"A Survey on Efficient Data Distribution by using Hybrid Data Placement Strategy in MapReduce Framework"

Rutuja J. Deshmukh, Masters of Engineering, Computer Engineering, Pune Institute of Computer Technology, Maharashtra, India

Prof. R.A. Kulkarni, Computer Engineering, Pune Institute of Computer Technology, Maharashtra, India

---***---

Abstract - ***With the recent emergence of services based on the Internet, MapReduce and distributed file systems like HDFS have become as the paradigm for the development of large scale data intensive applications. The data generated by social media is very large in amount and thus requires faster execution thus making the job throughput as a major challenge. Various issues like data distribution, task scheduling and job execution in large scale data processing are needed to be addressed. A novel approach of hybrid data placement strategy and adaptive cluster reconfiguration algorithm for improving data distribution in MapReduce framework is proposed. The approach includes data placement strategies: profiling and pipelining in rack aware data placement technique. The proposed strategy will provide better efficiency for replica placement by improving the response time.***

***Key Words*: MapReduce,, Profiling, Pipelining, Rack Aware Data placement.**

1. INTRODUCTION

Now-a-days peta-bytes of data has become the norm in industries. Handling, analyzing such big data is a challenging task. In recent years an increasing number of data parallel computing frameworks such as MapReduce and Hadoop are used to run data intensive applications and to conduct analysis. Datacenters usually have clusters formed using heterogeneous nodes. Systems like Hadoop can be used to manage such types of clusters but it cannot schedule jobs efficiently on this heterogeneous cluster when data is itself heterogeneous. MapReduce is an programming model for data intensive computing. A considerable percentage of the datacenters run large scale data intensive applications and MapReduce is an paradigm used for building such applications which incorporates mechanisms to ensure the reliability, load balancing, fault tolerance, etc. but such mechanisms can have an impact on energy efficiency as even idle machines remain powered on to ensure data availability.

1.1 HDFS Architecture:

HDFS is a component of Hadoop for file system management which includes hierarchical file organization. HDFS supports write once and read many times policy. It has master-slave architecture, the NameNode as the master and DataNodes are the slaves. There is an secondary NameNode which acts as a backup for NameNode. Basically, the data is divided into blocks of equal size 64MB block, and these blocks are placed on DataNode using default block placement policy that is the random block placement policy. The replication factor by default is 3. Heartbeats are sent to NameNode by the DataNode after regular interval of time to recognize the dead and alive nodes in the system. By default the time interval is of 3 second; if NameNode doesn't get any response from a DataNode in 10 seconds then that DataNode is treated as the dead node. A heartbeat contains node id, track of total storage space, number of data transfers going on at present and amount of storage space in use.

1.2 . MapReduce:

The computation of large volumes of data and the efficient use of computational resources with low execution time are some of the challenging issues. MapReduce framework abstracts the complexity of parallel applications. The MapReduce architecture consists of a master machine that manages other slave nodes. The Map and Reduce phases are handled by the programmer, while the intermediate is called as Shuffle which is created by the system during the job execution. The job is divided into several tasks .A number of slots is a set to indicate how many tasks can be processed simultaneously by each node.

1.3 Block Placement Strategy:

In HDFS, the file is divided into small chunks of equal size and are stored on nodes which helps to achieve fault tolerance. Each copy of block is called as a replica.

The default strategy used by HDFS is rack aware data placement strategy that means if the blocks of data are placed in one rack then their copy will be placed in another rack so as to achieve fault tolerance when there occurs node failure. Following is default block placement policy present in HDFS :

a) Place the first data block(replica) on DataNode which is either local node or any random node depending upon the HDFS client running in the cluster.
b) Place the second data block on a rack other than first block placement.
c) Place the third data block in the rack where the second block is placed.
d) If there are replicas (data blocks) remaining then it distributes them randomly across the racks present in network with the constraint that, in the same rack there are no more than two replicas.

This paper presents analysis on the data placement strategies for more efficient data distribution in the MapReduce framework and also cluster reconfiguration for proper utilization of the nodes in the cluster by the cluster reconfiguration .The rest of paper is organized as follows: Section II describes the related work. Section III describes the problem definition. Section IV describes motivation. Section V includes the proposed work. Section VI includes the mathematical formulation corresponding to proposed work. Section VII concludes the paper.

