Forensic Science International: Digital Investigation 32 (2020) 300909

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2020 EU — Proceedings of the Seventh Annual DFRWS Europe

Big Data Forensics: Hadoop 3.2.0 Reconstruction

Edward Harshany[a,*], Ryan Benton[a], David Bourrie[a], William Glisson[b]. [a] *University of South Alabama, School of Computing, Mobile, 36688, USA;* [b] *Sam Houston State University, College of Science and Engineering Technology, Huntsville, 77342, USA*

Keywords
Hadoop
Forensics
Big data
Reconstruction

A B S T R A C T

Conducting digital forensic investigations in a big data distributed file system environment presents significant challenges to an investigator given the high volume of physical data storage space. Presented is an approach from which the Hadoop Distributed File System logical file space is mapped to the physical data location. This approach uses metadata collection and analysis to reconstruct events in a finite time series.

* Corresponding author.

1. Introduction

Metadata management is vital to the Hadoop Distributed File System (HDFS). HDFS is designed to centrally manage all distributed file system metadata through the master server called the Namenode. The metadata details the structure of the distributed file system abstraction through file and directory attributes, mapping of data to data storage locations, and namespace hierarchy (Hadoop — Apache Hadoop 3, 2019). Expeditious assimilation of metadata is critical to successful data evidence recovery for a distributed system with high ingestion rates (Grispos et al., 2013).

HDFS is at the core of the Hadoop ecosystem which has evolved through version releases (White, 2015). Hadoop 3.2.0 version, released in 2018, includes 1) Hadoop Common containing utilities to support Hadoop modules, 2) HDFS, the Hadoop distributed file system handling architecture, 3) MapReduce, for cluster resource managing and data processing, and 4) Yet Another Resource Negotiator (YARN) another resource manager layer forming a data-computation framework (Hadoop — Apache Hadoop 3, 2019). The Hadoop ecosystem is comprised of four main layers employed in varying configurations providing data storage and processing solutions (White, 2015).

This study aims to investigate the effectiveness of utilizing a subset of metadata generated at the HDFS data storage layer to reconstruct file system operations and map data to physical data location. Once mapped, data evidence could be prioritized and targeted for preservation or further analysis.

URL: http://eh1721@jagmail.southalabama.edu

2. Methodology and experimental setting

Methods were confined to the construction of directories and file operations addition and deletion in a specific order. The timeline creates a directory structure within the HDFS namespace, adds files to the HDFS namespace, and deletes specific files from the HDFS namespace. The goal is to reconstruct the sequence of operations over this time period and discover file locations from the HDFS metadata. Fig. 1 shows the setting configured in a fully distributed mode with Hadoop 3.2.0. on available commodity hardware, each running 64-bit Ubuntu 18.0.4.2 operating system with version 4.15 Linux kernel.

Data block ID is used within the logical namespace to identify data blocks within a file. The Datanode local file system uses block ID as the file name to create files in its native file system and store in the Ext4 file's INode structure. Block replicas on Datanodes are represented by two files in the local file system. One contains the data itself and the other records metadata including checksums for the data and the generation stamp (Sremack, 2015).

3. Analysis and findings

HDFS image and edits files were recovered from the live system and converted to.XML files for offline analysis. These files contain serialized image data that must be converted for viewing. The XML files were then read by a parser developed to extract attributes. The most recent discovered image represented the most recent state as all edits were updated on the image. The image contains information on each individual INode including type, name, replication, timestamp information, associated block information, and sequential generation stamp for each data block present. This information can be compared with the INode directory section and a resultant HDFS logical file system namespace can be reconstructed.

The processed image data structures reveal absent inodes, data blocks, and generation stamps indicating file system modification from a previous state. Generation stamps are associated with data blocks during data block

