|| Volume 4 || Issue 3 || March 2019 || ISO 3297:2007 Certified ISSN (Online) 2456-3293

OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

LEVERAGING FILE REPLICATION IN DATA-INTENSIVE CLUSTERS WITH ENERGY ADAPTABILITY

KHAN SHAKEEL KHAN JAMEEL

PG Student, Computer Science & Engineering Department, EESGOI, Aurangabad

---

Abstract: File replication is a common strategy to improve data reliability and availability in large clusters. Reliability for each file under server failures based on the relationship between file reliability and replication factor when servers have a certain probability to fail. In the existing system more number of replica creates it require more energy consumption, time and storage space and some time system fails and cannot send immediate response to the user request. In the proposed system energy efficient adaptive file replication system using bloom filtering to reduce latency time. In the propose system increasing number of file replica according to user request or user priority and vice versa. Propose strategies in reducing file read latency, replication time and power consumption in large cluster. If multiple user send request for one file then system create multiple replica according to priority and get immediate response to the user.

I INTRODUCTION

1. EAFR aims to improve the data availability with the consideration of file popularity and file storage system efficiency.
2. In data intensive clusters, a large amount of files are stored, processed and transferred simultaneously.
3. To increase the data availability, some file systems create and store three replicas for each file in randomly selected servers across different racks.
4. Random selection of replica destinations requires keeping all servers active to ensure data availability, which however wastes power consumption.
5. The random selection of replica destinations does not consider destination bandwidth and request handling capacity, network congestions may occur due to capacity limitation of some links and server may become overloaded by data requests

1.1 MOTIVATION

The motivation of this system is some file systems create and store three replicas for each file in randomly selected servers across different racks. However, they neglect the file heterogeneity and server heterogeneity, which can be leveraged to further enhance data availability and file system efficiency. As files have heterogeneous popularities, a rigid number of three replicas may not provide immediate response to an excessive number of requests, So we propose a dynamic transmission rate adjustment strategy to prevent potential incast congestion when replicating a file to a server, a network aware data node selection strategy to reduce file read latency, and a load-aware replica maintenance strategy to quickly create file replicas under replica node failures. Random selection of replica destinations requires keeping all servers active to ensure data availability, which however wastes power consumption. The random selection of replica destinations does not consider destination bandwidth and request handling capacity, network congestions may occur due to capacity limitation of some links and server may become overloaded by data requests.

1.2 SCOPE

The scope of the system is to provide good load balancing. Memory consumption: the storage usage to store all file replicas (including the original copy) in the system. Maintenance overhead. An updates maintenance overhead is defined as the product of the latency of this update and the update message size. A files maintenance overhead is the sum of the maintenance overheads of the updates on all of its replicas.

Recovery latency: the time span from when the creation of file replicas in a failed server is initiated until all file replicas stored in the failed servers is recovered. A large amount of file are stored, processed and transferred simultaneously. To increase the data availability, some file systems create and store three replicas for each file in randomly selected servers across different racks. However, they neglect the file heterogeneity and server heterogeneity, which can be leveraged to further enhance data availability and file system efficiency. As file have heterogeneous popularities, a rigid number of three replicas may not provide immediate response to an excessive number of read requests to hot file, and waste resources (including energy) for replicas of cold files that have few read requests. Also, servers are heterogeneous in network bandwidth, hardware configuration and capacity (i.e., the maximal number of service requests that can be supported simultaneously), it is crucial to select replica servers to ensure low replication delay and request response delay. In this paper, we propose an Energy-Efficient Adaptive File Replication System (EAFR), which incorporates three components. It is adaptive to time-varying file popularities to achieve a good trade off between data availability and efficiency. Higher popularity of a file leads to more replicas and vice versa. Also, to achieve energy efficiency, servers are classified into hot servers and cold servers with different energy consumption, and cold file are stored in cold servers. EAFR then selects a server with sufficient capacity (including network bandwidth and capacity) to hold a replica. To further improve the performance of EAFR, we propose a dynamic transmission rate adjustment strategy to prevent potential incast congestion when replicating a file to a server, a network aware data node selection strategy to reduce file read latency, and a load-aware replica maintenance strategy to quickly create file replicas under replica node failures. Experimental results on a real-world cluster show the effectiveness of EAFR and proposed strategies in reducing file read latency, replication time, and power consumption in large clusters.

