OPTIMIZED AND INTEGRATED TECHNIQUE FOR NETWORK LOAD BALANCING WITH NOSQL SYSTEMS

Dileep Kumar M B[1], Suneetha K R[2]

[1]PG Scholar, [2]Associate Professor, Dept of CSE, Bangalore Institute of Technology, Bangalore, Visveshwaraiah Technological University,(India).

ABSTRACT

An integrated technique is proposed for handling social network interactions using Hadoop, which deals with huge computations as well as data analysis and Cassandra database's replication strategy for fault tolerance in distributed Environment , which in turn helps in data analytics. Hypergraph is used as an input to Mapreduce programming model of Hadoop, which reduce the write frequencies across the partitions. Honeypot technique is used as an early-warning and advanced security surveillance tool for data security against the attackers. Finally moving this setup to cloud infrastructure would render good performance.

Keywords: Cassandra File System (CFS), Hadoop (Map Reduce Programming Model), Hypergraph, Honeypot, L7 Load Balancer. Reverse Proxy Server

I. INTRODUCTION

The data around the world is growing day by day. The 80 percent of the world data is been generated from the last 4-5 years. The data from the social networks in the forms of tweets , messages , pictures and videos is the major contribution to this acceleration of the data. This acceleration of growth of data has led to many technologies around the world to manage and analyze in better way. In other words the loads of data generated from social networks has led to Big data analysis , Hadoop [1][2] technology , Data partitioning algorithms, Load balancing schemes and replication strategies for the availability of the data.

Load balancing is a means to distribute tasks over different resources. Load balancing has many kinds namely, Network Load balancing link Load balancing [10] [14] and server Load balancing are among the most common forms. It splits the users across different servers. Load balancer, it follows some etiquettes to accomplish its task. First, it will ensure that the servers are free. Then it contacts the server and if the response is positive, it will adds it in the available list. Load balancer may use the technique called round-robin where the servers are used in queue fashion. Since online social networks gained much more popularity when compared to email , Load balancing plays a vital role in managing social network data.

Earlier we used to work with Relational databases to manage and store the data.Down the lane, there occurred the challenge to manage the unstructured data which is not handled by the RDBMS which was purely schema dependent. This opens up the door for NoSQL [3] systems, which use data partitioning and replication to achieve scalability and availability.

The term "NoSQL" [5] is now getting popularity across the globe. It is targeted towards improvements over the relational databases. The databases categorized under NoSQL system, have variety of good characteristics,but most does not support strict transactions and strict relational model that are essential part of the relational design. The ACID(Atomic-Consistent –Independent _Durable) transactions of the relational model make it virtually impossible to scale across data centers while maintaining high availability, and the fixed schemas defined by the relational model are often inappropriate in today's world of unstructured and rapidly mutating data.

NoSQL (key/value, document, column, graph) brings together a wide variety of technologies under one roof. NoSQL databases can broadly be categorized into four main types, Key-Value databases, Document databases, Column family stores, Graph databases. key/value databases emerged by the inspiration of amazon's dynamo and distributed hash tables and they are designed to handle massive load . HBase (column)[4] is based on google's big table. HBase supports automotive rebalancing or re-partitioning and it is highly distributed. Cassandra taken the features from both dynamo and big table. Cassandra supports fast reads/writes

Query processing in social networks[7] plays vital role in performance of a server. It covers ,server load imbalance, the total number of I/O operations, and the number of servers processing a query.

In this paper, the work proposes a selective partitioning and replication method for data distribution in social networks by utilizing the Hadoop and Cassandra[8]. As a supportive input to this work,a novel Hypergraph[6] model to represent the social network interactions among multiple users and Honeypot [12] [13] technique to avoid attacks to personal data.

II. PROPOSED TECHNIQUE

Hadoop File System (**HDFS**) got emergence in the market due to usage of its commodity hardware thereby providing cost-effective storage for applications and it is mainly used for large scale computations and calculations. While Cassandra File System (**CFS**) has capability to run Analytics on the data it has received and It is fault tolerant in distributed Environment. So merging the capabilities of both HDFS with Cassandra CFS would let us perform both Analysis and Analytics on the data from line-of-business application. Using the **Hypergraph** technique before replicating the data will reduce the Write overhead of the newsfeed. The L7 load balancer called reverse proxy server is used, which makes a load balancing decision based on the content of the message. Integrating **Honeypot** technique with proposed system to avoid attackers to gain access to personal files and videos.

2.1 Algorithm

Input: Social Network User Interactions and Registration Details:

Output : Data processed & replicated with the help of HDFS and CFS [11]. Handles multiuser operations and Chooses less loaded partition for new user registration

2.1.1 Register User to Less Loaded Partition

```
for i=1 to n                    // Number of users.
    User_i ← Reg. details[i]
    P = (p1 + p2 + p3 +…+pn)    // P is a partition has sum of records in each Partition.
    X = P/n
    If (X == p1 && X==p2 &&… X==p3)
```

RandomPartiotion();

Else

For i=1 to Pn //Consider Partition p1,p2,…..pn

X[i] ←Count [records (P_i)] // x – is a array stores count(records) of each partition.

SORT the elements of array – x[ ] // Use Treemap(Data structure) for automatic sorting.

End for.

2.1.2 Algorithm: Data Handling

- Construct **Hypergraph** Users and their interactions with each other.
  HyperGraph H(V,E) ← User Interactions.
- Give Hypergraph as Input to **MapReduce** technique (Hadoop).
  MR ← H(V,E)
- **Reduce** method reads from CFS. **Map** method writes to CFS.
- Integrate **Honeypot** technique to this system so that it avoids attackers to corrupt data.

