ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

Adaptive Duplication Management in HDFS Based on Supervised Learning

G.SOWNDARYA [1], V.MATHESWARAN [2]

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India
2. Assistant Professor, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

ABSTRACT

The evolution of big data has created a phenomenon in application and solution development to extract, process and store useful information as it emerges to deal with new challenges. In this area, Apache Hadoop is one of the most renowned parallel frameworks. Not only is it used to achieve high availability, Apache Hadoop is also designed to detect and handle the failures as well as maintain the data consistency. Coming along with the development of Apache Hadoop, the Hadoop Distributed File System (HDFS) has been introduced to provide the reliability and high-throughput access for data-centric applications. Gradually, HDFS has become a suitable storage framework for parallel and distributed computing, especially for MapReduce engine, which was originally developed by Google to cope with the indexing problems on big data.

1. INTRODUCTION

The evolution of big data has created a phenomenon in application and solution development to extract, process and store useful information as it emerges to deal with new challenges. In this area, Apache Hadoop is one of the most renowned parallel frameworks. Not only is it used to achieve high availability, Apache Hadoop is also designed to detect and handle the failures as well as maintain the data consistency. Coming along with the development of Apache Hadoop, the Hadoop Distributed File System (HDFS) has been introduced to provide the reliability and high-throughput access for data-centric applications. Gradually, HDFS has become a suitable storage framework for parallel and distributed computing, especially for MapReduce engine, which was originally developed by Google to cope with the indexing problems on big data. To improve the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

reliability, HDFS is initially equipped with a mechanism that uniformly replicates three copies of every data file. This strategy is to maintain the requirements of fault tolerance. Reasonably, keeping at least three copies makes the data more reliable and more robust when tolerating the failures. However, this default replication strategy still remains a critical drawback with regards to the performance aspect. Intuitively, the purpose of inventing Apache Hadoop was to achieve better performance in data manipulation and processing. Therefore, this purpose should be carefully studied at every component. In the performance perspective, based on the well-known research of delay scheduling, if the task is placed closer to the required data source, the system can achieves faster computation and better availability. The metric measures the distance between the task and the corresponding data source can be referred to as the data locality metric. The main reason for the improvement is twofold. First, the network overhead

can be reduced on runtime due to the availability of the local data, and so no inter-communication is needed to transfer the required data from the remote nodes. Second, it is clear that the computation can start immediately on the input data which is locally available, and so no extra task scheduling effort is consumed. Consequently, it is meaningful to say that improving the data locality would immensely enhance the system performance in terms of availability and calculation time.

The main contributions of this research are as follows.

We designed an adaptive replication management (ARM) system to provide high availability for the data in HDFS via enhancing the data locality metric. As a result, the highly local available data improves the performance of the Hadoop system. It is worth noting that the erasure code is applied to maintain the reliability.

We proposed a complexity reduction method for the prediction technique in both hyper-parameter learning and training phases. This proposed method significantly increases the performance in terms of reaction rate for the replication strategy while still keeping the accuracy of the prediction.

We implemented ARM in HDFS and did an evaluation in order to practically verify the effectiveness of the proposed method as compared with the state of the art method.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

2 RELATED WORKS

In the replication area, there are two main methods: the proactive approach and the reactive one. For the proactive approach, the Scarlett solution [4] implements the probability as an observation and then calculates the replication scheme for each data file. The storage budget- limitation is also considered as a factor when distributing the replicas. Although this solution follows a proactive approach instead of using thresholds, the access rate of the data file as well as the suitable placement for replicas is not discussed thoroughly.