2. RELATED WORK

- DRAW: A New Data-gRouping-AWare Data Placement Scheme for Data Intensive Applications With Interest Locality:
  The data placement scheme is designed to work at rack level which considers the locality of interest for frequently accessed data blocks i.e. if two blocks are accessed successively then they are considered as related blocks and are grouped together. The random block placement strategy doesn't considers that as data grouping; if it is used then there is less possibility that related blocks are placed on the same DataNode and thus the MapReduce task is applied on multiple DataNodes . The advantage is maximum parallelism is achieved which enhance load balancing. The disadvantage is that there is a very less probability that continuously accessed blocks are needed to be related and the log file which is at NameNode is of huge size so reading it and gaining pattern from that is a major concern.
- A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments:
  A data placement strategy is designed that is different from the Hadoop default strategy. The original strategy in Hadoop assumes that each node has the same capacity in a homogeneous environment, and each node is assigned to the same workload. On the other hand, in a heterogeneous environment, this strategy may reduce the efficiency of Hadoop. The proposed strategy includes that the data are assigned to the nodes according to their execution abilities. Although the data or resources used by a node can be decreased or increased after the data are written into HDFS, the proposed strategy can dynamically adjust and reallocate data.
- MRA++: Scheduling and data placement on MapReduce for heterogeneous environments:
  The proposed method aims to address the main problems originating from the simplification of the MapReduce model while it is being implemented in clusters. The algorithm described allows the use of data-intensive applications in large-scale environments with the use of Internet. There is a short delay in the jobs setup phase that justifies the use of the proposal in heterogeneous environments.
- CoHadoop: Flexible Data Placement and Its Exploitation in Hadoop:
  The proposed method is an lightweight extension for Hadoop called CoHadoop in which a DataNode is selected randomly for every new key. Co-location is achieved by adding a property called locator and a locator table is maintained at master-node and data placement policy is modified so that it uses locator while placing blocks. Many files can be assigned to same locator, the grouping key is used to identify related logs. Co-location is done for all those files which are equivalent to the same key. A file which does not have any locator are placed with default block placement strategy. Log processing is also improved in CoHadoop. The efficiency is improved of many operations such as indexing, columnar storage, joins, grouping, sessionization, and aggregation . The disadvantage is that it leads to imbalance cluster so there is not proper load balancing.

3. PROBLEM DEFINITION

The main aim of the proposed system is to have effective distribution of the blocks of data and to have proper utilization of the nodes based on their computation capacities. The proposed strategy will have proper data distribution of data blocks in the MapReduce environment and there will be energy efficient utilization of the nodes in the cluster.

4. PROPOSED DATA PLACEMENT STRATEGY USING PROFILING AND PIPELINING

4.1 Replica Placement in HDFS using profiling:

When data or the request comes to the NameNode the cluster arranges the DataNodes in the order of there computing capacity. It is the responsibility of NameNode to place the data block replica on other nodes.

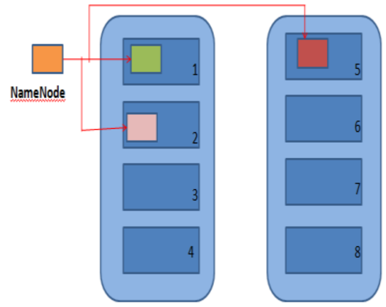

Figure 1: Replica Placement in HDFS using Profiling

4.2 Replica Placement in HDFS using Pipelining:

When data or the request comes to the NameNode the cluster arranges the DataNodes in the order of there computing capacity.It is the responsibility of DataNode to place the data block replica on other nodes.

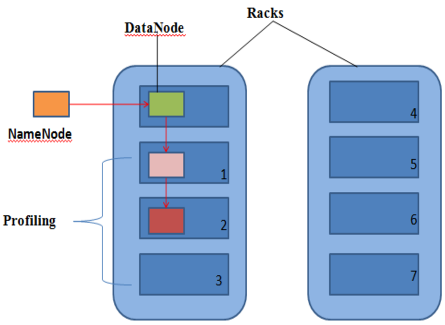

Figure 2: Replica Placement in HDFS using Pipelining

4.3 Default Rack Aware Data Placement:

a. ***On Node-On Rack-Off Rack***: When data or the request comes to the NameNode the cluster arranges the DataNodes in the order of there computing capacity. It is the responsibility of NameNode to place the data block replica on other nodes. The NameNode places the first replica on the node, second on the rack and the third off the rack of better availability.

