A Survey on Different File System Approach

Priya N. Parkhi
M.Tech [CSE]
Dept. of Computer Science and Engineering
St. Vincent Pallotti College of engineering, Nagpur, India

Vivek B. Kute
Associate Professor and HOD
Dept. of Computer Science and Engineering
St. Vincent Pallotti College of engineering, Nagpur, India

ABSTRACT
This paper, provide survey of the proposed namespace management schemes for file system. Namespace management can be used to reduce exhaustive search over all directories. Namespace using semantic correlation can also increase search ability. File system namespace as an information organizing infrastructure is a help to improve system's quality of service such as performance, scalability, and ease of use. This paper discusses the improvement to be made for future proposed namespace schemes. This paper provides reader with the basis for research in namespace schemes for file system.

Keywords
Semantic correlation, multi-dimensional attributes, locality sensitive hashing.

1. INTRODUCTION
File system manages storage and retrieval of data. It helps to find piece of data from large amount of data .This large data volume is due to explosive growth in social networking sites, business transactions, cloud computing etc. Almost all available file systems follow hierarchical tree based principle; that creates performance bottleneck as, lookup always starting from root directories. Fast data finding become difficult and time taking with the directories that become larger with time and no full (exact) path available; as entire directory has to be searched. In tree structure directory there is no correlation of access file with others so it's not possible to get correlated files.

This paper tries to review different file system techniques and design effective namespace management scheme by using best techniques to improve file system's quality of service.

From the survey of different papers this scheme use multidimensional attribute instead of only one or two attributes to overcome problem of traditional file system. It extracts semantic correlations among files and aggregate correlated files into manageable groups to achieve fast and accurate lookups. This scheme is implemented as a middleware in conventional file systems. Its correlations and file groups identifier used to facilitate file pre-fetching and data de-duplication. Semantic correlations efficiently identify among files by using Locality sensitive hashing algorithm

(LSH) [1]. LSH has the advantages of both locality preservation and fast identification. It works well in identifying correlated data but suffers from the space-overhead problem, as the amount of data increases. Since, LSH requires many hash tables to maintain correlated data; the space overhead becomes a potential performance bottleneck. To overcome this problem R-tree [2] structure replaces the original hash tables. It stores the correlated data, and represents their multi-dimensional attributes in the R-tree nodes

Namespace scheme design is integrated into modern file systems. Our goal is to complement existing file systems and improve system performance. This scheme improves system performance by considering following issues:

1) Considering multi-dimensional attributes, instead of one-dimensional attributes such as pathnames, to represent a file.

2) Using locality sensitive hashing (LSH) to automatically organize semantically correlated files without the involvement of end-users or applications to reduce performance overhead.

3) Use as transparent middleware that can be deployed without modifying the kernel or applications.

New namespace scheme provides users with two auxiliary namespace views, i.e., default (traditional hierarchy) and customized (New design)[3].

2. LITERATURE SURVEY
The performance of hierarchical file system degrades with increase in amount of data, hence direct control on database not possible. Therefore some lightweight in between database and raw storage is required. Due to its irrelevance, restrictiveness and limited performance hierarchical file system become handicap. So new way of data access and process is develop i.e. hFAD (hierarchical file system are dead) [4] ,that architecture avoid hierarchical namespace and uses tagged and search base namespace.

A special design for large storage file system introduces new metadata search technique such as hierarchical partitioning, signature files, snapshot base metadata, partitioning versioning. This techniques helps in improvement in performance, crawling and file updates. It shows faster search performance than existing systems. Spyglass use limited-dimensional correlations either in access time or in reference space.[5]

An abstraction Qufile[6] helps to encapsulate different physical representation of same logical data. It returns representation of data specification policy. This file system split mechanism and policy. It reduces context sensitive system and easily deploy on new system hence it already provide mechanism.. Developer need to write code for policy only .Qufile provides transparency to user and application. Different logical representation of same data achieve through view interface.

Achieving scalability and functionality requirement with growing dataset in hierarchical file system are difficult, hence semantic aware-organization called smart-store[7], which uses metadata of files into semantic-aware group by using information retrieval tools. The decentralized design of smart-store improves system scalability and reduces query latency. It is the first study to design and implements storage architecture for complex query such as top-k query. It reduces scope of complex query from Brute force search to single or minimum number of semantically co-related groups. It helps to optimize storage system design such as, de-duplication, cashing and pre-fetching. Linear Brute force and spyglass uses zero and one dimensional correlation respectively. But smart-store use semantic correlation which comes from higher dimensional correlation.

Selecting appropriate attribute is challenging constraints. When dimensionality increases performance get slower i.e. the curse of dimensionality. Two item close by in one space might be far away in another space and also two items are correlated when observe in one attribute subset might be uncorrelated to another attribute subset i.e. dimensionality heterogeneity [8].

Locality sensitive hashing (LSH)[9] method perform probabilistic dimension reduction of high dimensional data. The basic idea here is to mapped the similar item into same bucket with high dimensionality. LSH aims is, to maximization in probability of collision of similar items. It is a ways like nearest neighbor search.