Likewise in OPTIMIS, an interesting solution for anticipating the data file status has been proposed. In this approach, the data file is classified and engaged in the limited replication scenarios based on the algorithmic prediction of the demand for data file utilization. [5] proposed a secure hash message authentication code. A secure hash message authentication code to avoid certificate revocation list checking is proposed for vehicular ad hoc networks (VANETs). The group signature scheme is widely used in VANETs for secure communication, the existing systems based on group signature scheme provides verification delay in certificate revocation list checking. In order to overcome this delay this paper uses a Hash message authentication code (HMAC). However, the Fourier series analysis algorithm [6], which is usually used in the field of 'signal processing', is chosen for prediction without a compelling proof of the efficacy. As a consequence, this inappropriate choice may result in poor prediction. For the reactive approach, the cost-effective dynamic replication management (CDRM) method [7] is a costeffective framework for replication in a cloud storage system. When the workload changes, CDRM calculates the popularity of the data file and determines the location in the cloud environment. However, this technique follows a reactive model. As a result, by using threshold values, CDRM cannot adapt well to the rapid evolution of largescale systems.

Similarly, DARE [8] is another reactive model of replication for HDFS. In this model, the authors declare that the probabilistic sampling and competitive aging algorithms are used independently on each node to choose a replication scheme for each data file, as well as to decide the suitable location for each replica. However, there are two issues in this approach. First, the problem of long

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

tasks, which exists in a realistic system, is not considered carefully. In fact, the existence of this issue makes the system unstable. Second, the placement of the replication is judged without considering the file access pattern and system capacity of the destination nodes. For these reasons, DARE might not provide the expected effectiveness on some systems.

The elastic replication management system (ERMS) [9] takes into account an active/standby model for data storage in the HDFS cluster by implementing the complex event processing method to classify the data types. The advantage of ERMS as compared with CDRM and DARE is that it dynamically changes the thresholds for metrics based on the status of the HDFS cluster system. In addition, ERMS is equipped with the erasure code to determine and erase unpopular replicas so as to save the storage. Nevertheless, although CDRM, DARE and ERMS are developed in different ways, all of them encounter the same problems and limitations. Concretely, these solutions try to classify and implement various replicating scenarios for each type of data files by extracting and processing the obsolete information. For that reason, these approaches cannot generate an optimal replication strategy for parallel systems. The detail of this claim is that when some actions are chosen to handle the 'hot' data files, due to high latency and delay, these files may not be 'hot' anymore by the time the actions are engaged. As a consequence, the replicating decision cannot reflect the trends of the data utilization. Additionally, in the ERMS approach, the erasure code configuration is not clearly specified. For that reason, the storage-reliability of this approach is still not verified.

Discussions on erasure code are interesting. Commonly, it is accepted that not only the performance, but also the reliability is the mandatory aspect of HDFS. To fulfill this requirement, the replication and the erasure code are two types of fault tolerance techniques trying to obtain the same goal. While the replication is suitable for enhancing read operation, it suffers from a large storage overhead of up to 200% [10]. Even with the rapid decline in the cost for the storage facility, this overhead has still become the major problem; this is because the volume and velocity of Big Data dramatically increase at a rate faster than the infrastructure, and are required not only for the storage resources but also for the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

computation and network utilization [11]. As a result, many corporations including Facebook, Microsoft and Google think of the erasure coding approach as an alternative technique for saving the repository storage space while keeping the same level of reliability. In addition, the erasure coding approach has a long history of development in peer-to-peer systems to ensure the optimal fault tolerance with a low cost of storage [12]. The latest result of erasure coding is the applications to the Microsoft Azure Storage [13] as well as the new version of the Google File System along with some modules of HDFS.

2.1 EXISTING SYSTEM

In the replication area, there are two main methods: the proactive approach and the reactive one. For the proactive approach, the Scarlett solution implements the probability as an observation and then calculates the replication scheme for each data file. The storage budget-limitation is also considered as a factor when distributing the replicas. Although this solution follows a proactive approach instead of using thresholds, the access rate of the data file aswell as the suitable placement for replicas is not discussed thoroughly. For the reactive approach, the cost-effective dynamic replication management (CDRM) method is a costeffective framework for replication in a cloud storage system. When the workload changes,CDRM calculates the popularity of the data file and determines the location in the cloud environment. However, this technique follows a reactive model. As a result, by using threshold values, CDRM cannot adapt well to the rapid evolution of large scale systems.

