PERFORMANCE EVALUATION OF MULTILEVEL STORAGE OPTIMIZATION USING DIMENSION REDUCTION AND COMPRESSION FRAMEWORK IN DISTRIBUTED CLUSTERS

[1]N. S. KALYAN CHAKRAVARTHY, [2]N. SUDHAKAR, [3]E. SRINIVASA REDDY, [4]D. VENKATA SUBRAMANIAN, [5]P. SHANKAR

[1]Resesarch Scholar, Department of Computer Science, University College of Engineering & Technology, Acharya Nagarjuna University
[2]Professor, Department of Computer Science, Bapatla Engineering College
[3]Principal, University College of Engineering & Technology, Acharya Nagarjuna University
[4]Adjunct Professor, QIS College of Engineering and Technology, Department of CSE, Ongole, A.P
[5]Aarupadai Veedu Institute of Technology, Vinayaka Mission's Research Foundation, Chennai
E-mail: [1]suryaetg@yahoo.com, [2]suds_nagalla@yahoo.com, [3]edara_67@yahoo.com, [4]bostonvenkat@yahoo.com, [5]principal@avit.ac.in

Abstract - The increase in the volume of Big Data and IoT environments are due to different types and complexity of data generated from multiple sources especially from sensors and IoT devices. There are many proven dimension reduction techniques and data compression techniques evolved over last two decades. This paper provides brief overview of some of the critical steps to apply PCA and Erasure Codes in Hadoop Environment and also how Multilevel Storage Optimization is a good approach when compared with other storage compression techniques and dimension reduction techniques. The case study and evaluation proved that the performance of the framework is better in terms of number of blocks, utilization of CPU, Memory, IO and storage efficiency.

Keywords - Evaluation, Performance, MLSO-DRAC, Framework, Storage, Erasure, PCA

I. INTRODUCTION

Replication is one of the mechanism which has been used for more than two decades to provide high availability and reliability of the databases, systems and mission critical data. Hadoop Distributed File System(HDFS) uses three-fold replication technique to provide high availability and fault tolerance capabilities. Though three-fold replication provides greater reliability, it causes extra disk space as there are three copies of the same block made without considering the storage space required. There are many approaches and methods to further reduce the storage space. The recent research works confirms that Erasure Coding technique can be the best alternate to the three-fold replication and the traditional RAID levels.

Dimension reduction techniques are most popular in data warehousing and business intensive applications to reduce the unwanted dimensions to optimize the input data sets. This case study, using the framework, first tries to perform dimension reduction using proven Principal Component Analysis (PCA) technique and then apply the erasure coding to further reduce the storage space there by improve the efficiency in storage management. This section in this paper contains the details about the components and techniques involved in the framework and the performance evaluation of multi-level storage optimization framework.

In recent days, there are many advancements in storage, processing, networking of advances in networking. Especially many applications requires the data files which consumes MBs, GBs and TBs of data [1,2]. Hadoop has become one of the very popular open source platform which has been successfully implemented. It was developed by Dog Cutting and Mike Cafarella in the year 2006, as part of Apache Project and Open Source Framework, mostly written in Java language to allow distributed processing.

Map Reduce framework is built to support complex and huge data sets [3, 4].The HDFS stores the data blocks of the files by using an n-fold technique. The n value can be customized by the Hadoop admins. This type of replication usually leads to performance degrade of the cluster, network as well as wastage of storage space [5].HDFS inherits the design concept of the distributed file systems, which run on any commodity machines. HDFS is aimed to store large data sets in a user-friendly manner. To store such huge data, the files are stored across multiple machines. Hadoop jobs are usually split into smaller tasks to run in parallel. Whenever the workload and data volume increases, it is suitable to move the data sets into HDFS rather than using traditional file servers and database servers.

The primary concept is to use the data block for the job according to the location of the data block. This allow us to minimize the unnecessary data transmission between the machines or hereafter called nodes.