Rack Failure: If there is sudden rack failure then there may be a huge amount of data loss. So the following two scenarios are taken into account.

b. ***On Node-Off Rack-Off Rack****:* When data or the request comes to the NameNode the cluster arranges the DataNodes in the order of there computing capacity. It is the responsibility of NameNode to place the data block replica on other nodes.The NameNode places the first replica on the node, second off the rack and the third on the rack of better availability.

c. On Node-Off Rack-Off Rack: When data or the request comes to the NameNode the cluster arranges the DataNodes in the order of there computing capacity. It is the responsibility of NameNode to place the data block replica on other nodes. The NameNode places the first replica on the node, second off the rack and the third off the rack of better availability

5. CONCLUSIONS

The hybrid data placement strategy proposed using techniques like pipelining and profiling which will improve efficient data distribution and the cluster reconfiguration strategy applied to the hybrid approach which will improve the energy efficiency and there will be proper utilization of the node making the cluster more energy efficient.

ACKNOWLEDGEMENT

I am indeed thankful to my guide Prof .R. A. Kulkarni for their able guidance and assistance to complete this paper. I am grateful for their valued support and faith on me. I extend my special thanks to Head of Department of Computer Engineering, Dr. G.P. Potdar who extended the preparatory steps of this paper work

.

REFERENCES

[1] Nitesh Maheshwari,Radheshyam Nanduri, Vasudeva Varma,"*Dynamic energy efficient data placement and cluster reconfiguration algorithm for MapReduce framework",2012.*

[2] Xianglong Ye; Mengxing Huang; Donghai Zhu; PengXu, "A Novel Blocks Placement Strategy for Hadoop," *Computer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference*, May 30 2012-June 1 2012

[3] Wang, J.; Xiao, Q.; Yin, J.; Shang, P., "DRAW: A New Data-gRouping-AWare Data Placement Scheme for Data Intensive Applications With Interest Locality," *Magnetics, IEEE Transactions* ,June 2013

[4] Krish, K.R.; Anwar, A.; Butt, A.R., "hatS: A Heterogeneity-Aware Tiered Storage for Hadoop," *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium,* May 2014

[5] Shanjiang Tang; Bu-Sung Lee; Bingsheng He, "DynamicMR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters," *Cloud Computing, IEEE Transactions*,July-Sept. 2014

[6] Nishanth, S.; Radhikaa, B.; Ragavendar, T.J.; Babu, C.; Prabavathy, B., "CoHadoop++: A load balanced data co-location in Hadoop Distributed File System,"

Advanced Computing (ICoAC), 2013 Fifth International Conference , Dec. 2013

[7] Shabeera, T.P.; Madhu Kumar, S.D., "Bandwidth-aware data placement scheme for Hadoop," *Intelligent Computational Systems (RAICS), 2013 IEEE Recent Advances* ,Dec. 2013.

[8] Poonthottam, V.P.; Madhu Kumar, S.D., "A Dynamic Data Placement Scheme for Hadoop Using Real-time Access Patterns," *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference* ,Aug. 2013.

[9] M. Y. Eltabakh, Y. Tian, F. Ozcan, R. Gemulla, A. Krettek, J. McPherson. "CoHadoop: Flexible Data Placement and Its Exploitationin Hadoop," In proceedings of 37th Intern International Conference on Very Large Data Bases, 2011.

[10] Chia-WeiLee, Kuang-YuHsieh, Sun-YuanHsieh, Hung-ChangHsiao, "A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments, 2014.

[11] L. Hu, H. Jin, X. Liao, X. Xiong, H. Liu, Magnet: a novel scheduling policy for power reduction in cluster with virtual machines, in: Cluster Computing, 2008 IEEE International Conference, 2008.

[12] Julio C.S. Anjos a, Ivn Carrera, Wagner Kolberg ,Andre Luis Tibola,Luciana B. Arantes ,Claudio R. Geyer,MRA++: Scheduling and data placement on MapReduce for heterogeneous environments,2015.

[13] Apache Hadoop. http://hadoop.apache.org.

[14] Hadoop distributed file system. http://hadoop.apache.org/hdfs/.