2.1.1 Disadvantages of Existing System:

- Low on performance
- Decreasing the performance in terms of reaction rate

2.2 PROPOSED SYSTEM

The system proposes an approach to dynamically replicate the data file based on the predictive analysis. With the help of probability theory, the utilization of each data file can be predicted to create a corresponding replication strategy. Eventually, the popular files can be subsequently replicated according to their own access potentials. For the remaining low potential files, an erasure code is applied to maintain the reliability. Hence, our approach simultaneously improves the availability while keeping the reliability in

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

comparison to the default scheme. Furthermore, the complexity reduction is applied to enhance the effectiveness of the prediction when dealing with Big Data.

Advantages

- To provide high availability for the data in HDFS via enhancing the data locality metric. As a result, the highly local available data improves the performance of the Hadoop system.
- Increases the performance in terms of reaction rate for the replication strategy while still keeping the accuracy of the prediction

3.METHODOLOGY

3.1 MOTIVATION

In many parallel and distributed systems equipped with MapReduce engine, the processing jobs usually comprise a series of consecutive phases, namely map, shuffle and reduce. In the beginning, map phase reads the input from disk and prepares the intermediate data for other phases. Unless the system includes an expensively infinite band of network capacity, which only exists on very large scale computing, the bottleneck between computing nodes is unavoidable. Due to this fact, it would be optimal if the system can colocate map tasks along with the desired data, especially when the data size is large. Unfortunately, MapReduce scheduler is unable to always satisfy this requirement. Replicating uniformly or increasing the replication factor is not the key to accelerate the computation as well as reduce the slot contention and hot-spot issue. Note that the slot contention happens when the number of concurrent tasks accessing the data file surpasses the number of replicas.

Consequently, the tasks with no locally available data have to request for remote access or wait for the next available turns on the same data. Obviously, this issue dramatically decreases the system performance. In the other hand, the hot-spot issue, which is recognized as the attractive nodes to many tasks, makes the system imbalanced and wastes the idle computational capability. These issues must be solved to fulfill the capability of big data system, especially. To minimize the effect of slot contention and hot-spot issue, many approaches choose to improve the data locality and conduct the load balancing as seen in the RelatedWorks section. Nevertheless, as mentioned above, most of these methods are either maladaptive or inaccurate to provide the suitable replication strategies coping with

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

various data access patterns. It is worth noting that beside the growth in storage cost, the diversity of data access patterns is more critical affecting the performance, the replica management and the balance of the system. Thus, this reason motivates us to design a predictive approach (ARM) to truly enhance the data locality with regard to the system utilization and reliability. By proposing ARM, we expect that our study can be useful to any organizations or companies, which are interested in optimizing the performance within an affordable cost.

3.2 SYSTEM DESCRIPTION

Over the past years, the immerse popularity of Internet has produced a significant stimulus to peer-to-peer (P2P) file sharing systems. A recent large-scale characterization of HTTP traffic has shown that more than 75 percent of Internet traffic is generated by P2P applications. The percentage of P2P traffic has increased significantly as files such as videos and audios have become almost pervasive. The study also shows that the access to these files is highly repetitive and skewed towards the most popular ones. Such objects can exhaust the capacity of anode, leading to delayed response. File replication is an effective method to deal with the problem of over load condition due to flash crowds or hot files. It distributes load over replica nodes and improves file query efficiency. File consistency maintenance to maintain the consistency between a file and its replicas is in dispensable to file replication. Requiring that the replica nodes be reliably in for me do fall updates could be prohibitively costly in a large system.

