RBF: A New Storage Structure for Space-Efficient Queries for Multidimensional Metadata in OSS

Yu Hua[1] Dan Feng[1] Hong Jiang[2] Lei Tian[1]
[1]School of Computer, Huazhong University of Science and Technology, China.
[2]Department of Computer, University of Nebraska Lincoln, USA.

I. Introduction

Object-based storage systems (OSS) have evolved from the traditionally unintelligent management to intelligent and self-managed schemes by delegating low-level management activities to storage devices, and providing new functionalities, such as greater scalability and dynamic reconfiguration. An object consists of data, user-accessible attributes and device-managed metadata, which is responsible for mapping data structures, such as files and directories, to blocks on storage devices. Such metadata usually contains multidimensional information for representing the mapping relationship with objects that have both physical and logical attributes, such as access time, data size, request amount, access pattern, and QoS agreement. Thus, substantial storage space has to be used to store multidimensional metadata, along with highly complicated operations of querying multidimensional metadata, due to complex data organization structures.

Common query operations, mainly including point query and range query that are independent of the underlying data layout, allow one to search objects that have multidimensional attributes, with the help of multidimensional metadata.

- **Point query** determines whether a given object is a member of a data set. For example, a request of point query for multidimensional metadata may wish to know whether an object with “ID=xyz” and “data size=100GB” attributes is a member of the current storage system.
- **Range query** obtains all objects whose attribute values exist within the ranges of a query request. For example, a request of range query for multidimensional metadata may wish to know all the objects having “data size>150GB” and “access time<20:00” attributes.

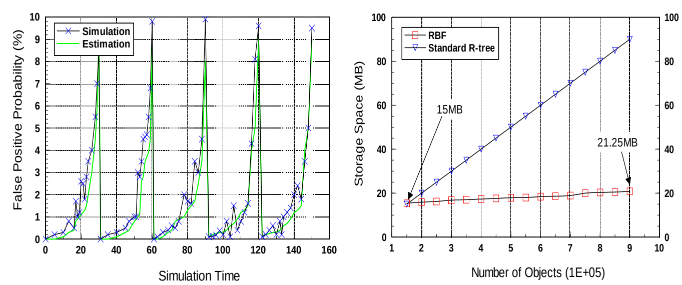
II. Approach

We propose a new space-efficient storage structure, called the R-tree with Bloom Filters (RBF), to store multidimensional metadata and achieve point and range query with low operational complexity. The basic idea of our RBF is to expand the classical R-tree [1] to incorporate space-efficient Bloom filters [2] in R-tree nodes, maintaining multidimensional range information and achieving space efficiency.

An R-tree, evolved from B-tree, can efficiently support multidimensional range queries by splitting data space with hierarchically nested bounding boxes, which can contain several data entities within certain ranges. Unfortunately, an R-tree cannot efficiently support point query because membership query can be executed only in the leaf nodes. Corresponding operations lead to a low query efficiency. On the other hand, a Bloom filter is a space-efficient data structure and can support point query very well. A Bloom filter can represent a set of items as a bit array using several independent hash functions and support the querying of items. This compact representation is a tradeoff as it achieves high space efficiency at the expense of a small probability of false positive in the membership query. Bloom filters are widely used in practical applications when space is at a premium. Therefore, we plan to combine R-tree and Bloom filter into our RBF by adding a space-efficient Bloom filter in each R-tree node to support point query with O(1) time complexity.

III. Current Status

We examine the storage structure and query techniques for multidimensional metadata management in the context of a petabyte-scale storage system being designed and developed at the Huazhong University of Science and Technology to handle both general-purpose and scientific computing workloads. Our architecture will eventually consist of tens of metadata servers (MDSs), thousands of object-based storage devices (OSDs) and allow for storing mass data for supporting Geographic Information System (GIS) applications. This work is supported by the National Basic Research 973 Program (Grant.2004CB318201).



(a) False Positive Probability. (b) Storage Space.

Fig. 1. Preliminary Results in terms of false positive probability and storage space.

So far, we have constructed a real mass storage system with a 10-terabyte capacity and implemented partial functions of point and range queries based on RBF structure. Figure 1 shows preliminary results in terms of false positive probability and storage space, indicating RBF’s potential advantages. We have also been exploring the load-balancing advantage of RBF to improve the scalability and reliability of our storage system. The objective of load balance in RBF is to make the nodes of the same layer represent approximately the same number of objects. Load balancing can efficiently decrease the false positive probability of Bloom filters in RBF nodes.

In the near future, we are extending our storage structure to take into account the queries for “hot spots” and a more accurate lookup scheme. We shall evaluate the performance on our real petabyte-scale storage system.

References

[1] Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis, “R-trees: Theory and applications,” *Springer-Verlag*, 2005.

[2] A. Broder and M. Mitzenmacher, “Network applications of Bloom filters: a survey,” *Internet Mathematics*, vol. 1, pp. 485–509, 2005.