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.

Outline

- Introduction
- Motivations
- Approach
- Current status
- Next step

Introduction: OSS

- Object-based Storage Systems (OSS):
  - Intelligent and self-managed schemes that delegate low-level management activities to storage devices;
  - Provide new functionalities, such as greater scalability, security, and dynamic reconfiguration.

Introduction: Object & Attributes

- An object consists of data, user-accessible attributes and device-managed metadata;
- Metadata 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:
  - For example, access time, data size, request amount, access pattern, and QoS agreement.

Introduction:
Point & Range Query in OSS

- **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.

Motivations

- 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.
- Our goal in this study:
  - *Space savings;*
  - *Achieve efficient point & range query for multidimensional attributes;*

Approach

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

R-tree & Bloom filter

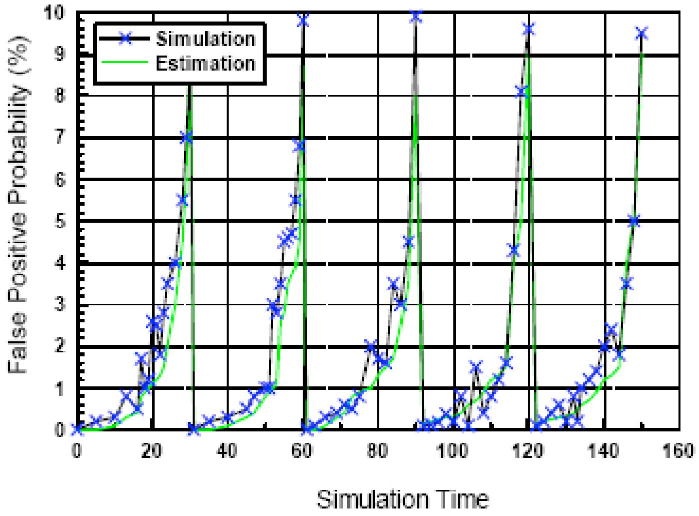
- An *R-tree* can efficiently support multidimensional range queries by splitting data space with hierarchically nested bounding boxes, which can contain several data entities within certain ranges.
  - Unfortunately, an R-tree <u>cannot efficiently support point query</u> because membership query can be executed only in the leaf nodes.
- *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 membership queries.
  - This compact representation is a tradeoff as it achieves high space efficiency at the expense of <u>a small probability of false positive</u> in the membership query.
- ***We 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.***

Current Status

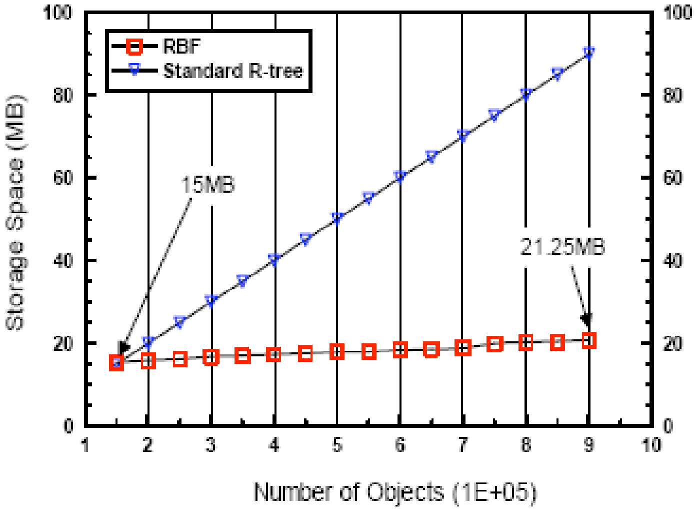
- We examine the storage structure and query techniques for multidimensional metadata management in the context of a *Petabyte-scale* storage system being designed and developed at the Huazhong University of Science and Technology (HUST) to handle both general-purpose and scientific computing workloads.
- Our architecture will eventually consist of:
- ➢ tens of metadata servers (MDSs);
- ➢ thousands of object-based storage devices (OSDs);
- ➢ allow for storing mass data for supporting Geographic Information System (GIS) applications.

Current Status

- We have constructed a real mass storage system with a 10-terabyte capacity and implemented partial functions of point and range queries based on the RBF structure.
- Preliminary Results:



(a) False Positive Probability.



(b) Storage Space.

Next step

- 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 will extend 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.