As seen from the figure 1, in HDFS, the data files especially the data blocks corresponding to the files were stored across multiple machines in a redundant fashion, to reduce the data loss. This redundancy feature also supports to run the application jobs in parallel.From the figure it is clear that the first copy

of the data block is written to the data node where the data file is created (affinity). The second copy is stored in another node, on the same storage RACK, whereas the third copy of the data block is stored in a different rack. This helps not only to minimize the cross rock network traffic but also to tolerate any switch failures during the data access and processing.

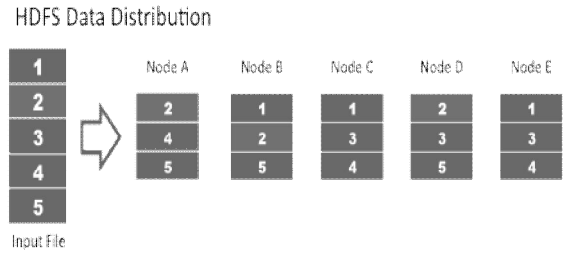


Fig.1. HDFS THREE FOLD DATA DISTRIBUTION

To optimize the storage, Dimension reduction, Encoding or Compression and or suitable techniques should be adopted. The Multilevel Storage Optimization is one of the architecture option which can help to incorporate both Dimension reduction and Compression to manage the storage effectively at two levels. Huge data volume in large sale big data environment is primarily due to the huge abnormal data in different format with variety of data types initiated from various sources [6].The primary objectives of PCA are the following:

- Selection and identification of the variables with high correlation
- Identification of hidden patterns from the variables
- Removal of noise and incomplete information
- Reduction of redundancy and the dimensions

