ALGORITHM OF BLOOM FILTER

A) Creation of Bloom filter.

{

BmFilter

(set A, hash functions, integer m) returns filter

filter = allocāte m bits

foreach a^i in A:

foreach hash function h^j:

filter $[h^j(a^i)]=1$

end foreach

end foreach

return filter

g

}

B) Algorithm for adding elements in array of Bloom filter. At time of adding elements there is need to hash that element.

{

for (i=1....k)
do

hash of all object (x) with hash function;
if (array [position] == 0 then)

array [position] = 1;

}

C) Algorithm for checking presence of element.

{

while array[position] == 1 and i <k;
do

x is member of list;

if array[position] == 0 then

x is not member of list;

increment position;
}

2. HADOOP ECOSYSTEM

Hadoop Ecosystem includes MapReduce, Apache Hive, PigLatin, Sqoop, Flume, zookeeper and HBase**.** Hadoop is not only single software infect it is bunches of technology.

In fig 3, Unstructured or structured data flow through flume or sqoop that can be stored in distributed manner. It will process with MapReduce functions, for further data techniques Hive or Piglatin will work. Apache Oozie is scheduler which schedule of task to run. In fig 3, Unstructured or structured data flow through flume or sqoop that can be stored in distributed manner. It will process with

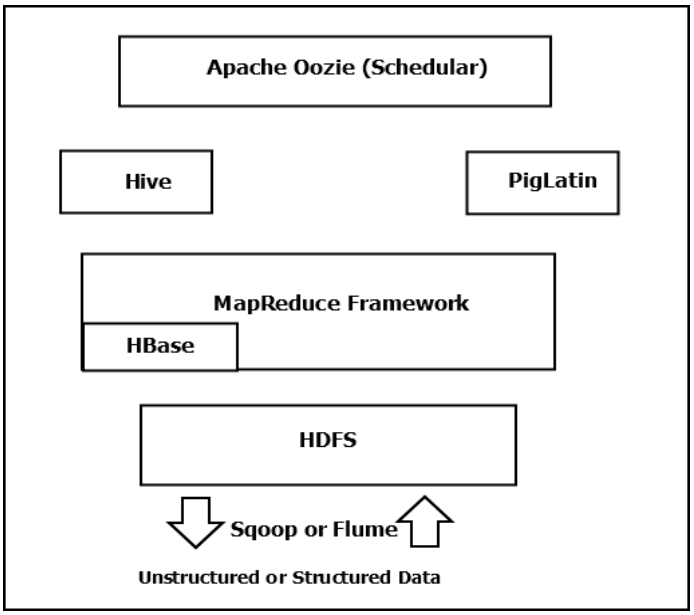

Fig 3: Hadoop Ecosystem

MapReduce functions, for further data techniques Hive or Piglatin will work. Apache Oozie is scheduler which schedule of task to run.

2.1 FUNDAMENTAL OF HBASE

HBase is an open source, non-relational, distributed database modeled after Google's Big Table and written in Java. HBase store data in HDFS and it is random, real time access to database [5]. HBase structure consist columns with column family in which several columns created with unique row-id.

HBase is approached over traditional RDBMS. **Its features are :**

- It is Column-Oriented in comparison to RDBMS.
- It has flexible schema so that column can be added anywhere.
- No need to define blank column as null.
- Tight integration with MapReduce.
- It is good for semi-structured and structured data.

Facebook use HBase for storing database and provide facilities for chat, messaging etc. [6]. Facebook users make interaction with web layer, that store data in separate HBase cluster. HBase it-self use HDFS to store data [5]. HBase Master, HRegion Server and HBase Client are the main component of HBase.

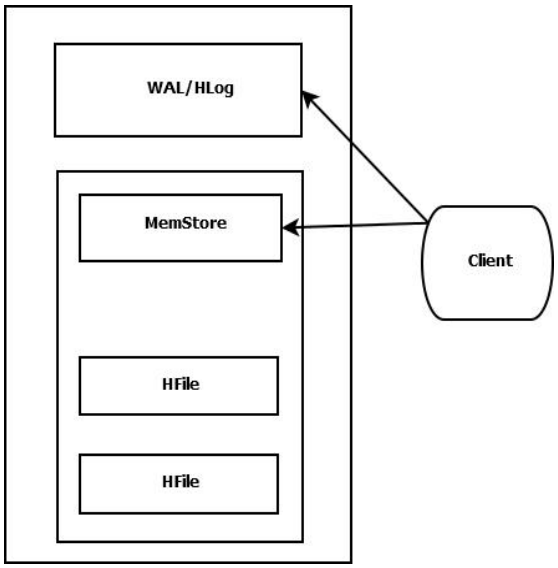

Fig 4: HBase write path

In Fig 4: In HBase data is written in two places
1) WAL/HLog
2) MemStore
MemStore is write buffer, data keep add on in this until memory will be filled. [7]

