Job Feature Aware File Location Optimization

Makoto Nakagami†
Electrical Eng. and Electronics
Kogakuin Univ. Graduate School
Tokyo, Japan
cm19036@ns.kogakuin.ac.jp

Jose A.B. Fortes
Advanced Comp. and Info.
Systems (ACIS) Lab, Dept. of ECE
University of Florida
Gainesville, USA
fortes@ufl.edu

Saneyasu Yamaguchi
Dept. of Info. and Comm. Eng.
Kogakuin University
Tokyo, Japan
sane@cc.kogakuin.ac.jp

KEYWORDS

Hadoop, HDFS, Big data

1 INTRODUCTION

I/O performance improvement is essential for improving the big data processing platform such as Hadoop. Because of its large size, HDDs are used in many big data analyses. We have proposed a method for improving the Sequential I/O performance of Hadoop by controlling file placement location in ZBR HDDs [1][2]. In this paper, we explain a method for improving Hadoop I/O performance by optimizing file placement location in an HDD and evaluate its performance.

2 OPTIMIZATION OF FILE STORAGE

The method proposed in [1] improves the sequential I/O performance of Hadoop by placing a file at the outermost location in usable areas at every placing time. This does not take the features of jobs into account. We call this *outermost method*. The method of [2] optimizes file placement location considering the features of Hadoop jobs. We call this *job-aware method*. This method divides a target HDD into two areas, which are outer and inner areas. Files can be stored in the outer area per the following order of priority.
1) temporary files used by I/O-intensive processes.
2) temporary files not used by I/O-intensive processes.
3) permanent files used by I/O-intensive processes.
4) permanent files not used by I/O-intensive processes.

3 EVALUATION

We measured the performance with Map-heavy, Shuffle-heavy, and Reduce-heavy jobs [2]. Map-heavy jobs are CPU-intensive and their files are temporary. Shuffle-heavy jobs are I/O-intensive and their files are temporary. Reduce-heavy jobs are I/O intensive and their files are permanent. As shown in Fig. 1, in the case of the *job-aware method,* the files of Shuffle-heavy and Map-heavy jobs are stored in the outer area and files of Reduce-heavy jobs are stored in the inner area.

To evaluate these methods, a series of Hadoop job set wes executed. One job set comprised 27 job groups, wherein a sequence of map-, shuffle-, and reduce-heavy groups were repeated 9 times. Each job group comprised 20 jobs.

Figure 2 shows the average time required to execute 5 job sets. The results in this indicate that the job-aware method reduced the overall execution time by 15.4% and 11.1% compared to that of the normal and outermost methods, respectively.

4 CONCLUSION

In this paper, we evaluated the performance of the outermost and job-aware methods. For future work, we plan to evaluate these methods for concurrent jobs in the fully distributed mode.

ACKNOWLEDGMENTS

This work was supported in part by JST CREST Grant Number JPMJCR1503, Japan. This work was also supported by JSPS KAKENHI Grant Numbers 17K00109, 18K11277. This work is also funded in part by a grant (NSF ACI 1550126 and supplement DCL NSF 17-077) from the National Science Foundation, USA.

REFERENCES

[1] Eita FUJISHIMA Kenji NAKASHIMA Saneyasu YAMAGUCHI, "Hadoop I/O Performance Improvement by File Layout Optimization", IEICE TRANSACTIONS on Information and Systems, Vol.E101-D No.2 pp.415-427, doi: 10.1587/transinf.2017EDP711

[2] Makoto Nakagami, Jose A.B. Fortes, Saneyasu Yamaguchi, "Job-aware Optimization of File Placement in Hadoop, " BDCAA 2019 The 1st IEEE International Workshop on Big Data Computation, analysis, and Applications, COMPSAC 2019, July 2019.

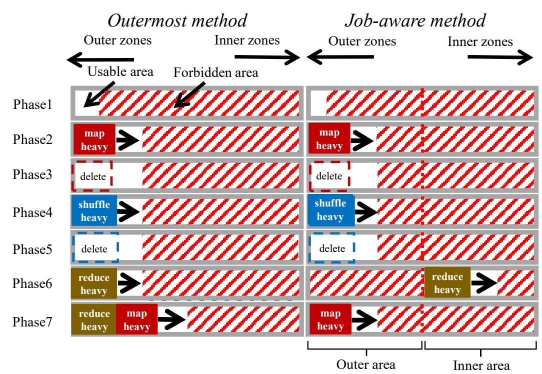

Figure 1: File-storage strategies of these methods

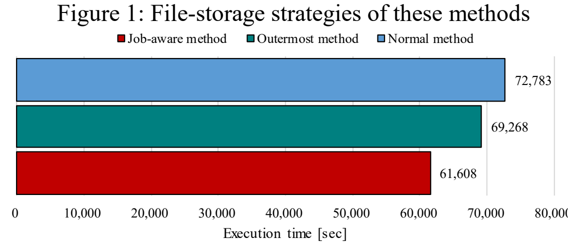

Figure 2: Total execution time of job set