Thus, file replication should proactively reduce unnecessary replicas to minimize the over head of consistency maintenance, which in turn provides guarantee for the fidelity of consistency among file replicas considering file replication dynamism. File replication dynamism represents the condition with frequent replica node generation, deletion, and failures. Despite the significant interdependencies between file replication and consistency maintenance, they have been studied separately. In most current file replication methods, file owners rigidly specify replica nodes and the replica nodes passively accept replicas. The methods were designed without considering the efficiency of subsequent file consistency maintenance.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

These methods make it difficult to adjust the number of replicas to the time-varying utilization of replicas and to ensure that all replicas are fully utilized. The number of replicas has a significant impact on the overhead of file consistency maintenance. Large number of replicas needs more updates hence high consistency maintenance overhead and vice versa. Therefore, the methods lead to high overhead for unnecessary file replications and consistency maintenance. Though these methods generally can be applied to all file replication methods, they cannot be exploited to their full potential without considering time varying and dynamic replica nodes. Structure-based methods assume relatively stable replica nodes, which does not hold true in practice due to dynamic replica nodes caused by file replication. Replicas nodes may be continuously and rapidly generated, deleted, and fail. Such file replication dynamism will lead to unsuccessful update propagation and significantly high overhead for structure maintenance. System-wide message spreading will generate tremendously unnecessary redundant messages. In addition, they cannot guarantee that all replica nodes can receive a update message. Therefore, without taking into account file replication dynamism, consistency maintenance generates unnecessary over head and cannot help to guarantee the fidelity of replica consistency. Furthermore, as in file replication, passively accepting update messages makes it difficult to avoid unnecessary updates in order to reduce overhead.

Uncoordinated deployment of file replication and consistency maintenance techniques can negate each other's efforts and lead to suboptimal or even low system performance. As a result, on one hand, file replication is faced with a challenge to minimize the number of replicas to reduce the consistency maintenance overhead, without compromising its efficiency in hot spot and query latency reduction. On the other hand, consistency maintenance is faced with a challenge to guarantee the fidelity of replica consistency in a timely fashion with low overhead considering file replication dynamism. This makes it important to integrate the two techniques to enhance their mutual interactions and avoid their conflicting behaviors, ensuring that the two techniques can be exploited to their fullest capacities.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

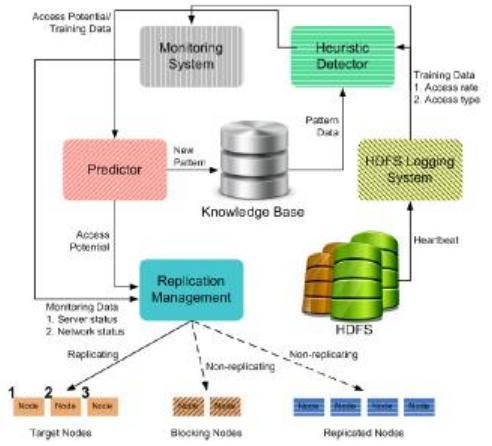

Fig. 1: Architecture of Adaptive Replication Management (ARM) system.

The system presents an Integrated file Replication and consistency Maintenance mechanism (IRM) that achieves high efficiency in file replication and consistency maintenance at a significantly lower cost. IRM integrates file replication and consistency maintenance in a harmonized and coordinated manner. Basically, each node actively decides to create or delete a replica and to poll for update based on file query and update rates in a totally decentralized and autonomous manner. It replicates highly queried files and polls at a high frequency for frequently updated and queried files. IRM avoids unnecessary file replications and updates by dynamically adapting to time varying file query and update rates. It improves replica utilization, file query efficiency, and consistency fidelity. A significant feature of IRM is that it achieves an optimized trade-off between overhead and query efficiency as well as consistency guarantees.

3.3 MODULE DESCRIPTION

3.3.1 MODULE 1

File is sharing into IRM of equal size and k simultaneous connections are used. Client downloads a file from P2P at a time. Each peer sends a replication to the client.

3.3.2 MODULE 2

File is divided into many p2p and user downloads file replication sequentially one at time. The client randomly chooses the source peer at each time slot and download the file replication from each peer in the given time slots.

3.3.3 MODULE 3

Whenever a user completes a replication from its current source peer, the user randomly selects a new source peer and connects to it to retrieve a new p2p. Switching source peers based on chunk can reduce average time varying file download replications and updates.

3.3.4 MODULE 4

File replication is an effective method to deal with the problem of overload

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

condition due to flash crowds or hot files. It distributes load over replica nodes and improves file query efficiency. File consistency maintenance to maintain the consistency between a file and its replicas is indispensable to file replication. Requiring that the replica nodes be reliably informed of all updates could be prohibitively costly in a large system.