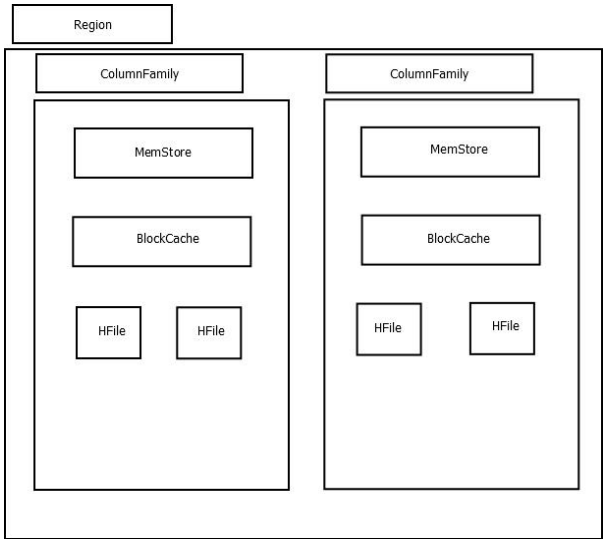

Fig. 5: Column family structure for read

In fig 5, Each column family consists only one MemStore but it may have many HFile. Each HFile size is 64KB in HBase. HBase maintain index of HFiles. A region server may have many region servers as in fig 6. [8] HMaster that decided by zookeeper is responsible to assign the regions to each HRegion server when HBase get started. Zookeeper provides coordination services with HMasters. Many distributed application including HBase [9] using Zookeeper. many real time application such as stormy[10] and twitter storm [11] also using services of zookeeper.

3. PROPOSED ARCHITECTURE

HBase have capacity to store large amount of data. Data can be loaded using pig, sqoop, using HBase shell and client API. HBASE have region servers for hadoop interface. [8]
Code that use to load data to HBase[8]

```
raw data = LOAD 'input.csv' USING
PigStorage(',') AS
(listing_id : chararray, fname : chararray, lname
: chararray);
dump raw data;
STORE raw data INTO
'hbase://sample names USING
org.apache.pig.backend.hadoop.hbase.HBaseStorage
('info:fname info:lname');
```

Data can be loaded through sqoop to HBase.
Code that loading data to HBase by sqoop. [8]
sqoop import
–connect jdbc:mysql:// <ip address><database name >
–username <username _for_mysql _user >
-password<password>
-table<mysql_tablename>
-hbase- table<hbase_target_name>
-column-family<column_family_name>
-hbase-row-key<row_key_column>
-hbase-create-table
Data that store in HBase create HFiles that shifted to datanodes[8] from which datanodes can be treated with same manner with namenode as server.