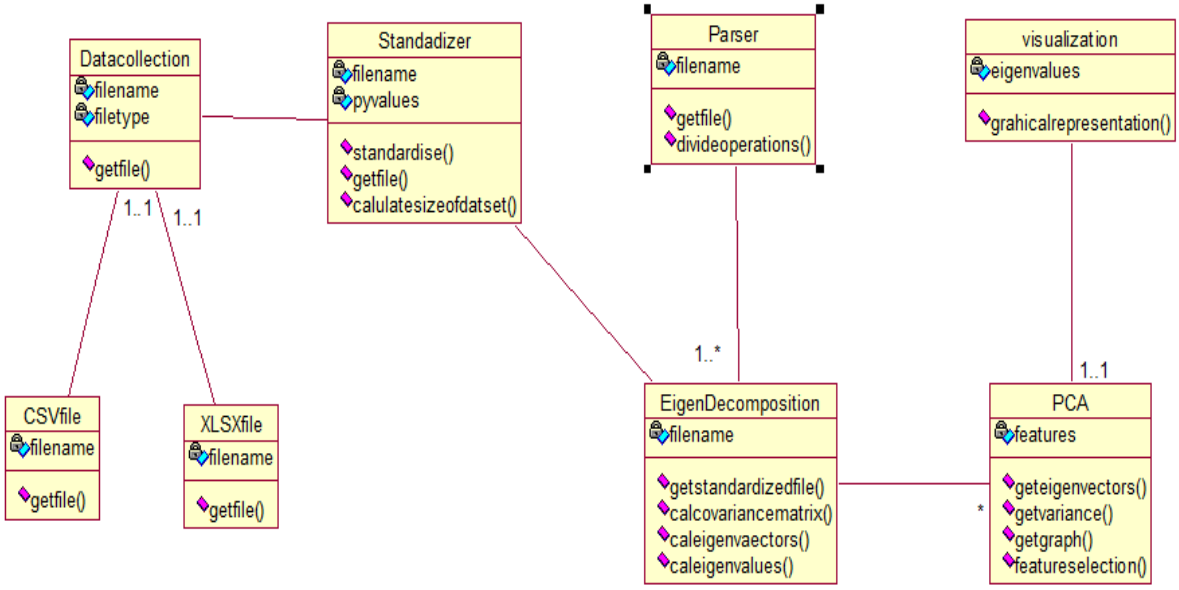


Fig.2. UML CLASS DIAGRAM OF PCA

Fig 2depicts the classes used to construct PCA coding for applying against the data set. The first one is Parse which helps to parse the file, whereas the second component is to standardize the data. The third and fourth components are core components. The last important step is to select the desired features to make a feature vector. The ordered eigenvalues from the higher to a lower value gives us the significant order for the dimensions. The eigenvalues and vectors can be obtained using the corresponding N variables.The Principal Component (PC) of the input data set is based on the highest eigenvalue. The PCs are selected on how many eigenvalues were considered for an effective analysis. To reduce the dimensions, this proposed work considered only the first p eigen values and ignored the rest of the values.

Erasure codes operates on chunks with equal sizes. By applying Erasure code in Hadoop 2.x and 3.0 versions, one can get two chunks namely parity chunk and data chunks.Generation of parity chunks using the data chunks is called encoding. During the failure in any of the disks, the erasure applies decoding to get the actual data using the parity bit using XOR. Erasure Coding follows Reed Solomon approach and the configurations such as 6,3 or 10,4 can be decided based on the needs. In this 6 indicates number of data chunks and 3 indicates parity chunks. Similarly, 10 indicates 10 data disks and 4 indicates 4 parity disks.

II. DETAILS EXPERIMENTAL

2.1. Materials and Procedures

The Hadoop cluster used for the performance evaluation has six physical nodes each running CentOS 6.5 with Hadoop 3.x. Each and every node has Intel i7, 500 GB hard disks, 4 GB Memory, 200 GB hard disk. The first evaluation includes data generated from IoT/Sensors. The second data set has all the internal exam results of the students. All the User Interface components are developed in Java Script (Angular JS) and the PCA code was implemented using Python. The Erasure codes were implemented using the standard Java APIs available in the open source community groups.Table 1 shows the number of different test cases which are conducted for the performance evaluation of different options. Thesetest cases are applied to evaluate overall performance of the cluster. The test cases include the regular Hadoop with 3 fold technique (without RAID), with 0 fold technique (WITH RAID Level 5), with Erasure Code, With PCA, With PCA and Erasure Code. The results are gathered for the storage efficiency using the Erasure code implemented in the cluster. The technology framework which provides all the features are given in figure 3.

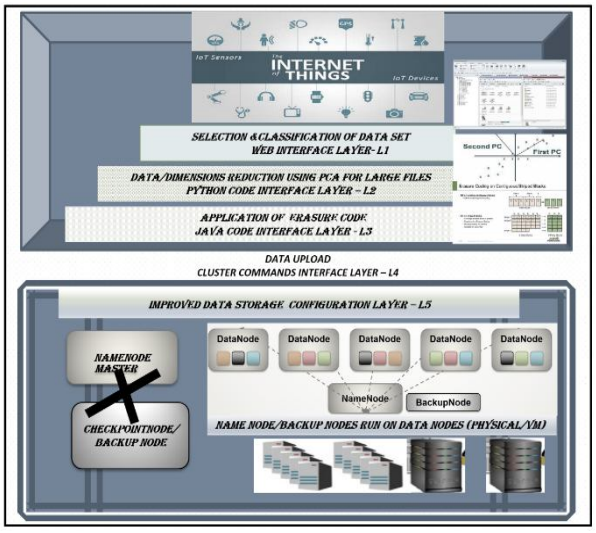


Fig.3. Multi-Level Storage Optimization Framework

In Figure 3, the first layer is implemented in Angular-JS and Java SE environment for selecting the files/data sets. The first level of optimization begin in selecting PCA. The second level of optimization lies on the Erasure Code implementation. The users can select the PCA or Erasure Code depending on their requirements. The third level of optimization is done at the configuration of the environment. As seen in the Figure, the name nodes are created in the data node machines (physical or virtual). By creating the primary name node/master in the data node, the infrastructure admin or cluster admin can reduce the disk space used as well as can witness better utilization of CPU, IO, Network and Memory.