II LITERATURE SURVEY

1. Yuhua Lin and Haiying Shen, "EAFR: An Energy-Efficient Adaptive File Replication System in Data-Intensive Clusters", IEEE Transactions On Parallel And Distributed Systems, Vol. 28, No. 4, April 2017.

In data intensive clusters, a large amount of files are stored, processed and transferred simultaneously. To increase the data availability, some file systems create and store three replicas for each file in randomly selected servers across different racks. However, they neglect the file heterogeneity and server heterogeneity, which can be leveraged to further enhance data availability and file system efficiency. As files have heterogeneous popularities, a rigid number of three replicas may not provide immediate response to an excessive number of read requests to hot files, and waste resources (including energy) for replicas of cold files that have few read requests. Also, servers are heterogeneous in network bandwidth, hardware configuration and capacity (i.e., the maximal number of service requests that can be supported simultaneously), it is crucial to select replica servers to ensure low replication delay and request response delay. In this paper, we propose an Energy-Efficient Adaptive File Replication System (EAFR), which incorporates three components. It is adaptive to time-varying file popularities to achieve a good tradeoff between data availability and efficiency. Higher popularity of a file leads to more replicas and vice versa. Also, to achieve energy efficiency, servers are classified into hot servers and cold servers with different energy consumption, and cold files are stored in cold servers. EAFR then selects a server with sufficient capacity (including network bandwidth and capacity) to hold a replica. To further improve the performance of EAFR, we propose a dynamic transmission rate adjustment strategy to prevent potential incast congestion when replicating a file to a server, a networkaware data node selection strategy to reduce file read latency, and a load-aware replica maintenance strategy to quickly create file replicas under replica node failures. Experimental results on a real-world cluster show the effectiveness of EAFR and proposed strategies in reducing file read latency, replication time, and power consumption in large clusters.

2. M. Rengasamy, P. Anitha, "Energetic Key Exchange Protocol Authentication for Similar Network File Systems", International Journal for Innovative Researchin Science & Technology— Volume 2 — Issue 05 — October 2015.

The key establishment difficulty is the most important issue and we learn the trouble of key organization for secure many to many communications for past several years. The difficulty is enthused by the propagation of huge level dispersed file systems behind parallel admission to manifold storage space plans. Our task focal points on the present Internet ordinary for such file systems that is parallel Network File System [pNFS], which creates employ of Kerberos to set up similar session keys flanked by clients and storage strategy. Our appraisal of the obtainable Kerberos bottom procedure demonstrates that it has a numeral of boundaries: (a) a metadata server make possible key swap over sandwiched between the clients and the storage devices has important workload that put a ceiling on the scalability of the procedure; (b) the procedure does not make available frontward confidentiality; (c) the metadata server produces itself all the assembly keys that are used between the clients and storage devices, and this intrinsically shows the way to key escrow. In this system, we suggest a assortment of authenticated key swap over procedures that are intended to tackle the above problems. We demonstrate that our procedures are competent of plummeting up to roughly 54% of the workload of the metadata server and concomitantly at the bottom of onward confidentiality and escrow freeness. All this necessitates only a minute portion of greater than before calculation in the clouds at the client.

3. Sagar S. Lad, NaveenKumar, Dr. S. D. Joshi, "Comparison study on Hadoop's HDFS with Lustre File System",

International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-8, November 2015.

Map/Reduce is a distributed computational algorithm, which is originally designed by Google, Mapreduce is expanding in popularity and is being utilized for many large-scale jobs. The open-source Hadoop system has the most common implementation of Map/Reduce. For the fundamental storage backend, Hadoop by default manages the Distributed File System (HDFS), however Hadoop originally was planned to be compatible with other FS (file systems). Apart from 0THDFS, Hadoop does provide few other types of FS i.e. KFS, S3.Hadoop uses Java interface provided by these file systems. Lustre doesnt contain JAVA wrapper. Lustre doesnt accept like hadoop does. Lustre provides a POSIX-complaisant interface for UNIX file system. Common problems with Hadoop plus HDFS as a platform can be solved with Lustre as a backend system.

4. Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler, "The Hadoop Distributed File System". The Hadoop Distributed File System (HDFS) is designed to store very large data sets reliably, and to stream those data sets at high bandwidth to user applications. In a large cluster, thousands of servers both host directly attached storage and execute user application tasks. By distributing storage and computation across many servers, the resource can grow with demand while remaining economical at every size. We describe the architecture of HDFS and report on experience using HDFS to manage 25 petabytes of enterprise data at Yahoo!

