Petaminer: Using ROOT for Efficient Data Storage in MySQL Database

Computing in High Energy and Nuclear Physics
Prague | Czech Republic | 21 - 27 March 2009

Jack Cranshaw
David Malon
Alexandre Vaniachine
Argonne National Laboratory

Valeri Fine
Jérôme Lauret
Brookhaven National Laboratory

Paul Hamill
Tech-X Corporation

1. Introduction

High Energy and Nuclear Physics (HENP) experiments store Petabytes of event data and Terabytes of calibration data in ROOT files. The Petaminer project is developing a custom MySQL storage engine to enable the MySQL query processor to directly access experimental data stored in ROOT files. Our project is addressing the problem of efficient navigation to PetaBytes of HENP experimental data described with event-level TAG metadata, which is required by data intensive physics communities such as the LHC and RHIC experiments. Physicists need to be able to compose a metadata query and rapidly retrieve the set of matching events, where improved efficiency will facilitate the discovery process by permitting rapid iterations of data evaluation and retrieval.

Our custom MySQL storage engine enables the MySQL query processor to directly access TAG data stored in ROOT TTrees. As ROOT TTrees are column-oriented, reading them directly provides improved performance over traditional row-oriented TAG databases. Leveraging the flexible and powerful SQL query language to access data stored in ROOT TTrees, the Petaminer approach will enable rich MySQL index-building capabilities for further performance optimization.

2. Architecture

The Petaminer architecture is shown in Figure 1. The Petaminer storage engine plugs in to a MySQL server. The engine invokes ROOT and FastBit to read and index ROOT files. Standard MySQL client tools and APIs are used to query ROOT data from the MySQL server.

SQL data is stored in database tables as rows, which optimizes writes and updates. ROOT data is stored in columns, which is optimal for fast reads and large-scale data mining. Figure 2 compares SQL and ROOT data organization.

The Petaminer engine maps ROOT constructs such as TTrees and attributes to MySQL schema elements such as tables and columns. This permits users to intuitively compose SQL queries on ROOT data structures. Figure 3 shows a mapping of ROOT file structure to MySQL database schema.