https://doi.org/10.1016/j.fsidi.2020.300909

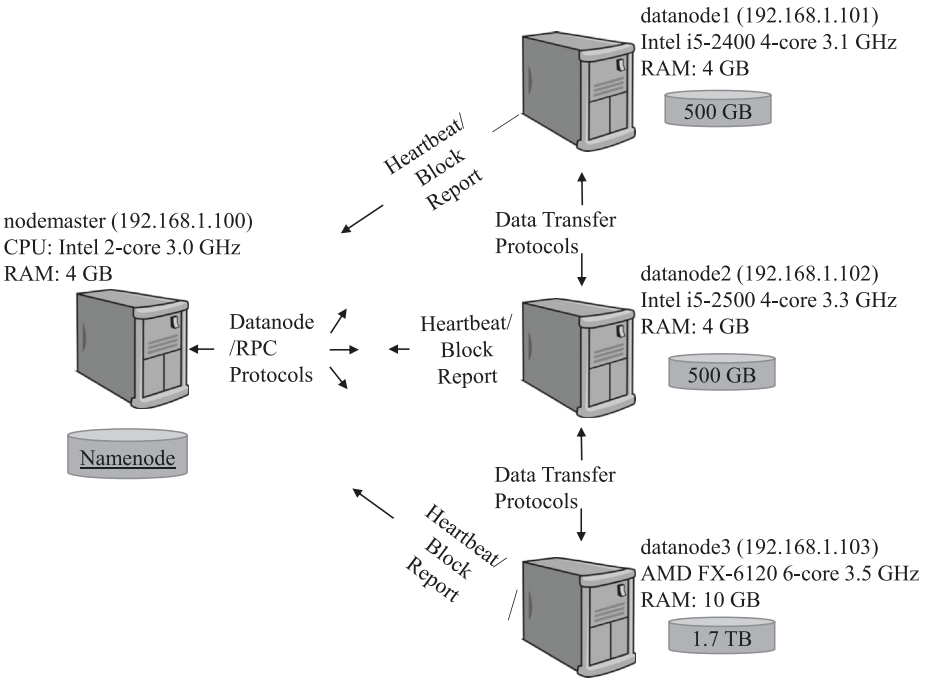

Fig. 1. Cluster specifications.

Table 1
Block placement.

Time Series	File	Block ID	Dnode 1	Dnode 2	Dnode 3
1	file1.txt	1073741840	X		X
2	file2.txt	1073741841		X	X
3	file2.txt	1073741842		X	X
4	file2.txt	1073741841		~~X~~	~~X~~
5	file2.txt	1073741842		~~X~~	~~X~~
6	file3.txt	1073741843	X		X
7-43					
44	file4.txt	1073741881	X		X
45	file4.txt	1073741882	X		X
46	file1.txt	1073741840	~~X~~		~~X~~

modification operations. Given this property, it is deduced that blocks of data associated with the missing generation stamps were created and deleted in a previous state. In some cases, data block modifications can be determined.

Namenode audit logs reveal data block data node destinations including deleted files. Block allocation is discovered with the generation stamp appended to the Block ID with an underscore. The Datanodes are identified via IP addresses and port numbers. In one case, we have discovered that the single block for file1.txt was written to data node3 and data node1 in the cluster. Times are converted for the period boundary and the Namenode log file entries collected within the interest period. Similarly, Datanode audit log file entries can be searched for the period of interest to verify block replication. Table 1 shows a partial block placement list. The probability of recovery of deleted files is a function of block pool space, post-deletion writes to disk, disk drive types, and the local file system disk management scheme among other variables.

We conclude the findings from this experiment support further research into the potential of high-level event reconstruction from namespace metadata. The capacity to wholly reconstruct and map to data nodes with the outlined approach is a function of the audit log and edits files archiving policies.

4. Conclusions and future work

This research presented an approach to reconstruct events over a time period in a Hadoop 3.2.0 HDFS to reveal system storage state transitions. The approach used only HDFS metadata at the data storage layer and demonstrates through inherent metadata materialization sequencing properties it is possible to utilize a relatively small subset of metadata for rapid reconstruction. Future work entails expanding the file operations and type data set to replicate a realistic big data environment, automating the reconstruction and developing a formal definition of the process.

Acknowledgements

This work being reported partially supported by the National Science Foundation under Grant No. CNS-1726069.

References

Grispos, G., Glisson, W.B., Storer, T., 2013. Using smartphones as a proxy for forensic evidence contained in cloud storage services. In: Proc. Annu. Hawaii Int. Conf. Syst. Sci, pp. 4910–4919.
"Hadoop – Apache Hadoop 3.2.0," 2019. [Online]. Available: https://hadoop.apache.org/docs/r3.2.0/index.html. [Accessed: 27-Mar-2019].
Sremack, J., 2015. Big Data Forensics – Learning Hadoop Investigations.
White, T., 2015. Hadoop: the Definitive Guide, fourth ed.