Emerging Database Technologies and Their Applicability to High Energy Physics: A First Look at SciDB

D Malon, J Cranshaw, P van Gemmeren, Q Zhang
Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA
E-mail: malon@anl.gov

Abstract. Traditional relational databases have not always been well matched to the needs of data-intensive sciences, and to the needs of high energy physics data stores in particular. To address this mismatch, members of the database community and people involved with large scientific data stores in a variety of disciplines have inaugurated an open-source project, SciDB, that aims to develop and deliver database technologies suited to the needs of data-intensive sciences. This paper describes early experience using the first release of SciDB with an initial subset of high energy physics data structures and query patterns. It examines the early capabilities of SciDB, and describes requirements that further development must address if emerging database technologies such as SciDB are to accommodate the data structures, query patterns, computations, and use cases of high energy physics.

1. Introduction

While relational databases are widely used throughout the sciences, such databases are often not used to store the scientific data themselves at any significant scale. There are many reasons for this: no native support for necessary data types, including arrays, no scientific query or transform operators (not even relatively standard spatial and temporal query operators), a transaction model that limits scalability and potential for parallel and distributed processing, even though many data are in practice read-only (with updates handled by additional data versions rather than by changes to existing data), insufficient versioning, provenance tracking, and infrastructure to ensure reproducibility, and more. For many scientific applications (and even for commercial data mining applications), the row-oriented storage of a conventional relational database leads to performance issues even for certain analysis operations that are feasible in principle (and a commercial marketplace for column-oriented databases is beginning to emerge (cf. Vertica [1]). The experimental particle physics community has traditionally, for these and other reasons, turned to domain-specific file-based solutions such as ROOT [2], sometimes with a more technology-neutral intervening layer such as that provided by the LHC common persistence project POOL [3]. Other communities have also often tended to adopt domain-specific strategies.

A series of workshops (XLDB: for eXtremely Large DataBases [4]) has been inaugurated in recent years, bringing together a number of commercial database vendors, some of the leading figures in the

To whom any correspondence should be addressed.

Published under licence by IOP Publishing Ltd
U.S. academic database research community, and representatives of several scientific disciplines. One outcome of these workshops has been an attempt to document the requirements of these disciplines, and to propose how the database community might address them. A consensus developed in those workshops and in subsequent meetings that there is sufficient commonality among the requirements of several scientific disciplines that a common product might indeed be capable of addressing their needs, and out of this, the open-source SciDB project [5] was born.

2. SciDB

Motivated by the common needs of several scientific communities and by the requirements of the Large Synoptic Survey Telescope (LSST) [6] in particular, the SciDB project was initiated in the fall of 2008 to develop and deliver a database system designed with those needs in mind. The founders summarize the driving requirements as follows [7]):

1. A data model based on multidimensional arrays, not sets of tuples
2. A storage model based on versions and not update in place
3. Built-in support for provenance (lineage), workflows, and uncertainty
4. Scalability to 100s of petabytes and 1,000s of nodes with high degrees of tolerance to failures
5. Support for "external" data objects so that data sets can be queried and manipulated without ever having to be loaded into the database
6. Open source in order to foster a community of contributors and to insure that data is never "locked up" — a critical requirement for scientists.

The SciDB team identifies as key features of their eventual product its array-oriented data model, its support for versions, provenance, and time table, its architecture to allow massively parallel computations, scalable on commodity hardware, grids, and clouds, its first-class support for user-defined functions (UDFs), and its native support for uncertainty.

The SciDB data model supports nested multi-dimensional arrays—often a natural representation for spatially or temporally ordered data. Array cells can be tuples, or other arrays, and the type system is extensible. Sparse array representation and operations are supported, with user-definable handling of null or missing data.

SciDB allows arrays to be “chunked” (in multiple dimensions) in storage, with chunks partitioned across a collection of nodes. Each node has processing and storage capabilities (allowing shared-nothing operation). Chunk “overlaps” are definable so that certain neighborhood operations are possible without communication among nodes.

The underlying architectural conception is of a shared-nothing cluster of tens to thousands of nodes on commodity hardware, with a single runtime supervisor dispatching queries and coordinating execution among the nodes’ local executors and storage managers.

An array query language is defined, and refers to arrays as though they were not distributed. A query planner optimizes queries for efficient data access and processing, with a query plan running on a node’s local executor/storage manager, and a runtime supervisor coordinating execution.