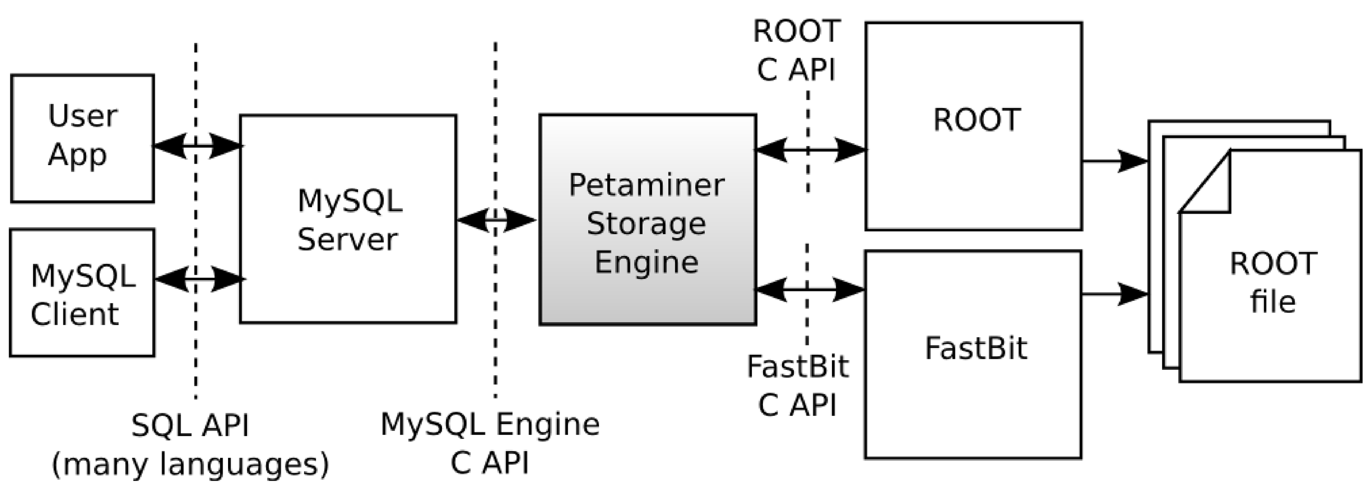

Fig. 1: Petaminer high-level system architecture

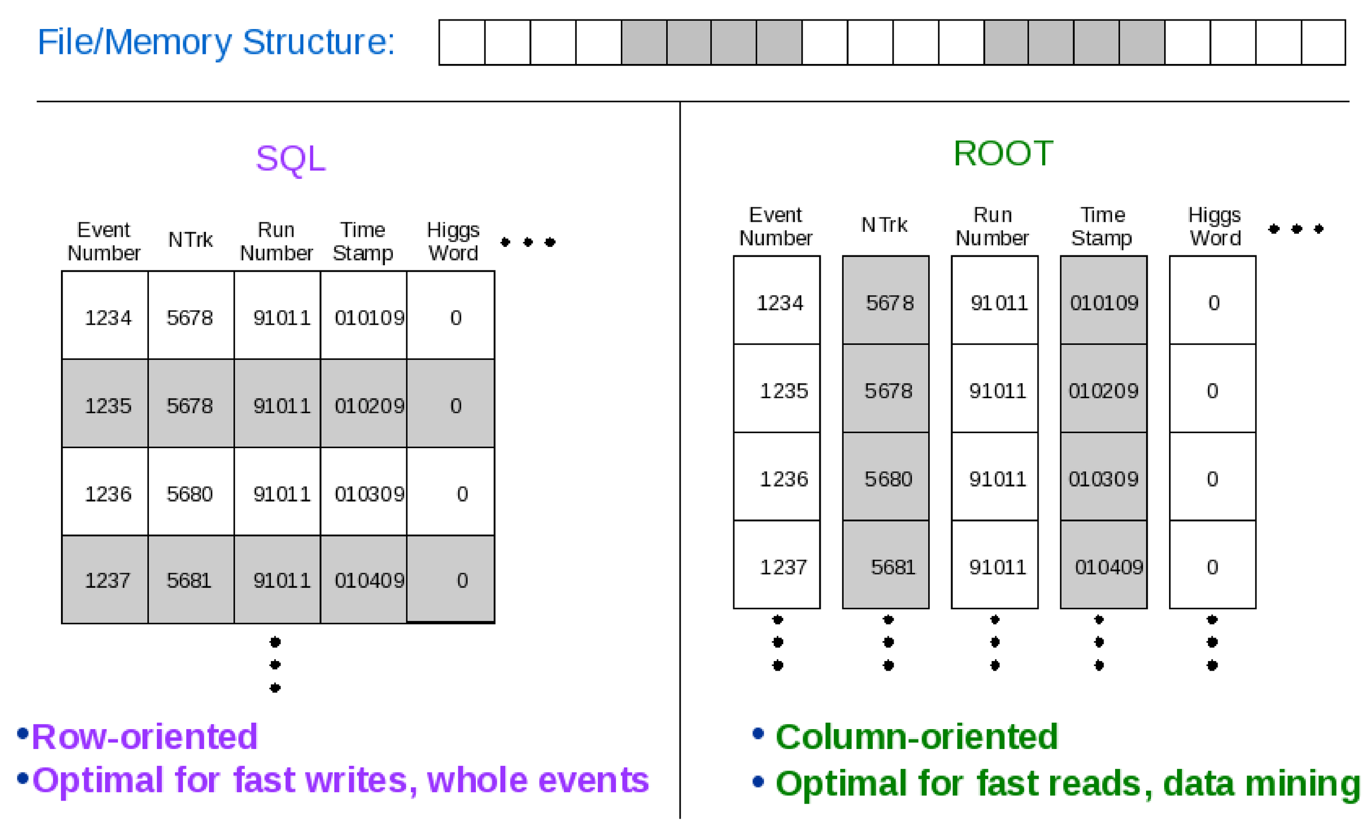

Fig. 2: SQL versus ROOT data organization

ROOT	MySQL
` TFile MyData.root { TTree tree { TBranch<Int_t> a1 TBranch<Float_t> a2 TBranch<Char_t> a3 } } `	` TABLE MyData { a1 INT, a2 DOUBLE, a3 VARCHAR } `

```
ROOT
TFile MyData.root {
  TTree tree {
    TBranch<Int_t> a1
    TBranch<Float_t> a2
    TBranch<Char_t> a3
  }
}
```

```
MySQL
TABLE MyData {
  a1 INT,
  a2 DOUBLE,
  a3 VARCHAR
}
```

Fig. 3: Mapping ROOT data to MySQL schema

3. Performance

The Petaminer prototype permits queries run via MySQL to read data directly from ROOT and be returned as SQL result sets. The Petaminer engine can index ROOT tables using FastBit bitmap indexing and use this index to optimize queries. Unindexed Petaminer queries have ~40% speed improvement over unindexed MySQL queries on simulated TAG DB data. Indexed Petaminer queries are ~10x faster than unindexed Petaminer queries. MySQL indexed queries are 5-6x faster than Petaminer indexed queries which suggests potential for further performance improvement. Figure 4 compares the performance results of queries using the Petaminer engine prototype versus generic MySQL.

