SmartStore: A New Metadata Organization Paradigm with Semantic-Awareness

Yu Hua* Hong Jiang† Yifeng Zhu‡ Dan Feng* Lei Tian*†
*Huazhong University of Science and Technology †University of Nebraska-Lincoln ‡University of Maine
{csyhua, dfeng, ltian}@hust.edu.cn {jiang, tian}@cse.unl.edu zhu@eece.maine.edu

I. Introduction

Fast and flexible metadata retrieving is critical in the next-generation data storage systems. As the storage capacity approaches the Exabyte level and the stored files number is in the billions, directory-tree based metadata management widely deployed in conventional file systems can no longer meet the requirements of scalability and functionality. At the same time, new I/O interfaces are of great necessity to allow users to flexibly locate target files in a large-scale storage system. Of particularly desirable interfaces are range query and top-k query, where the former identifies files whose attribute values are within a given range, while the latter locates *k* files whose attributes are closest to given values. For example, a user may wish to obtain the answers to `Range Query:` *"Which experiments did I run yesterday that took less than 30 minutes and generated files larger than 2.6GB?"* and `Top-K Query:` *"I can not accurately remember a previously created file but I know that its file size is around 300MB and it was last visited around Jan.1, 2008. Can the system show 10 files that are closest to this description? "*.

This work-in-progress report proposes a novel decentralized semantic-aware metadata organization paradigm, called *SmartStore* [1], to efficiently organize file metadata into a semantic R-tree through semantic analysis on file metadata, which enables efficient complex queries including range and top-k queries.

Different from other state-of-the-art work, such as Spyglass [2] and Connections [3], SmartStore aims to *construct* a semantic-aware metadata organization by identifying semantic correlation of file multi-dimensional attributes, not solely relying on locality of namespace and access patterns in *existing* systems. Therefore, SmartStore becomes more flexible and adaptive in organizing file metadata by constructing potentially multiple logical semantic R-trees when considering different available attributes in different real-world applications. The co-existence of multiple organization forms renders SmartStore decentralized and scalable.

II. Approach

The basic idea behind SmartStore is to group and store files according to their metadata semantic correlations. A tool named Latent Semantic Indexing (LSI) [4] is used to measure the semantic correlations. In our design, metadata semantics are used to help aggregate highly correlated files into groups that are more likely to successfully serve a complex query. Thus, query and other relevant operations can be completed within one or a small number of such groups. Our goal is to avoid or minimize brute-force search that is widely used in a directory-tree based file system during a complex query. In addition, semantic grouping can also improve the system scalability. Most operations, such as insertion/deletion and queries, can be executed in parallel within some groups.

Figure 1 uses an example to compare conventional file systems and SmartStore. The SmartStore consists of index and storage units based on grouping and mapping operations to serve range and top-k queries.

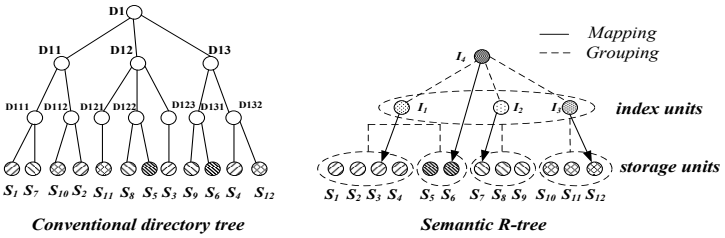
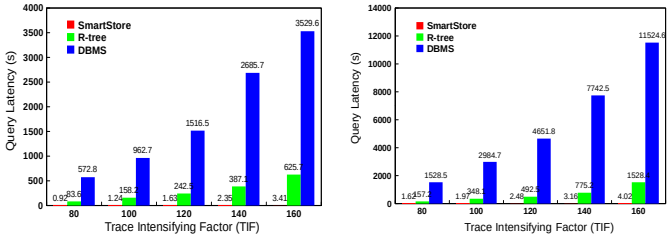


Fig. 1. Comparisons with conventional file system.

III. Current Status

We have implemented a prototype of SmartStore that iteratively aggregates files into groups until a single group is obtained. These groups are organized into a semantic R-tree data structure. The SmartStore prototype is implemented in Linux environments. In order to emulate the I/O behaviors of a large storage system, we scale up the I/O traces both spatially and temporally. The scale-up factor is denoted as the *Trace Intensifying Factor* (TIF). Figure 2 compares the query latency of SmartStore, R-tree without semantic grouping and one conventional DBMS in terms of range and top-k queries using the *MSN* trace [5].



(a) Range query. (b) Top-k query.

Fig. 2. Query latency comparisons using *MSN* trace.

Future work will extend current implementation to improve fault tolerance and information consistency in SmartStore, and to enhance the exploration and extraction of semantics. In addition, we also hope to compare SmartStore with other state-of-the-art architectures when examining real-world applications.

References

[1] Y. Hua, H. Jiang, Y. Zhu, D. Feng, and L. Tian, "SmartStore: A New Metadata Organization Paradigm with Metadata Semantic-Awareness for Next-Generation File Systems," *Technical Report, University of Nebraska-Lincoln, TR-UNL-CSE-2008-0012*, November, 2008.

[2] A. W. Leung, M. Shao, T. Bisson, S. Pasupathy, and E. L. Miller, "Spyglass: Fast, Scalable Metadata Search for Large-Scale Storage Systems," *FAST*, 2009.

[3] C. Soules and G. Ganger, "Connections: using context to enhance file search," *SOSP*, 2005.

[4] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent Semantic Indexing: A Probabilistic Analysis," *Journal of Computer and System Sciences*, vol. 61, no. 2, pp. 217–235, 2000.

[5] S. Kavalanekar, B. Worthington, Q. Zhang, and V. Sharda, "Characterization of storage workload traces from production Windows servers," *IEEE International Symposium on Workload Characterization*, 2008.