ERMS: Elastic Replication Management System for HDFS

Zhendong Cheng, Zhongzhi Luan, You Meng, Depei Qian, Alain Roy, Gang Guan

Sino-German Joint Software Institute, Beihang University, China
University of Wisconsin–Madison, USA
Tencent Research, China

INTRODUCTION

The Hadoop Distributed File System (HDFS) has been widely adopted to build cloud storage systems. It provides reliable storage and high throughput access to large-scale data by Map/Reduce parallel applications.

Based on data access patterns, the data in HDFS is classified into three types:

- **Hot data**: the popular data, which means the data receives not only a large number of concurrent accesses, but also a high intensity of access.
- **Cold data**: the unpopular data that is rarely accessible.
- **Normal data**: the rest

Data replication has been widely used as a means of providing high performance, reliability and availability. Triplication policy has been favored in HDFS not only because it can be easily implemented, but also for its high performance, and reliability. However, there have been two problems:

- In a large and busy HDFS cluster, the hot data could be accessed by many distributed clients concurrently. Replicating hot data only on three different nodes is not enough to avoid contention for datanodes storing the hot data.
- The triplication policy comes with a high overhead cost in terms of management for the cold data. Too many replicas may not significantly improve availability, but bring unnecessary expenditure instead. The management cost of cold data, including storage and network bandwidth, will significantly increase with the high number of replica.

In view of these issues, we designed and implemented ERMS, an elastic replica management system for HDFS. ERMS introduces an active/standby storage model, takes advantage of a high-performance complex event processing (CEP) engine to distinguish the real-time data types, and brings in an elastic replication policy for the different types of data. ERMS uses Condor to increase the replication number for hot data in standby nodes, and to remove the extra replicas after the data cooling down. The erasure codes could be used to save storage space and network bandwidth when the data becomes cold data.

Elastic Replication Management System for HDFS

◆ System Architecture

The architecture of ERMS is showed in Fig. 1. It automatically manages the replication number and replica placement strategy in HDFS clusters.

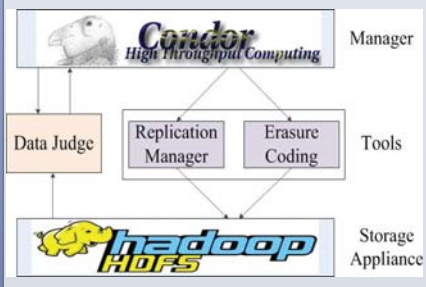

Figure 1: System Architecture of ERMS

- HDFS is the basic storage appliance. The Data Judge Module obtains the system metrics from HDFS clusters and uses CEP to distinguish current data types in real-time.
- According to the different types of data, the manager of ERMS could schedule replication manager tool and erasure coding tool to manage the replicas of data.
- Condor would be an good choice for the manager.

Time window is one of the major features of CEP systems. ERMS makes use of CEP analyzing HDFS audit logs to tell the data types in HDFS. Taking advantage of the time window t_w of CEP, ERMS obtains concurrent accesses number τ within the time t_w and then distinguishes the real-time data types. The data is hot data or cold data if τ is higher than τ_M or lower than τ_m.

◆ Active/Standby Storage Model

ERMS introduces an active/standby storage model. This model classifies the store nodes into two types: active nodes and standby nodes.

We use an active/standby storage model. Standby nodes might be better than active nodes when the active nodes are heavily used. The standby nodes only store the extra replica of hot data.

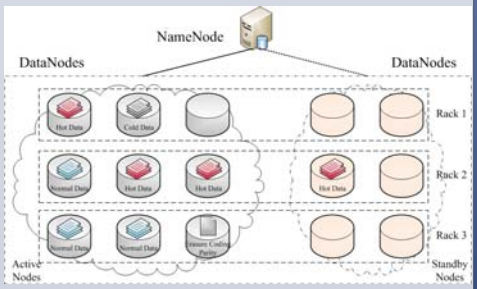

Figure 2: Active/Standby Storage Model

EXPERIMENTAL EVALUATION

We evaluated ERMS in a private cluster with one namenode and fifteen datanodes (ten active nodes and five standby nodes) of commodity computer.

We implemented ERMS in Hadoop-20, which is Facebook's real-time distributed Hadoop, modified the replica placement mechanism and added configuration parameters to suit the ERMS.

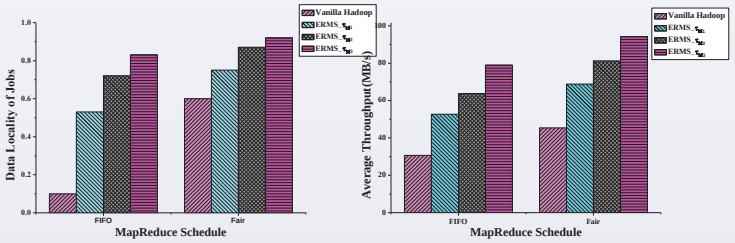

Figure 3: The Performance of ERMS

We run jobs synthesized from the SWIM, which provides one mouth job trace and replay scripts of a Facebook 3000-machine production cluster trace. We evaluate data locality and average reading throughput of these jobs under different thresholds ($\tau_{M1}>\tau_{M2}>\tau_{M3}$). Data locality and reading throughput are two critical metrics for performance of HDFS. Data locality could reduce pressure on the network fabric. The results show that ERMS could effectively improve data locality and reading throughput, as shown in Fig. 3. The threshold τ_M is also an important parameter. It is a tradeoff between system performance and storage cost.

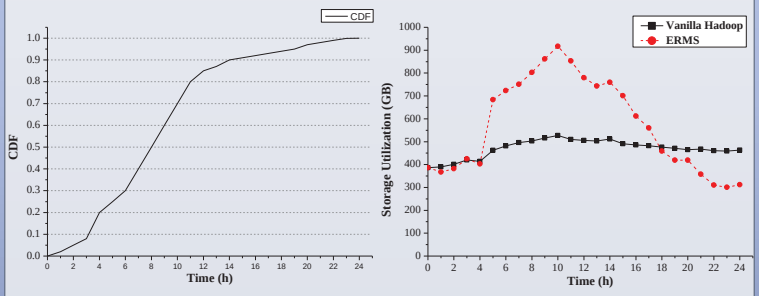

Figure 4: CDF of Data Accessing and Storage utilization

We also experiment with erasure codes. For the cold data, which concurrent accesses number τ is lower than τ_m, we use Reed Solomon codes to encode it, with a replication factor of one and four coding parities. The results show that this erasure codes doesn't heart data reliability and reduce storage overhead.

CONCLUSION

We present the design and implement of ERMS, an elastic replica management system for HDFS that seeks to increase data locality by replicating the hot data while keeping a minimum number of replicas for the cold data. ERMS dynamically adapt to changes in data access patterns and data popularity, and impose a low network overhead. The active/standby storage model and replica placement strategy used by ERMS would enhance the reliability and availability of data.

In the future, we plan to :

- investigate more effective solutions to detect and predict the real-time data types.
- evaluate ERMS in real cloud systems, which are provide by Tencent and HuaWei.

ACKNOWLEDGMENT

This work was partially supported by the National High Technology Research and Development Program ("863"Program) of China under the grant No. 2011AA01A203.

CONTACT

Author: Zhendong Cheng
Email: zhendong.cheng@jsi.buaa.edu.cn
Sino-German Joint Software Institute
BeiHang University (BUAA)
No.37 XueYuan Road, HaiDian District,
Beijing, P.R.China, 100191