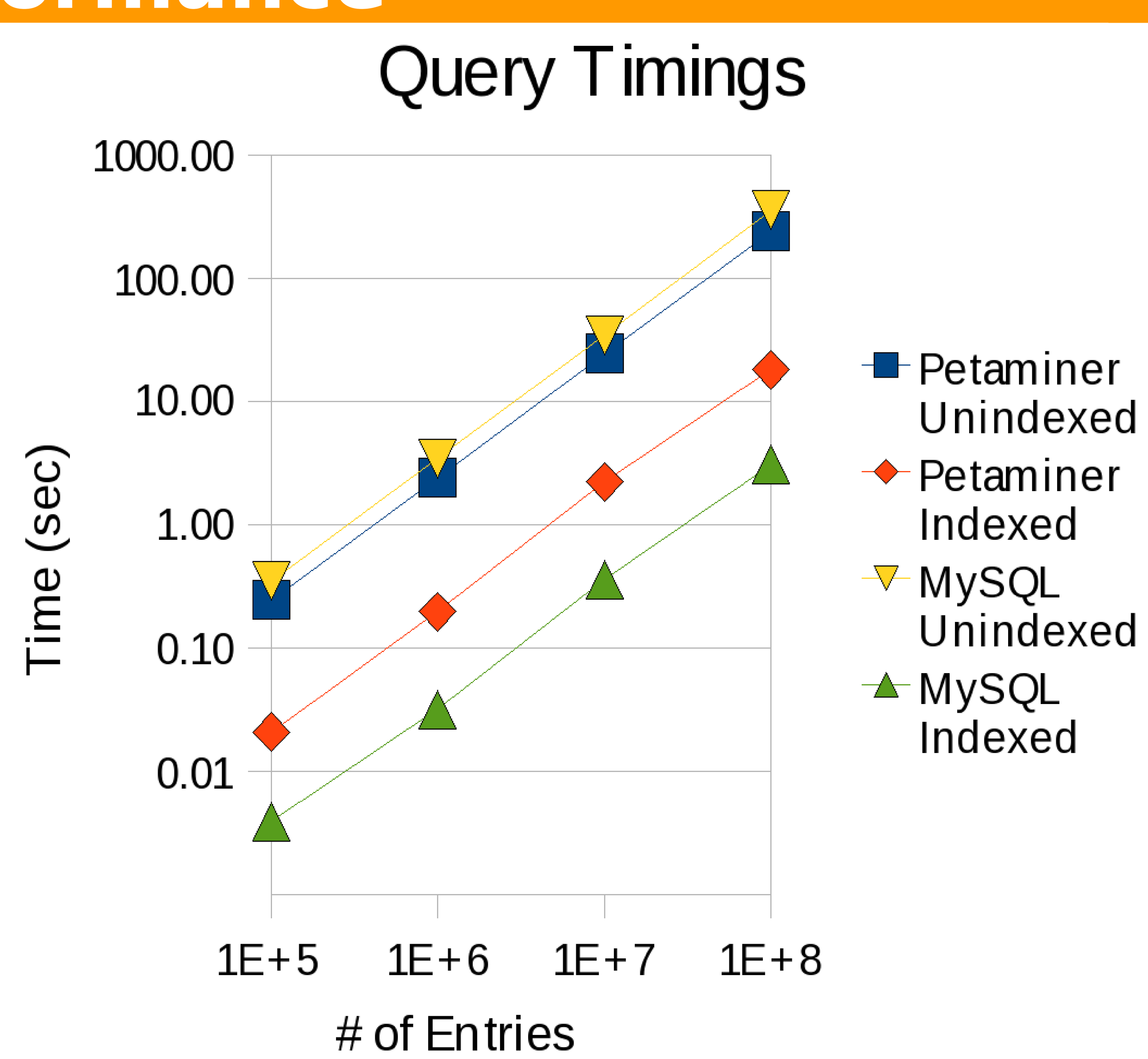

Fig. 4: Petaminer versus MySQL query performance (log-log scale)

4. Results and Future Work

The Phase I prototype integrates MySQL, ROOT and FastBit to demonstrate the feasibility of using MySQL to query ROOT data. This approach leverages the ease of use of SQL and the performance and scalability of ROOT to provide efficient query tools with automatic schema migration for evolving TAG DB metadata. The performance results confirm our hypothesis that queries against column-based ROOT data can provide increased efficiency over row-based SQL tables. The prototype has been installed at the Argonne US Atlas Support Center (ASC) for testing and evaluation.

Phase II will further optimize performance, demonstrate the capability to read Petascale data, implement additional functionality, and build a production-quality system. Possible Phase II tasks include query optimization; reading data from PROOF distributed storage; integration of ROOT TFormulas as MySQL User-Defined Functions; an embedded version of Petaminer that permits using SQL to read ROOT data without a MySQL server; query parallelization; and testing and evaluation on realistic Petascale distributed data.