The Array Query Language (AQL) is a declarative SQL-like language with array extensions. There are a number of array-specific operators, and linear algebraic and matrix operations are provided. The language is extensible with Postgres-style user-defined functions, and interfaces to other packages (Matlab, R, …) will be provided.
3. High energy physics data

High energy physics event data stores typically comprise several successively derived event representations, beginning with raw or simulated data and progressing through reconstruction into streamlined event representations suitable for analysis. Current-generation experiments such as those at the Large Hadron Collider (LHC) [8] deliver data volumes in the tens of petabytes, even before replication. The ATLAS experiment [9] at the LHC provides a representative example. Such stores are more than vast repositories of data [10][11]—they comprise a navigational infrastructure [12], associated metadata both within event store files and external thereto [13], support for both transient and persistent data models and for schema and model evolution [14][15], and associated discovery and selection infrastructure [16][17][18]. For the purposes of this paper we limit ourselves to a description of standard data products and their content.

RAW data are events as delivered by the detector via the ATLAS Event Filter for reconstruction, and are essentially a serialization of detector readouts, trigger decisions, and Event Filter calculations, in bytestream format. Even the RAW data are the heterogeneous output of almost 100 sub-detectors, which are further divided into many layers, sectors and elements, reflecting a complex geometrical structure. The RAW event size is about 1.6 megabytes, arriving at a rate of 200-400 hertz. With the expected duty cycle of the ATLAS detector and the LHC, ATLAS anticipates recording more than three petabytes of RAW data per year.

Event Summary Data (ESD) refers to event data written as the output of the reconstruction of RAW data, and Analysis Object Data (AOD) provides a reduced event representation, derived from ESD, suitable for physics analysis. Reconstruction combines the measurements of all the sub-detectors to produce complex objects, such as tracks, vertices, clusters and jets, which are implemented in C++, using the full expressive power of the language, including multiple and virtual inheritance, polymorphism, templated classes and methods, Standard Template Library and Boost classes, and a variety of external packages. ESD and AOD are stored in POOL ROOT files. A current size estimate for ESD is 1.4 megabytes per event. AOD size is just below 200 kilobytes per event on average.

Event tags (TAG) are event-level metadata records derived from AOD, with content chosen to support efficient identification and selection of events of interest to a given physics analysis or detector performance study. For direct navigation to and retrieval of upstream event data, TAGs store references to event data stored in POOL ROOT and bytestream, in addition to attributes describing event properties, such as the number of jets in an event and their momenta. The representation of TAGs, consisting of only built-in data types, is much simpler than that of other data products.

To facilitate queries for event selection, TAG data are stored in a relational database as well as in files. The TAG size is approximately 1 kilobyte per event. With their small size, their simple data types, and their amenability to storage in relational databases, TAGs provide a natural initial test case for evaluation of fledgling database technologies. Management of upstream data products would require a more mature technology, with support for much more complex data types or for in situ data.
4. Experience

While SciDB’s goals are ambitious, the project is in its early stages, and at the time of these experiments only a preliminary release, Release 0.5, was available for testing. Given this fact, performance measurements would not have been particularly meaningful. Our evaluations instead focused upon exercising the skeleton functionality available in that early release. We chose for our tests a set of event-level metadata records from early LHC proton-proton collision data, far simpler in structure than ATLAS raw and reconstructed data [19]. We developed software to import such records into SciDB’s native storage format. Because an important component of early functionality was support for sparse data, we imported a subset of the data twice, exercising both sparse and dense representations.

There is no natural spatial partitioning of such records into chunks, and temporal partitioning is largely irrelevant, but event selection is readily parallelizable by partitioning the event collection into N disjoint subsets, each independently queryable. For such data, the concept of chunk overlap is irrelevant, but we nonetheless experimented with chunking with and without overlap in our suite of functionality tests.

Importation of data into SciDB was straightforward. The SciDB data model was adequate to support event-level metadata records, though limitations in array nesting caused us to represent certain variable-length arrays as fixed-length records of a maximum length. Such variable-length arrays arise naturally because, for example, the number of electrons or muons or photons or jets varies from event to event, and the properties of those physics objects are an integral part of event-level selection criteria. This is a restriction that should be lifted in subsequent releases, but it is not unlike what is done sometimes for the sake of performance when such data are imported into relational databases.

Every available SciDB operator was exercised and worked well enough, though there were, as one might expect in a preliminary release, some apparent bugs, particularly in mixed sparse/dense array operations. Command line behavior and functionality were primitive, but sufficient to allow testing of the operator suite. Query functionality was sufficient to support simple but important domain-specific selections, such as finding events with at least N jets with energies above a specified threshold.