Test #	Number of Name Nodes	Number of Data Nodes	Dimension Reduction	Storage Replication
1A & 1B	2	4	With & Without PCA	3 fold / NO RAID
2A & 2 B	1	4	With & Without PCA	3 fold / NO RAID
3A & 3 B	0	4	With & Without PCA	3 fold / NO RAID
4A & 4B	2	4	With & Without PCA	0 fold / RAID
5A & 5B	1	4	With & Without PCA	0 Fold / RAID
6A & 6 B	0	4	With & Without PCA	0 Fold/ RAID
7A & 7B	2	4	With & Without PCA	Erasure Code
8A & 8B	1	4	With & Without PCA	Erasure Code
9A & 9B	0	4	With & Without PCA	Erasure Code

Table 1: Performance Test Cases

In Table 1, the first column has test case ids, the second column indicates number of physical named nodes created in VM or dedicated machine. The third column contains the count of number of data nodes. The fourth column is meant for applying dimension reduction technique using PCA to perform preprocessing of the data before adding it to the cluster. This is important consideration for this benchmarking to find out how much space is saved before applying and after applying PCA. The last column is the storage replication method opted for each test case. There are three important storage replication methods were considered in this benchmark. They are default Hadoop 3 fold replication, RAID Level 5 and Erasure Code.

Another important consideration in the benchmarking as part of the frame work is the placement of name node as seen in the configuration layer (Layer-5). The tests were conducted by keeping the name nodes in separate machines as well as keeping the name nodes as part of the data nodes. The test cases 3A, 3B, 6A, 6B, 9A and 9B were conducted to keep the name nodes as part of the data nodes. For example, in 3A test case, two name nodes are created in data nodes whereas in 3B test case, one name node is created in the data node.

III. RESULTS AND DISCUSSION

To enhance the compressions, intelligent erasure coding technique is necessary to enhance the poor performance of Hadoop cluster. This also varies

based on the hardware components used to build the cluster as well as the design guidelines followed. In case of RS configurations with storage schema of 6 data disks and 3 parity disks and with 10 data and 4 parity disks gives better storage optimization when compared to 3-fold technique, The RS configuration tolerate even four disk failures with less than 50% overhead in the storage space.

Multiple metrics such as usage of CPU, Memory, IO, Network, File Size and Number of Blocks consumed are collected for different file sizes varying from MBs to GBs on multiple test beds or experiments. Table 4 contains the various measurements collected in the real time run while adding files to the cluster. In the first phase of the work, multiple case studies were conducted to compare the differences between RAID-5 and the traditional 3-fold replication. The initial benchmarking proved that compared to RAID1 and RAID5, Erasure coding works were much better and saved significant amount of disk space. Secondly, the default 3 Fold replication technique consumes more disk blocks and increases the percentage of CPU from both the user and systems' perspective when compared with RAID-5 and Erasure Coding.

File Size	Storage replication	%user CPU	%system CPU	Disk Space Consumed
587.47 MB	0 Fold / Erasure	1.38	17.20	881.20 MB
1.33GB	0 Fold / Erasure	1.35	18.17	1.995 GB
1.4GB	0 Fold / Erasure	1.46	19.01	2.1 GB
587.47 MB	RS/Erasure With PCA	1.36	15.18	801.10 MB
1.33GB	RS/Erasure With PCA	1.31	17.16	1.895 GB
1.4GB	RS/Erasure With PCA	1.47	19.05	1.96 GB
587.47 MB	0 Fold / RAID-5 NO PCA	1.48	19.48	981.20 MB
1.33GB	0 Fold / RAID-5 NO PCA	1.45	18.92	2.01 GB

Table 2: Performance Test Results

The performance test results shows that Erasure with PCA coding reduces the storage space as well as increases the performance of overall cluster in terms of CPU, IO and Memory when compared with RAID-5 and default 3 fold replication. Secondly, by keeping the name node as part of the data node and additional 3 to 4 GB of space can be saved apart from utilizing the Memory and CPU effectively.