2.1.3 Alogorithm: Manipulate/Download file

i←User

for j=1 to n //all files of user i (1to n)

if permission($File_j$) == 1 //permission granted from $user_i$

Delete/Edit/Download the $File_j$ j++;

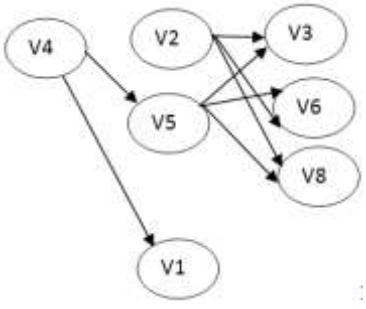
III. APPLICATION OF HYPERGRAPH:

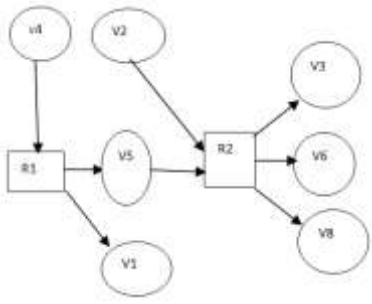
As before stated , Hypergraph[9]is taken as an input to MapReduce Programming model, in turn MapReduce model communicates with Cassandra database CFS. Hypergraph can be analysed by the following example: Directed Hypergraph is used and the edges are called Hyperedges. Here an edge can connect to any number of vertices unlike in graph where it is restricted to only two vertices.

Say R1 and R2 are Replicators. We have ,

R1: V4→V1+V5

R2: V2+V5→ V3+V6+V8





Fig(1)- Graph **Fig(2)- Hypergraph**

As shown in the Fig(1) , a normal Graph Vertex V2 and V5 sends requests/data file to {V3,V6,V8} through point to point link. In Fig(2), the Hypergraph shows usage of Replicator (R1,R2) which takes input from sender and replicates to respective people in that partition, thereby reducing the write frequency.

3.1 Input_Format

Src_Graph_id<tab>Source_vertices<tab>Destination_vertices<tab>Dest.Graph_id

Consider Group1{V2,V4}, Group2{V1,V5}, Group3{V3,V6,V8}

Grp1<tab>V4<tab>V1<tab>Grp2

Grp1<tab>V4<tab>V5<tab>Grp2

Grp1<tab>V2<tab>V3<tab>Grp3

Grp1<tab>V2<tab>V6<tab>Grp3

Grp1<tab>V2<tab>V8<tab>Grp3

Grp2<tab>V5<tab>V3<tab>Grp3

Grp2<tab>V5<tab>V6<tab>Grp3

Grp2<tab>V5<tab>V8<tab>Grp3

3.2 Sample_Output

Source_Group_id<tab>Source_vertives<tab><Replicator_id>

Grp1<tab>V4<tab>R3

Grp1<tab>V2<tab>R2

Grp2<tab>V5<tab>R2

VI. CONCLUSION

The proposed technique uses both the qualities of Hadoop and Cassandra to perform both data analysis and analytics. Hypergraph is been used to handle multi way relations in social networks and given as a input for MapReduce. In turn, MapReduce is made to communicate with Cassadra database called CFS , which is highly fault tolerant in distributed environment. Finally the work suggest to integrate the Honeypot technology to this system to avoid attackers from corrupting the stored data. Moving this setup to cloud will render good performance.

REFERENCES

[1] Shavachko K, Kuang H, Radia S and Chansler R. The Hadoop Distributed File System in Proceedings of the 26th IEEE Symposium on Massive Storage Systems and Technologies, 2010.

[2] Hadoop. http://hadoop.apache.org, 2009

[3] Robin Hecht Stefan Jablonski, University of Bayreuth " NoSQL Evaluation A Use Case Oriented Survey" International Conference on cloud and Service Computing, 2011.

[4] Kellerman, Jim. "HBase: Structured storage of sparse data for Hadoop",13 November 2009.

[5] B. G. Tudorica and C. Bucur, "A comparison between several NoSQL databases with comments and notes", Roedunet International Conference (RoEduNet), 10th, IEEE, (2011) June, pp. 1-5. 2011.

[6] U. Catalyurek and C. Aykanat, "Hypergraph-Partitioning-Based Decomposition for Parallel Sparse-Matrix Vector Multiplication,"IEEE Trans. Parallel and Distributed System,Vol. 10, no. 7, pp. 673-693, http://dx.doi.org/10.1109/71.780863, July 1999

[7] Ata Turk, R. Oguz Selvitopi, Hakan Ferhatosmanoglu, and Cevdet Aykanat. "Temporal Workload aware Replicated Partioning for Social Networks".IEEE Trasactions on Knowledge and Data Engineering, Vol. 26, No 11, November 2014.

[8] M. Y. Becker and P. Sewell. Cassandra: Flexible trust management, applied to electronic health records. In Proceedings of the 17th IEEE Computer Security Foundations Workshop, June 2004.

[9] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. Multilevel hypergraph partitioning: Application in vlsi domain. IEEE Transactions on VLSI Systems, 1998 (to appear). A short version appears in the proceedings of DAC 1997.

[10] S. Aote and M. U. Kharat, "A game-theoretic model for dynamic load balancing in distributed systems", in Proc. The International Conference on Advances in Computing, Communication and Control (ICAC3 '09), New York, USA, pp. 235-238, 2009.

[11] https://wiki.apache.org/cassandra/ArchitectureInternals.

[12] P. Baecher, M. Koetter, M. Dornseif, and F. Freiling. The Nepenthes platform: An efficient approach to collect malware. In Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID), 2006.

[13] Google Hack Honeypot, 2005. http://ghh.sourceforge.net/.

[14] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology(ICWET), February 2011, pages 370-375.

[15] A Platform Computing Whitepaper, 'Enterprise Cloud Computing: Transforming IT', Platform Computing, pp6, viewed 13 March 2010.