III SYSTEM DESIGN

In this project, proposed EAFR to reduce file read latency, power consumption and replication completion latency. EAFR adaptively increases the number of replicas for hot files to alleviate intensive file request loads, and thus reduce the file read latency, and also decreases the number of replicas for cold files without compromising their read efficiency.

IV SYSTEM ARCHITECTURE

In large data-intensive clusters, most popular files are generally small in size, while large files seldom get read [5]. Therefore, replicating and migrating popular files is relatively light in storage and bandwidth cost. Taking advantage of this characteristic in clusters, EAFR increases the number of replicas of popular files in order to boost their availability and reduces the number of replicas of cold files in order to save resources. Fig. 2 shows an overview of EAFR. EAFR divides servers into hot servers and cold servers: hot servers consume more power and provides prompt response for file requests; while cold servers stay in sleeping mode with 0 percent CPU utilization and low energy consumption.

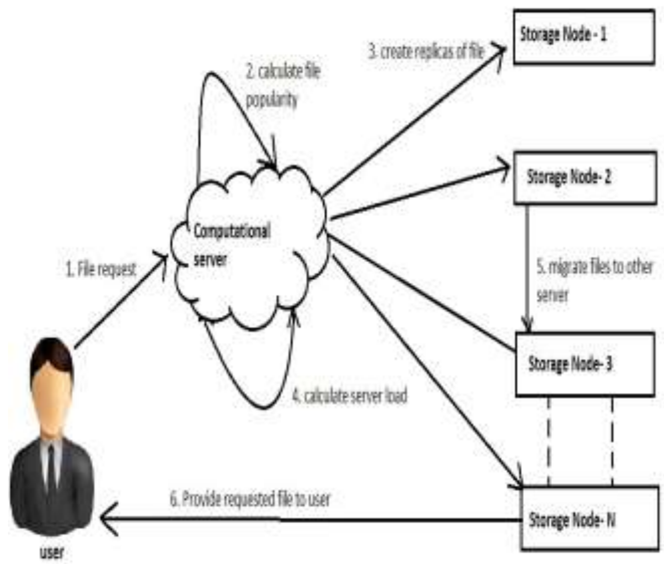

Figure 1: Architecture Diagram

V RESULTS

Figure 2: Cluster 1

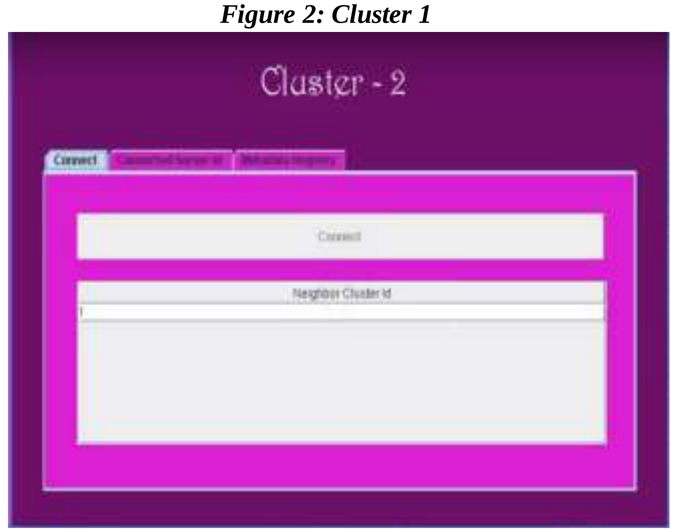

Figure 3: Cluster 2

Figure 4: Server 1

Figure 5: Enter Cluster ID

Figure 6: Transfer Time

VI CONCLUSIONS

The popularity of data-intensive clusters places demands for file systems such as short file read latency and low power consumption. In this system a dynamic transmission rate adjustment strategy to prevent potential incast congestion when replicating a file to a server, a network aware data node selection strategy to reduce filer read latency, and a load-aware replica maintenance strategy to quickly create file replicas under replica node failures. Experimental results from a real-world large cluster show the effectiveness of EAFR and the proposed strategies in meeting the demands of file systems in large clusters.