The analysis of the simple data set can be done with the help of the PCA tool available in the NumXL package for Microsoft Excel or MATLAB, as well as using the Python code as there are standalone APIs available which can be used to do the coding quickly. Before running the Principal Component Analysis, the data must be preprocessed. All the fields must be of a similar data type hence, the variables such as GENDER and WINTER/SUMMER were labeled accordingly. For example, WINTER is taken as 1 and SUMMER as 0. It is also essential that there are no redundant or invariant variables, as the PCA tool will give an error if a certain column has no variance. It is safe to exclude such variables with constant values as they have no correlation with other variables. Due to these conditions, certain variables have been omitted and finally, a total of 24 variables were considered by the initial preprocessing. The data set has readings for 10 patients. The PCA code has produced Variance, proportion, and cumulative proportion of the 24 principal components. A dimension reduction was carried out to reduce the 24 dimensions to 12 and 9. The reduction to 12 dimensions resulted in a final communality that was above 80% for all the variables. The reduction to 9 dimensions lead to certain variables having a final communality in the range of 60-75% and others above 80%. So apart from the storage space saving by using Erasure Code, name nodes running in data nodes, the PCA further reduces the file size by at least 20%. For example, if the file size is 200GB, then close to 40 GB space can be saved.

CONCLUSIONS

Compared to the existing algorithms and methods, the proposed MLSO-DRC framework supports in increasing the performance of CPU, Memory and Disk IO by 20%. This work focuses on bringing the resource optimization at multi levels. If the users are much concerned about 100% accuracy and completeness of the data, then the PCA option shouldn't be selected. The challenges of the heterogeneous cluster in terms of resource optimization can also be solved by resource optimization policies. So, there exists a need to find predictive and proactive policies and scheduling mechanism which provides fairness to tenants whenever storage is allocated.

ACKNOWLEDGMENTS

The authors wish to express sincere thanks to the management, system engineers and cloud admins of QIS College of Engineering and Technology for

providing high end servers as well as the multiple test data to validate the framework in a short time frame.

REFERENCES

[1] Konstantin Shvachko, HairongKuang, Sanjay Radia, Robert Chansler, The Hadoop Distributed File System, Proceedings of MSST2010, May 2010.

[2] G.D.H.C.Du, Recent Advancements and Future Challenges of Storage Systems, Proceedings of the IEEE, 96(11): 1875-186, 2008.

[3] J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters, Communications of the ACM, Vol 51. (1), pp. 107-113, 2008.

[4] G. Ananthanarayanan, et al., PACMan: coordinated memory caching for parallel jobs, Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012.

[5] S. Sharma, R. Shandilya, S. Patnaik, and A. Mahapatra, A. Leading NoSQL models for handling Big Data: a brief review., International Journal of Business Information Systems, Inderscience, 2015.

[6] L. Pamies-Juarez, A. Datta, F. Oggier, Rapidraid: Pipelined erasure codes for fast data archival in distributed storage systems, in: INFOCOM, 2013 Proceedings IEEE, IEEE, 2013, pp. 1294-1302.

[7] S. Shahrivari, Beyond Batch Processing: Towards Real-Time and Streaming Big Data, Computers. Vol. 3 (4), pp. 117-129, 2014.

[8] Xu, C., Goldstone, R., Liu, Z., Chen, H., Neitzel, B., & Yu, W. (2016). Exploiting Analytics Shipping with Virtualized MapReduce on HPC Backend Storage Servers. IEEE Transactions on Parallel and Distributed Systems, 27(1), 185–196.

[9] G. Dimakis, K. Ramchandran, Y. Wu and C. Suh,A Survey on Network Codes for Distributed Storagel, Proceedings of the IEEE, vol. 99, no. 3, pp. 476--489, March 2011.

★★★