Flexible Management of Data Nodes for Hadoop Distributed File System

Wooseok Ryu
Dept. of Healthcare Management
Catholic University of Pusan
Busan, Republic of Korea
e-mail: wsryu@cup.ac.kr

***Abstract*—Hadoop Distributed File System (HDFS) is a file system, which stores big data in a distributed manner. Although HDFS cluster provides a great scalability, it requires numerous dedicated data nodes, which makes it difficult for a small business enterprise to construct a big data system. This paper presents a novel mechanism for flexible management of data nodes in the HDFS cluster. A block replication scheme is also presented to ensure availability of data. Using the proposed scheme, storage capacity of HDFS cluster can be dynamically increased by using existing hardware systems.**

Keywords-Hadoop; HDFS; flexibility; node management.

I. INTRODUCTION

A big data system makes it possible to identify meaningful information by extracting and analyzing massive data created in the enterprise. Currently, Apache Hadoop [1] is one of the most popular open source distributed framework for big data analytics. In the Hadoop, Hadoop Distributed File System (HDFS) is provided to ensure reliability and high availability of data storage under distributed computing environment. The MapReduce framework is also used to provide parallel processing of big data stored in HDFS. Due to its scalability and fault tolerance, Hadoop system can process huge volume of data on a large-scaled cluster with 10,000 processing cores [2].

Among the various applications, health care is a very promising field for big data analytics [3]. Hospitals also demand big data analysis to improve quality of care and to establish management innovation strategy [4]. However, relatively small business domains, such as small-and-medium sized hospitals, have difficulty in adopting a big data system despite the potential for data analysis because of its high cost. As an alternative way, we can consider using existing systems used for daily business to minimize adoption cost for constructing a Hadoop cluster. These systems are normally used for routine work, joining them into the Hadoop cluster should be carefully investigated.

This paper discusses a cluster management technique for flexible management of data nodes in Hadoop. Data nodes in the original Hadoop are dedicated systems for the cluster and cannot be casually added to or removed from the cluster. This paper first analyzes node commission and de-commission mechanism of Hadoop and proposes a flexible management mechanism for the cluster, which allows existing systems to be added to or removed from the cluster dynamically. Using the proposed mechanism, existing systems used for daily work during business hours can be redirected to the Hadoop cluster out of hours, which maximizes system utilization. The rest of this paper is organized as follows. Section II describes the flexible management mechanism for Hadoop. A block replication for ensuring availability of data is presented in Section III. Section IV concludes the paper.

II. FLEXIBLE MANAGEMENT MECHANISM

A. Node Management in HDFS

A HDFS cluster consists of one name node which manages namespace of the file system and a number of data nodes that store user data. When the HDFS starts, a namenode daemon in the name node starts, followed by starting each datanode daemon in the data node. The HDFS can add a new data node to the cluster or remove an old data node from the cluster without stopping all services, named commissioning and decommissioning, respectively.

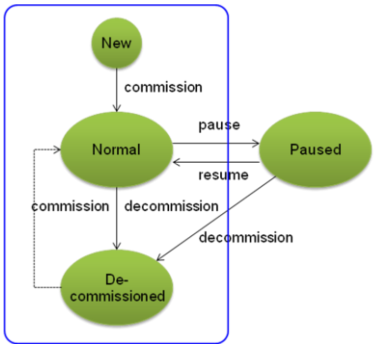

Figure 1. State transition diagram for a data node.

A boxed area of Figure 1 shows a state transition diagram for a data node in the original HDFS. When a new data node is about to join the cluster, a commissioning procedure is called to the node and the state of the node is changed to normal. After then, the node can store data blocks as requested by the name node. A normal node can be removed from the cluster via decommissioning procedure when the node is decrepit or malfunctioning. When the procedure is being executed, all data blocks stored in the node are moved to other normal nodes followed by the removal from the cluster. Note that a dotted line depicted in Figure 1 indicates that a decommissioned node could be commissioned again, however it is not practical because it required additional network overloads for the repeated copy of data blocks.

B. Flexible Data Node Management Mechanism

Providing flexibility for the Hadoop file system implies that nodes in the Hadoop cluster can be temporarily removed from the cluster and can re-join the cluster at any time. It is different from previous commission and decommission mechanisms because a decommissioned node is assumed to be permanently removed from the cluster.

