International Journal of Modern Physics A

The New EventStore Data Management System For The CLEO-c Experiment

Chris Jones, Valentin Kuznetsov, Dan Riley, Gregory Sharp
Cornell University, Ithaca, NY 14853, USA

We discuss the new CLEO-c EventStore: a pluggable system that scales from personal needs up to collaboration-wide managed data. Its indexing and versioning features allow physicists to quickly access raw, reconstructed and MC data. It does simple bookkeeping of desired datasets and guarantees the reproducibility of various details of physics analysis. A wide range of physics queries and a variety of file formats are supported.

Keywords: EventStore; Data management; CLEO-c.

1. Introduction

In five years, the CLEO-c experiment will collect 200 TB of data and provide physicists a unique opportunity to measure some elements of CKM matrix to a few percent, probe "glueballs" and test many aspects of Lattice QCD, Heavy Quark Effective Theory and flavor physics [1]. Such very large data volumes require substantial management oversight. It is expected that new analysis techniques, e.g., Partial Wave Analysis (PWA), will rely on fast, reliable, random data access. The existing data management system based on Objectivity/DB™ suffers from many problems, including unnatural partitioning of data, slow data access and scalability issues. In addition, the use of proprietary software leads to unpredictable software upgrade cycles and licensing policies. The need to resolve such problems led us to start development of the EventStore data management system.

2. General Description

The CLEO-c EventStore holds the raw data produced by the CLEO-c detector, reconstructed data, user-generated skims and Monte Carlo data samples. Data can be stored in native file formats, relocated for load balancing, replicated and distributed among disks, HSM and caching systems. To provide efficient data access, EventStore uses additional index and location files. The format-independent index file is used for fast event finding, and the format-dependent location file provides for random access to the data. The additional 5% overhead to the data size is acceptable for EventStore needs. One of the desired features of EventStore is the persistent versioning information about stored data. Users access data by specifying a date-stamp. The underlying database resolves it to specific versions of the reconstructed data for each run range present in EventStore.

EventStore supports three models of event data access. In the first, the "personal" EventStore holds user skims or subsets of arbitrary data. It is stored on the local disk of a laptop or desktop computer. The embedded SQLite database provides indexing and versioning information. It supports a set of basic physics queries, such as dates and run ranges.

In the second model, the "group" EventStore manages a substantial set of data residing on a pool of disks designated for a particular group of physicists. A MySQL server is used as the backend.

Finally, the "collaboration" EventStore represents a large distributed system for data analysis with many servers and automated data management. The entire CLEO-c data will be stored in such a system. The event data will be located using the directory service and will be made available from hierarchical storage management systems using the data caching service. The system will be based at Cornell and will use a backend RDMBS (several candidates are currently under consideration). An external MetaData database can be accessed as a web service from legacy applications, supporting a variety of physics queries. Data can be requested by specifying an energy range, detector conditions, etc.

3. Implementation

For simplicity, we will skip a discussion of the CLEO-III/c data model, which can be found elsewhere [2]. To work with EventStore appropriate *EventStoreModule* (personal, group or collaboration) needs to be loaded. Data to process is set by a user request specifying the date-stamp chosen for data analysis. Optional run ranges, datasets and other physics queries can be provided. As shown in Fig. 1, the EventStore module resolves the query through SQL queries to the underlying DB and retrieves whichever index and location files are needed. The data delivery layer

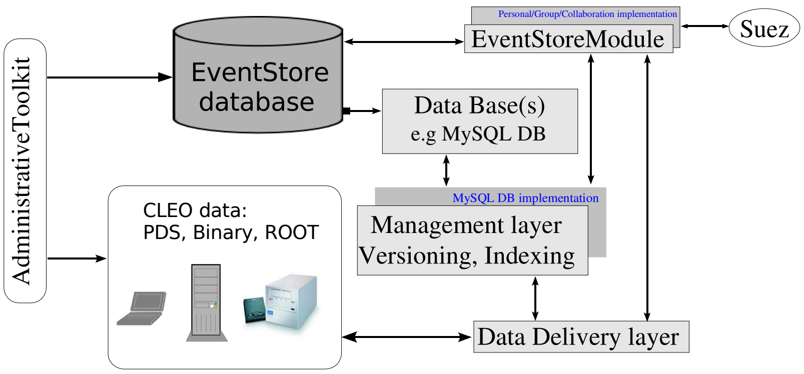

Fig. 1. The EventStore workflow. The shaded boxes represent a concrete implementation based on MySQL database whose table contents are shown on the left hand side.

opens the index file and searches the index of the related location file for the given run and event number. The location file contains information about stream type [2] and associated offsets in the data file for this record. Finally, using the offset, the data delivery layer accesses the requested data and makes them available to the user.

4. Results

A pluggable system allows us to use different DB backends for different EventStore sizes. We benchmarked the performance compared to a chain of files. Figure 2

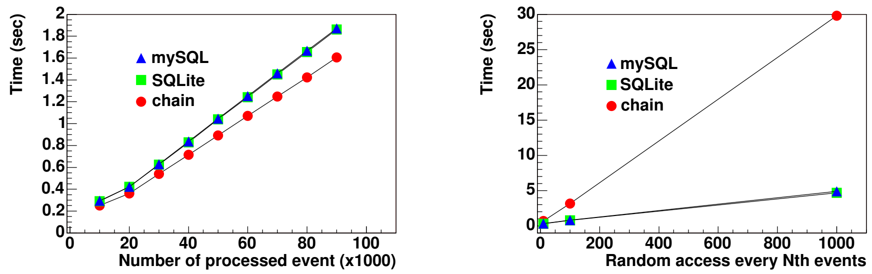

Fig. 2. Sequential (left) and random (right) data access. EventStore using SQLite DB (green rectangle), EventStore using MySQL (blue triangle) versus chain of files (red circles).

summarizes our results. Regardless of the underlying DB (SQLite or MySQL) EventStore is only 12% slower then a chain of files. At the same time, tremendous improvements have been achieved for random data access, a factor of 6 when accessing every 1000th event. EventStore demonstrates decent performance, gaining at least a factor of 5 compared to Objectivity/DB$^{\mathrm{TM}}$ on Solaris and reaching 11K event/sec rate on a 500MHz/512Mb UltraSparc node. On a Linux/Pentium/1GHz/256Mb with local IDE disk, our peak is 25K event/sec, far beyond our expectations. For a complete discussion see [3].

Acknowledgments

This work was supported by the National Science Foundation.

References

1. R. A. Briere et al., CLEO-c and CESR-c: A New Frontier of Weak and Strong Interactions, CLNS 01/1742.
2. M. Lohner, C. D. Jones, S. Patton, M. Athanas, P. Avery, The CLEO III Data Access Framework, CHEP'98, Chicago, USA, 1998.
3. C. D. Jones, V. Kuznetsov, D. Riley, G. Sharp, EventStore: Managing Event Versioning and Data Partitioning using Legacy Data Formats, to be published in proceedings of CHEP'04, Interlaken, Switzerland, 2004