5. Conclusions and future work

Even in its early stages, SciDB shows promise for array-structured data, and particularly for spatial data. For the derived data that constitute the bulk of most proton collider data, SciDB capabilities may not be a natural match, but for event-level metadata, with support for nesting of variable-length arrays, the SciDB data model may be useful. Some raw data from collider experiments may also be amenable to representation in SciDB, though the heterogeneity of detectors (ATLAS at LHC could be considered to consist of almost one hundred different detectors) will be a challenge. It is likely that high energy physics experiments would profit from the scalable shared-nothing parallelism, though array concepts like overlap, and native array operations, may be less useful. Support of user-defined functions, due in the next SciDB releases, will definitely be of interest: computational/combinatorial operators are routinely used in event-level selection, and are seldom easily or efficiently implemented in relational systems.

There are a variety of emerging technologies that could benefit high energy physics data storage and analysis, including simpler column-wise databases, but also non-database approaches to scalable data
access and analysis (“no SQL” and alternatives) that should also be investigated. It is always a challenge simultaneously to take advantage of third-party technologies and to support efficient domain-specific analysis at multi-petabyte scales. Technologies that support a hybrid approach, allowing, for example, a domain-specific storage format and toolkit like ROOT as a storage backend and a source of plug-in operators, are attractive options [20]. SciDB promises to support such hybrid strategies and will provide explicit APIs for such purposes, and is a technology well worth tracking in the coming years.

6. References

[1] www.vertica.com
[2] root.cern.ch
[3] pool.cern.ch
[4] www-conf.slac.stanford.edu/xldb/
[5] The SciDB Development team, "Overview of SciDB, Large Scale Array Storage, Processing and Analysis," SIGMOD'10 Conference, 2010.
[6] Borne K, Becla J, Davidson I, Szalay A, and Tyson J, 2008, "The LSST Data Mining Research Agenda," AIP Conf. Proc. 1082, pp 347-351.
[7] www.scidb.org/about/history.php
[8] O. Brüning et al., "LHC Design Report, v.1-3", CERN-2004-003-V-1 to CERN-2004-003-V-1-3.
[9] ATLAS Collaboration, "ATLAS Detector and Physics Performance Technical Design Report,” CERN-LHCC-1999-14 and CERN-LHCC-1999-15.
[10] Malon D, 2005, "What your next experiment's data will look like: Event stores in the Large Hadron Collider era," Int. J. Mod. Phys. A20 pp 3871-3873.
[11] Van Gemmeren P and Malon D, 2009, "The event data store and I/O framework for the ATLAS experiment at the Large Hadron Collider,” IEEE Int. Conf. on Cluster Computing and Workshops, p. 1.
[12] Malon D, Van Gemmeren P, Cranshaw J, and Schaffer A, 2006, "Sailing the petabyte sea: Navigation infrastructure in the ATLAS event store,” Computing in High Energy and Nuclear Physics, Mumbai, India.
[13] Malon D, Van Gemmeren P, Hawkings R, and Schaffer A, 2008, "An inconvenient truth: File-level metadata and in-file metadata caching in the (file-agnostic) ATLAS event store,” J. Phys. Conf. Ser. 119 042022.
[14] Malon D, Van Gemmeren P, Nowak M, and Schaffer A, 2006, "Schema evolution and the ATLAS event store,” Computing in High Energy and Nuclear Physics, Mumbai, India.
[15] Malon D, Van Gemmeren P, Schaffer A, Binet S, Nowak M, Snyder S, and Cranmer K, 2008, "Explicit state representation and the ATLAS event data model: Theory and practice," J. Phys. Conf. Ser. 119 042024.
[16] Malon D, Cranshaw J, and Karr K, 2006, "A flexible, distributed event-level metadata system for ATLAS," Computing in High Energy and Nuclear Physics, Mumbai, India.
[17] Cranshaw J, Goosens L, Malon D, McGlone H, and Viegas F, 2008, "Building a scalable event-level metadata service for ATLAS,” J. Phys. Conf. Ser. 119 072012.
[18] Cranshaw J, Cuhadar-Donszelmann T, Gallas E, Hrivnac J, Kenyon M, McGlone H, Malon D, Mambelli M, Nowak M, Viegas F, Vinek E, and Zhang Q, 2010, "Event selection services in ATLAS,” J. Phys. Conf. Ser. 219 042007.
[19] Van Gemmeren P and Malon D, 2010, "Event metadata records as a testbed for scalable data mining", J. Phys. Conf. Ser. 219 042057.
[20] Cranshaw J, Malon D. Vaniachine A, Fine V, Lauret J, and Hamill P, 2010, "Petaminer: Using ROOT for efficient data storage in MYSQL database," J. Phys. Conf. Ser. 219 042036.
Acknowledgments

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.