This paper proposes an additional state named *paused* as depicted in Figure 1. The paused node maintains data blocks and can join the cluster again [5]. This means that a paused node has temporarily left from the cluster but the node will be rejoined when the system becomes available for analysis. The paused node can be used to other work, e.g. everyday business. The node rejoins the cluster when the node becomes idle. The pause procedure is as follows.

- A data node requests to name node via ssh which executes a script in the name node. The script adds the node descriptor to *dfs.host.pause* property in *hdfs-site.xml*.
- Refresh data nodes by calling *dfsadmin –refresh Nodes*, which removes data nodes from the cluster.
- Kill a *datanode* daemon running in the data node.

Paused nodes need to be managed in the name node because the paused nodes cannot be considered as target nodes for block reallocation which is executed by *HDFS balancer* daemon. This is achieved by *dfs.host.pause* property, which stores descriptors of paused nodes. The resume procedure for rejoining the cluster is similar to the commissioning procedure. However, data blocks of the resumed node should be checked for consistency since the data might be removed from the cluster while paused.

III. Block Replication Scheme

Hadoop replicates each data block to separated nodes to provide fault tolerance and availability of accessing data. Default value of replication level is 3 [1]. When a block of certain node is unavailable, another copy from another node can be accessed. However, in the flexible management mechanism, another copy can also be inaccessible since any node can be paused at any time. To mitigate this problem, this paper divides distributed nodes into two types of clusters; one is a core cluster and the other is a flexible cluster as shown in Figure 2.

Data nodes in the core cluster is as same as the typical Hadoop cluster, which stores data blocks and used for analytical processes at all times. On the other hand, data nodes in the flexible cluster can be used for both daily business and data storage triggered by pause and resume mechanism discussed in Section 2. In this system, at least one replica should be stored in data nodes in the core cluster to guarantee minimum availability of data. For example, one replica is stored in the core cluster and two replicas are stored in any nodes in the flexible clusters. When all data nodes in the flexible clusters are paused, the core cluster with one replica can be used for Hadoop processing. If all data nodes are normal, extended data storage with improved processing power will be provided.

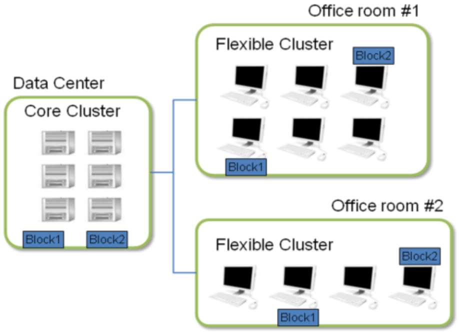

Figure 2. Block replication in the flexible cluster

The number of data nodes in flexible clusters and overall storage capacity are not linearly correlated since it is bounded by replication factor. Assuming each data node has the same storage capacity and the minimum replication factor for core cluster is 1, the maximum utilization can be achieved when the number of data nodes in the flexible cluster becomes twice the number of data nodes in the core cluster.

IV. Conclusion

This paper discussed a node addition and deletion mechanism of the HDFS and proposed a flexible node management mechanism which enables dynamic management of the Hadoop cluster. A block replication scheme is also presented to ensure minimum availability under flexible node clusters. Using the proposed mechanism, scale of the Hadoop cluster can be dynamically changed as the cluster can utilize existing systems, which implies that small business domains can efficiently construct a big data processing system without much cost. As a future work, evaluating the performance of this mechanism needs to be performed on a real business environment.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. NRF-2016R1C1B1012364).

References

[1] Apache Hadoop, http://hadoop.apache.org/, retrieved: Mar, 2017.

[2] C. W. Lee, K. Y. Hsieh, S. Y. Hsieh, and HC Hsiao, “A Dynamic data placement strategy for hadoop in heterogeneous environments,” Big Data Research, 2014, vol. 1, pp.14–22, 2014.

[3] W. Raghupathi and V. Raghupathi, “Big data analytics in healthcare: promise and potential,” Health Information Science and Systems, vol. 2, article 3, pp. 1–10, 2014.

[4] R . Miniati, et al. "Hospital-based expert model for health technology procurement planning in hospitals." Engineering in Medicine and Biology Society, IEEE, 2014, pp. 3504–3507.

[5] W. Ryu, “Dynamic Cluster Management of Hadoop Distributed Filesystem”, Proc. Conference on Korea Information and Communication Engineering, KIICE, Oct. 2016, Vol. 20, No. 2, pp. 435–437.