4. CONCLUSION

This system proposes an IRM that achieves high efficiency at a significantly lower cost. Instead of passively accepting replicas and updates, nodes autonomously determine the need for file replication and validation based on file query rate and update rate. It guarantees the high utilization of replicas, high query efficiency and fidelity of consistency. Meanwhile, IRM reduces redundant file replicas, consistency maintenance overhead, and unnecessary file updates. Simulation results demonstrate the effectiveness of IRM in comparison with other file replication and consistency maintenance approaches. Its low overhead and high effectiveness are particularly attractive to the deployment of large-scale P2P systems.

The replication management system which is truly adaptive to the characteristic of the data access pattern. The approaches not only pro-actively perform the replication in the predictive manner, but also maintain the reliability by applying the erasure coding approach. It also proposes a complexity reduction method to solve the performance issue of the prediction technique. In fact, this complexity reduction method significantly accelerates the prediction process of the access potential estimation.

4.1 FUTURE WORK

It is found that IRM relying on polling file owners still cannot guarantee that all file requesters receive up-to-date files, although its performance is better than other consistency maintenance algorithms. It has been planned to further study and explore adaptive polling methods to fully exploit file popularity and update rate for efficient and effective replica consistency maintenance.

REFERENCES

[1] "What is apache hadoop?" https://hadoop.apache.org/, accessed: 2015-08-13.

[2] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 17, March 2017

Stoica, "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling," in Proceedings of the 5th European conference on Computer systems. ACM, 2010, pp. 265–278.

[3] K. S. Esmaili, L. Pamies-Juarez, and A. Datta, "The core storage primitive: Cross-object redundancy for efficient data repair & access in erasure coded storage," arXiv preprint arXiv:1302.5192, 2013.

[4] G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg,I. Stoica, D. Harlan, and E. Harris, "Scarlett: Coping with skewed content popularity in mapreduce clusters." in Proceedings of the Sixth Conference on Computer Systems, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 287–300. [Online]. Available: http://doi.acm.org/10.1145/1966445.1966472

[5] Christo Ananth, M.Danya Priyadharshini, "A Secure Hash Message Authentication Code to avoid Certificate Revocation list Checking in Vehicular Adhoc networks", International Journal of Applied Engineering Research (IJAER), Volume 10, Special Issue 2, 2015,(1250-1254)

[6] A. Papoulis, Signal analysis. McGraw-Hill, 1977, vol. 191. [7] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "Cdrm: A cost-effective dynamic replication management scheme for cloud storage cluster." in Cluster Computing (CLUSTER), 2010 IEEE International Conference on, Sept 2010, pp. 188–196.

[8] C. L. Abad, Y. Lu, and R. H. Campbell, "Dare: Adaptive data replication for efficient cluster scheduling." in CLUSTER. IEEE, 2011, pp. 159–168.

[9] Z. Cheng, Z. Luan, Y. Meng, Y. Xu, D. Qian, A. Roy, N. Zhang, and G. Guan, "Erms: An elastic replication management system for hdfs." in Cluster Computing Workshops (CLUSTER WORKSHOPS), 2012 IEEE International Conference on, Sept 2012, pp. 32–40.

[10] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, "Xoring elephants: Novel erasure codes for big data," in Proceedings of the VLDB Endowment, vol. 6, no. 5. VLDB Endowment, 2013, pp. 325–336.

[11] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: a scalable and fault-tolerant network structure for data centers," ACM SIGCOMM Computer Communication Review, vol. 38, no. 4, pp. 75–86, 2008.

[12] A. Duminuco and E. Biersack, "Hierarchical codes: How to make erasure codes attractive for peer-to-peer storage systems," in Peerto-Peer Computing, 2008. P2P'08. Eighth International Conference on. IEEE, 2008, pp. 89–98.

[13] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci et al., "Windows azure storage: a highly available cloud storage service with strong consistency," in Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM, 2011, pp. 143–157.