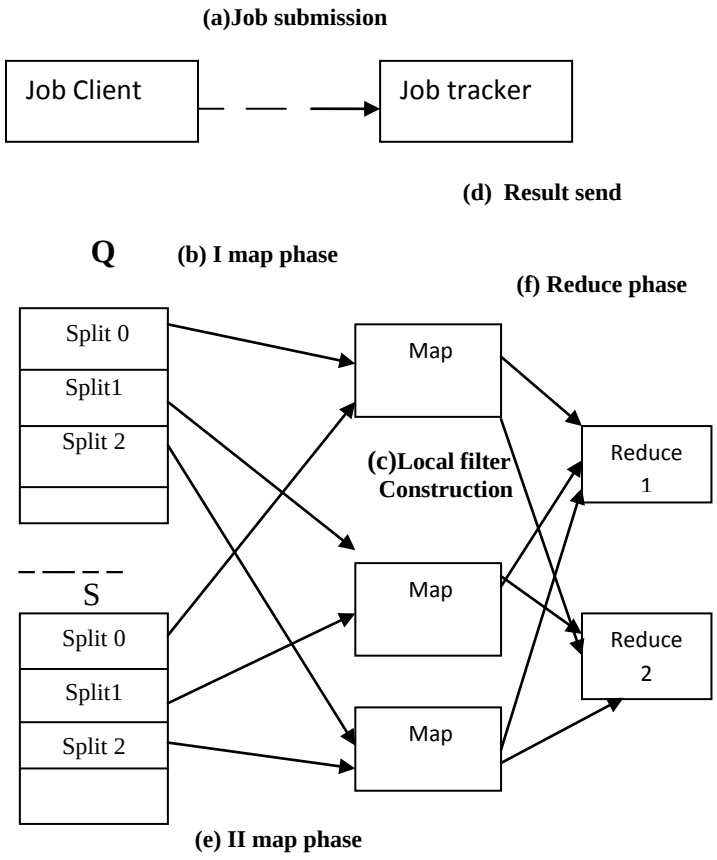

Fig 6: Proposed Implementation of Bloom filter

A. Whole Scenario

As fig 6 showing the proposed implementation of filter for fast accessing data on datanode.

(a)Job Initialization: Job is initialize and submitted to namenode.Where jobtracker run. Namenode contain all information that need to executed in hadoop.job tracker read job files from distributed file system then create map reduce function.

(b) I Map Phase: Jobtracker assign task for tasktrackers. These tasktrackers keep sending signals to prove its aliveness.

(c) Local Filter Creation: bloom filter need to create in each mapper.each mapper create <key, value> pair based on bloom filter that produce intermediate results. Each mapper have its own result after that combined result are send to jobtracker.

(d) Result combination: tasktrackers need to send result to jobtracker.This time tasktrackers will send only filtered records. Filter records having all information but it take less space to provide all.

(e) II Map Phase : This record also perform same work as I map phase.now all that submit to jobtracker where first record already there.now there is perform of join operation on both filters.

(f) Reduce Phase: This phase collect all intermediate records and run reduce function to provide results in output path.

B. Bloom filter construction

Each mapper create bloom filter array which store large amount od data in small size. this will transfer to jobtracker where operation may be implemented. Jobtracker construct global filter to execute operation.

4. CONCLUSION

In this paper, We provide to decrease rate of transfer of data. Using Bloom filter technique that implement on datanodes reduce the size of data travel that need to use in join operations. Datanodes will send array of different data sets that store Bloom filtered data to

jobtrackers. So according to this scenario, operation that need to be done on whole data set, will work with only stored array of Bloom filtered data that is reduce amount in size.

REFERENCES

[1] J. Dean and S. Ghemawat "Mapreduce: Simplified data pro-cessing on large clusters" In Proceedings of the 6th USENIX Symposium on Opearting Systems Design & Implementation (OSDI), pages 137150, 2004

[2] P. Costa, A. Donnelly, A. Rowtron and G. OShea"Camdoop:Exploiting in-network Aggregation for Big Data application." Proceeding USENIX NSDI 2012

[3] B. H. Bloom "Space/time trade-offs in hash coding with allowable errors." Commun. ACMvol. 13, no. 7, pp. 422426, 1970.

[4] Navendu Jain, Mike Dahlin, Renu Tewari "Using Bloom Filters to Refine Web Search Results" In Proc. 7th WebDB page 25-30. (2005)

[5] Konstantin Shvachko, Hairong Kuang, Sanjay Radia,and Robert Chansler "The Hadoop Distributed File System." In Proceedings of the 26th IEEE Symposium on Mass Storage Systems and Technologies (MSST10), Incline Village, Nevada, May 2010.

[6] Tyler Harter, Dhruba Borthakur, Siying Dong, Amitanand Aiyer, Liyin Tang, Andrea C. Arpaci-Dusseau and Remzi H. Arpaci-Dusseau. "Analysis of HDFS under HBase: a facebook messages case study." Proceeding FAST'14 Proceedings of the 12th USENIX conference on File and Storage Technologies Pages 199-212.

[7] Mehul Nalin Vora "Hadoop-HBase for Large-Scale Data."2011 International Conference on Computer Science and Network Technology, Dec. 2011, Pages 601 – 605

[8] hadoop the definitive guide 3rd edition

[9] Apache Hive. Available at http://hive.apache.org

[10] Loesing, Simon, Martin Hentschel,Tim Kraska, and Donald Kossmann."Stormy: an elastic and highly available stream-ing service in the cloud."InProceedings of the 2012 Joint EDBT/ICDT Workshops, pp. 55-60. ACM, 2012.

[11] Marz, N. "A Storm is coming" http://engineering.twitter.com/2011/08/storm-is-comingmore-details-and-plans.html, August2011