Design and Implementation of a Metadata-Rich File System

Sasha Ames (student author, UCSC, LLNL), Maya B. Gokhale (LLNL),
Carlos Maltzahn (UCSC)

Lawrence Livermore National Lab
University of California, Santa Cruz

Please consider for both poster and WiP sessions.

Despite continual improvements in the performance and reliability of large-scale file systems, the management of user-defined file system metadata has changed little in the past decade. The mismatch between the size and complexity of large scale data stores and their ability to organize and query their metadata has led to a de facto standard in which raw data is stored in traditional file systems, while related, application-specific metadata is stored in relational databases. This separation of data and metadata requires considerable effort to maintain consistency and can result in complex, slow, and inflexible system operation.

To address these problems, we have developed the Quasar File System (QFS), a metadata-rich file system in which files, attributes, and file relationships are all first class objects. In contrast to hierarchical file systems and relational bases, QFS defines a directed graph data model composed of files and their relationships. QFS includes Quasar, an XPATH-extended query language for searching the file system. Our prototype implementation QFS manages the graph data model with memory-based indices for attributes, adjacency lists of files parents/children, and a query planner and processor specific to our language and data layout.

To evaluate QFS we chose an application that annotates documents with user-defined metadata for our experimental workload and have devised several query scenarios. Our results show a factor of two performance increase for data ingest to our QFS prototype implementation over the to conventional file system and Postgres database with up to 39 GB of user-defined metadata created. In the query experiments, simple and complex (5 join) queries, specifically for files, show QFS answering queries at 3 to 20 times faster on average. Response time measurements of semantic queries, for attribute values, exhibit Postgres 10 to 30 percent faster. Our future work will include improvements to the QFS query planner/processor and indexing structures to improve the consistency of query performance over the entire range of queries.