REFERENCES

[1] L. Mashayekhy, M. Nejad, D. Grosu, Q. Zhang, and W. Shi, Energy-aware scheduling of MapReduce jobs for big data applications, IEEE Trans. Parallel Distrib. Syst. TPDS, vol. 26, no. 10, pp. 27202733, Oct. 2014.

[2] Lustre File System. (2015).

[3] Version 2 parallel virtual file system. (2015).

[4] S. Weil, S. Brandt, E. Miller, D. Long, and C. Maltzahn, Ceph: A scalable, high-performance distributed file system, in Proc. 7th Symp. Operating Syst. Des. Implementation, 2006, pp. 307320.

[5] Z. Cheng, et al., ERMS: An elastic replication management system for HDFS,
in Proc. CLUSTER Workshops, 2012, pp. 3240.

[6] Q. Chen, J. Yao, and Z. Xiao, Libra: Lightweight data skew mitigation in MapReduce, IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 9, pp. 25202533, Sep. 2014.

[7] A. Verma, G. Dasgupta, T. Nayak, P. De, and R. Kothari, Server workload analysis for power minimization using consolidation, in Proc. Conf. USENIX Annu. Tech. Conf., 2009, pp. 2828.

[8] J. Luo, L. Rao, and X. Liu, Temporal load balancing with service delay guarantees for data center energy cost optimization, IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 3, pp. 775784, Mar. 2014.

[9] K. Shvachko, K. Hairong, S. Radia, and R. Chansler, The hadoop distributed file system, in Proc. IEEE 26th Symp. Mass Storage Syst. Technol., 2010, pp.
110.

[10] L. Mashayekhy, M. Nejad, D. Grosu, Q. Zhang, and W. Shi, Energy-aware scheduling of MapReduce jobs for big data applications, IEEE Trans. Parallel Distrib. Syst. TPDS, vol. 26, no. 10, pp. 27202733, Oct. 2014.

[11] Y. Chen, A. Ganapathi, A. Fox, R. Katz, and D. Patterson, Statistical workloads for energy efcient mapreduce, Tech. Rep. UCB/EECS-2010-6, Univ. California, Berkeley, CA, USA, 2010.

[12] M. Chowdhury, S. Kandula, and I. Stoica, Leveraging endpoint exibility in data-intensive clusters, in Proc. ACM SIGCOMM Conf., 2013, pp. 231242.

[13] A. Greenberg, et al., VL2: A scalable and exible data center network, Proc.ACMSIGCOMM Conf. Data Commun. , 2009,pp.5162.

[14] W. Yu, Y.Wang, and X. Que, Design and evaluation of networklevitated merge for hadoop acceleration, IEEE Trans. Parallel Distrib. Syst, vol. 25, no. 3, pp. 602611, Mar. 2014.

[15] P. Bodik, I. Menache, M. Chowdhury, P. Mani, D. Maltz, and I. Stoica, Surviving failures in bandwidth-constrained datacenters, in Proc. SIGCOMM Conf. Appl. Technol. Archit. Protocols Comput. Commun., 2012, pp. 431442.

[16] M. Bourguiba,K.Haddadou,I.El Korbi, and G.Pujolle, Improving network I/O virtualization for cloudcomputing, IEEETrans.ParallelDistrib.Syst,vol.25,no.3,pp.673681,Mar.2014.
[17] J. Zhang, F. Ren, L. Tang, and C. Lin, Modeling and solving TCP incast problem in data center networks, IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 2, pp. 478491, Feb. 2015.
[18] Q.Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster, in Proc. IEEE Int. Conf. Cluster Comput., 2010, pp. 188196.
[19] G. Ananthanarayanan, et al., Scarlett: Coping with skewed content popularity in MapReduce clusters, in Proc. IEEE 7th Int. Conf. Cloud Comput., 2011, pp. 408415.
[20] L. Abad, Y. Lu, and R. Campbell, Dare: Adaptive data replication for efficient cluster scheduling, in Proc. IEEE Int. Conf. Cluster Comput., 2011, pp. 159168.
[21] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, Hedera: Dynamic ow scheduling for data center networks, in Proc. 7th USENIX Conf. Netw. Syst. Des. Implementation, 2010, pp. 1919.
[22] M. Chowdhury, M. Zaharia, J. Ma, M. Jordan, and I. Stoica, Managing data transfers in computer clusters with orchestra, in Proc. ACM SIGCOMM Conf., 2011, pp. 98109.