Nearest neighbor search[10] finds closest point. Closeness is measured in the terms of dissimilarity function. In dissimilarity function less similar are the object, larger the function value. By finding approximate nearest neighbor in high dimension, helps to remove curse of dimensionality

Database system need index mechanism to retrieve data quickly. Traditional indexing method is not suitable for data of non-zero size located in multidimensional space. Dynamic index structure called R-Tree fulfills this need and also it provides algorithm for searching and updating. It is height balance tree similar to B-tree. It can split data space into hierarchically nested bounding boxes that may contain several data entities within the bounding box. It can efficiently support point, range and top-k queries by maintaining index records in its leaf nodes. An index record is a reference pointer to data. It use solid minimum bounding rectangles i.e. bounding boxes, to indicate the queried regions. It is a completely dynamic index structure that is able to efficiently support data updates and provide efficient query services by visiting only a small number of nodes in a spatial search. R-tree structure replace the original hash tables, store the correlated data, and represent their multi-dimensional attributes in the R-tree nodes.[11]

In most of the traditional file systems design has number of disk operations as, it has metadata lookup. It perform all metadata lookup in main memory. Haystack[12] help to achieve high throughput, low latency, fault tolerance and cost effectiveness. Haystack is incrementally scalable.

Semantic file system (SFS) [11] is one of the first file systems. It extends the traditional file system hierarchies by allowing users to search customized file attributes. It also provides more storage abstraction than traditional file system. SFS introduce concept of virtual directories. SFS file system provides flexible access to system contents. It does not consider the semantic context implicitly and explicitly represented in file metadata when serving complex queries.

3. PROPOSED APPROACH

The drawbacks of spyglass [2], smart-store [4], qufile [3] and Semantic file system [11] can be removed. The basic solutions to file systems are limited by the inherent weaknesses of the directory-tree naming scheme. The weakness of conventional namespace schemes have exposed as the data volume and complexity keep increasing rapidly.

1. Limited system scalability

2. Dependency on end-users to organize and lookup data.

3. Lack in metadata-semantics exploration.

The multi-dimensional attributes can be used to overcome the problem of the weaknesses 1 and 2 instead of using one-dimensional attribute such as pathnames. The metadata of files that are strongly correlated are automatically aggregated and stored together in namespace. While performing file lookup, namespace will also present the files that are strongly correlated to this searched file. This allows the user to access the correlated files easily without performing additional searches or directory tree navigations.

Locality sensitive hashing (LSH) removes weakness 3 by automatically organizing semantically correlated files without the involvement of end-users or applications. It has very little performance overhead since LSH has a low complexity of probing constant-scale buckets. This design represents each file based on its semantic correlations to other files. To optimize the overall system design, the semantics residing in files correlation are obtained from multiple dimensions. It is implemented as a transparent middleware that can be deployed/embedded in most existing file systems without modifying the kernels or applications.

Proposed design will provide users with two namespace views, i.e., default (traditional hierarchy) and customized (semantic namespace). Both views hide the complex details of the physical representation of individual files. Our approach is particularly helpful in avoiding brute-force search, which is time consuming in large file systems. We aim to design a new approach that helps quickly to locate target files in a large scale file system.

4. SUMMARY AND CONCLUSION

This paper is a survey of namespace management schemes for file system which shows why traditional file system is not efficient as data volume increase. The proposed approach is an efficient method to identify semantic correlations among files by using a simple and fast LSH-based lookup. In addition, the semantic correlation helps to improve some system functions, such as data de-duplication and file pre-fetching. In future, this system will be release for public use.

5. REFERENCES

[1] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li, "Multi-Probe LSH: Efficient Indexing for High-Dimensional Similarity Search," Proc. VLDB, pp. 950-961, 2007.

[2] M. Seltzer and N. Murphy, "Hierarchical File Systems are Dead," Proc. 12th Conf. Hot Topics in Operating Systems (HotOS'09), 2009.

[3] A.W. Leung, M. Shao, T. Bisson, S. Pasupathy, and E.L. Miller, "Spyglass: Fast, Scalable Metadata Search for Large-Scale Storage Systems," Proc. Seventh USENIX Conf. File and Storage Technologies (FAST), 2009.

[4] K. Veeraraghavan, J. Flinn, E.B. Nightingale, and B. Noble, "quFiles: The Right File at the Right Time," Proc. USENIX Conf. File and Storage Technologies (FAST), 2010.

[5] Y. Hua, H. Jiang, Y. Zhu, D. Feng, and L. Tian, "SmartStore: A New Metadata Organization Paradigm with Semantic-Awareness for Next-Generation File Systems," Proc. ACM/IEEE Supercomputing Conf (SC),2009.

[6] P. Indyk and R. Motwani, "Approximate Nearest Neighbors:Towards Removing the Curse of Dimensionality," Proc. 30th Ann.ACM Symp. Theory of Computing (STOC), 1998.

[7] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of imensionality," Proc. 30th Ann. ACM Symp. Theory of Computing (STOC), 1998.

[8] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," ACM SIGMOD Record, vol. 1, pp. 47-57, 1984.

[9] D. Beaver, S. Kumar, H. Li, J. Sobel, and P. Vajgel, "Finding a Needle in Haystack: Facebooks Photo Storage," Proc. Ninth USENIX Conf. Operating Systems Design and Implementation(OSDI), 2010.

[10] D.K. Gifford, P. Jouvelot, M.A. Sheldon, and J.W.O. Jr, "Semantic File Systems," Proc. Symp. Operating Systems Principle (SOSP), 1991.

[11] Yu Hua, Hong Jiang, Yifeng Zhu and Lei Xu," SANE: Semantic-Aware Namespace in Ultra-Large-Scale File Systems", VOL. 25, NO. 5, MAY 2014.