Scalable metadata search for large-scale geo-distributed storage systems

Dimitrios Vasilas

April 23, 2017

EuroSys - EuroDW 2017

Research Problem - Motivation

Key-value storage systems

- **Key value stores increasingly adopted for their performance, scalability, availability**
  Apache Casandra, CouchBase Server, Redis, Riak

- **Well-suited for various use cases**

  Product recommendations
  Ad servicing
  Session management

Data retrieval

- **Simple interface to store, retrieve and update data**
  Simple get, put, delete commands. Do not require complex query language
  Simplicity of the model makes the systems fast, scalable and flexible

- **Various use cases require search based on partial information**
  Difficult to implement applications that need to retrieve data by information other than their key

Use case: Lifecycle management

- **Applying policies to the backup, archival and migration of data.**
- **Example queries**
  - "Large objects not accessed recently"
  - "Objects created since the last system backup, and flagged as important"

Use case: User-defined tags

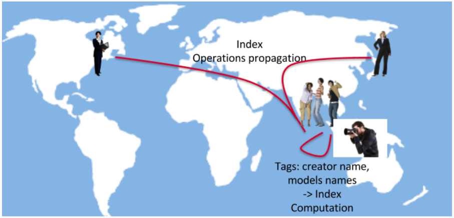
- **Various applications**

    Image
    Music
    Geospatial
    Biomedical

- **Advanced photo album**

  Photos are tagged with

    Location taken
    Person appearing
    View count



Research problem

Enable location and retrieval of data in object storage systems, based on partial information

Design and implement a metadata search sub-system for object storage systems

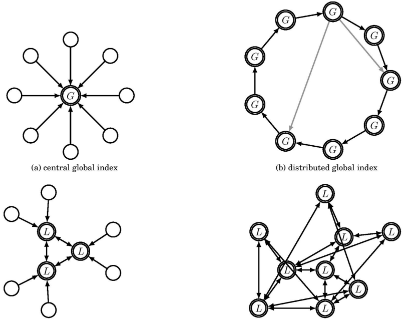
Challenges

- **Dataset size**
  Petabytes of data
  Billions of objects
- **Mutable data**
  Concurrent updates and queries
- **Geo-distributed index**
  Updates and queries from clients in different geographic locations
- **Mix of data types**
  Metadata include text, integers and complex data types (ACLs)

Existing Systems

Search in Peer-to-Peer systems

- **Centralized index**
  Limited scalability
  Single point of failure
- **Local indices**
  Peers index their local files
  Query flooding - Poor scaling
  Aggregated local indices
- **Distributed index**
  Keyspace divided among peers
  Built over a distributed hash table
  Only exact match queries



(a) central global index (b) distributed global index (c) aggregated local indices (d) strict local indices

Other approaches

- **Range and multi-attribute queries in peer-to-peer systems**
  Additional mechanisms over DHTs in order to preserve data locality
- **Metadata search in large-scale file systems**
  Leverage namespace locality in the hierarchical structure of file systems
- **MapReduce-based techniques**
  Widely adopted for tasks involving parallel computation

Our approach

Metadata search on Amazon S3

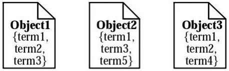
- **Extend Amazon S3 API with metadata search**
  List objects based on their metadata attributes
- **Multi-attribute queries**
  AND, OR logical operators
- **Exact match and range queries**

Distributed Inverted Index

- **Inverted index**
  Map metadata attribute values to sets of objects

- **Multi-attribute and range query support**
  Encode metadata attributes as text
  Index entries sorted lexicographically

- **Implementation relies on CRDTs**
  Replicated data types
  Convergence of conflicting operations

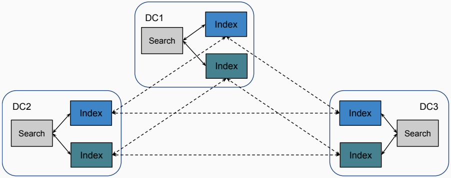


Key	List of Objects
term1	{Object1, Object2, Object3}
term2	{Object1, Object3}
term3	{Object1, Object2}
term4	{Object3}
term5	{Object2}
...	...

Distributed Inverted Index

- **Geo-distributed index**
  Peers organized in groups
  Each group assigned an attribute
  Index replicated among peers of a group

- **Implementation based on AntidoteDB**
  Highly-available, geo-distributed key-value store
  Embedded CRDT support



Next Steps

Next steps

- **Experimental evaluation**
  Evaluate performance and availability in a geo-distributed environment
  Evaluate the impact of CRDTs and AntidoteDB on the systems performance

- **Integrate in Scality's storage system**
  Experiment in a real-world environment

Future directions

- **Measure and bound index staleness**
  Estimate the amount of staleness between index and content
  Bound staleness below an application-specific threshold
- **Search on text and semi-structured data**
- **General model search in P2P networks**

Summary

- **How to efficiently locate and retrieve data based on partial information**
  Extend the simple key-value interface
  Maintain performance of the storage system
- **Petabyte scale**
- **Geo-distributed environment**
- **Mutable data**