# REPORT FROM THE 2nd WORKSHOP ON EXTREMELY LARGE DATABASES

***Jacek Becla*[*1] *and Kian-Tat Lim*[2]**

*SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA*
[*1] *Email:* becla@slac.stanford.edu
[2] *Email:* ktl@slac.stanford.edu

## *ABSTRACT*

*The complexity and sophistication of large scale analytics in science and industry have advanced dramatically in recent years. Analysts are struggling to use complex techniques such as time series analysis and classification algorithms because their familiar, powerful tools are not scalable and cannot effectively use scalable database systems. The 2nd Extremely Large Databases (XLDB) workshop was organized to understand these issues, examine their implications, and brainstorm possible solutions. The design of a new open source science database, SciDB that emerged from the first workshop in this series was also debated. This paper is the final report of the discussions and activities at this workshop.*

**Keywords:** Analytics, Database, Petascale, Exascale, Very large databases, Extremely large databases

## 1 EXECUTIVE SUMMARY

The 2nd Extremely Large Databases (XLDB) workshop focused on complex analytics at extreme scale. Participants represented database-intensive scientific and industrial applications, database researchers and DBMS vendors.

**Complex analytics**. Many examples of complex analytical tasks were described. Industrial applications were often in the area of finding and understanding patterns in customer behavior. These industrial analyses frequently use techniques similar to analyses run by scientists for discovering patterns and outliers, such as time series analysis and classification.

Dataset sizes are growing dramatically, and the growth rate is increasing. The largest projects are now adding tens of petabytes of new data per year. Analysis tools such as R, MATLAB and Excel are not keeping pace, forcing analysts to generate memory-sized summaries or samples instead of using all the data. Both the structure of these immense datasets and the techniques applied to them are becoming more complex as well, so XLDB systems must remain flexible in their data representation, processing, and even hardware. An exciting possibility to maximize flexibility while reducing cost, although one requiring some cultural change, is to deliver analytics as a service, using a central XLDB to support distributed and diverse analyst communities. Administrative costs must be kept from scaling with the rapidly growing data sizes, so self-adjusting systems that can keep running normally in the face of hardware faults are required.

SQL's set orientation and low-level ODBC/JDBC interfaces increase the barriers for analysts to use databases. An array-based data model that more intuitively matches the types of data found in science and even in many industries can help break down these barriers. Integration with analytical tools and with familiar procedural languages such as C++ and IDL will also assist. The invention of new languages that more directly capture the analyst's intent is also a possibility, although these face adoption hurdles. The procedural-oriented "MapReduce" camp and the declarative-oriented "database" camps are converging as each grows to understand the benefits of the other.

As analytics become more complex and involve ever larger data sets, the reproducibility of an analytical workflow and its results becomes very important. While provenance and reproducibility have typically been associated with science, industry is now increasingly seeing the need for this feature, which is most easily handled within the database. On the other hand, perfect reproducibility can be unduly expensive or even impossible, so the ability to optionally relax consistency guarantees is also important.

**SciDB**. The initial XLDB activities resulted in an effort to build a new open source science database, called *SciDB*. To date, the SciDB founders have identified initial partners, assembled a database research brain trust, collected detailed use cases, completed initial design, organized funding, founded a non-profit corporation and started recruiting technical talent. The SciDB design is based on a hierarchical, multi-dimensional array data model with associated array operators, including equivalents to traditional relational operations. Queries will be expressed through a parse-tree representation with expected bindings to MATLAB, C++, Python, IDL and other tools and languages. SciDB will run on incrementally scalable clusters or clouds of commodity hardware. Optionally, it will operate on "in situ" data without a formal database loading process. It will support uncertainty, provenance, named versions and other features requested by science users.

SciDB is managed through a non-profit foundation. Design is being done by the brain trust led by Mike Stonebraker and David DeWitt. Broad science involvement and the presence of some high-end commercial users have allowed the team to capture detailed requirements and use cases to help validate the initial design. The effort is supported by large industrial partners such as eBay and Microsoft. The first version of the SciDB system is expected to be available in late 2010.

**Next steps**. It was agreed that SciDB should remain an activity independent from XLDB. A science challenge will be created to enable various XLDB systems, including SciDB, to measure their capabilities against a common standard. The next XLDB workshop is projected to occur at CERN to take advantage of proximity of the VLDB conference in Lyon, France, in August, 2009 (http://vldb2009.org/). Its main goals will include connecting with non-US XLDB efforts and with more science disciplines.

## 2 ABOUT THE WORKSHOP

The 2nd Extremely Large Databases Workshop continued a series providing a forum for discussions related to extremely large databases. It was held at SLAC in Menlo Park, CA, from September 29 to 30, 2008. The main goals were to:

- continue to understand major roadblocks related to extremely large databases with an emphasis on complex analytics,
- continue bridging the gaps within the XLDB community including science, industry, database researchers and vendors,
- build the open source SciDB community.

The workshop's website can be found at: http://www-conf.slac.stanford.edu/xldb08.

The agenda is reproduced in the Appendix.

The workshop organizing committee was composed of Jacek Becla / SLAC (chair), Kian-Tat Lim / SLAC, Celeste Matarazzo / LLNL, Mayank Bawa / Aster Data, Oliver Ratzesberger / eBay and Aparajeeta Das / LG CNS.

### 2.1 Participation

Participation in the workshop was by invitation only in order to keep the attendance small enough to enable interactive discussions without microphones and to ensure an appropriate balance between participants from each community.

The workshop was attended by 64 people from industry (database users and vendors), the sciences (database users) and academia (database research). Compared to the 1st workshop, the 2nd workshop had significantly higher attendance from the academic and database research communities.

The names and affiliations of all the attendees can be found on the workshop's website.

## 2.2 Structure

The bulk of the time was spent in highly interactive discussions focusing on complex analytics and administrative features needed by extremely large database setups. A significant fraction of the workshop was spent on SciDB, a new database project that was initiated as a result of the 1st XLDB workshop and its follow-up activities. A small fraction of the workshop was devoted to "war stories" from CERN/LHC, Pan-STARRS, and eBay in which lessons learned from these users currently struggling with extremely large datasets were presented.

## 2.3 About this report

The structure of this report does not map directly to the agenda, as we attempted to capture overall themes and the main threads of discussion.

Sections 3 through 5 cover complex analytics, with emphases on examples, effects on data representation and effects on processing, respectively. Section 6 discusses SciDB. Section 7 documents the consensus on the next steps to be taken and the future of the XLDB workshops.

We have intentionally de-emphasized the names of specific projects and participants in order to draw out the commonalities and differences within and between the scientific and industrial communities.

# 3 COMPLEX ANALYTICS — INTRODUCTION

The focus of this workshop was intended to be on complex analytics using extremely large databases. It was pointed out that even relatively simple computations become complex when applied to peta-scale datasets. The goals were to explore beyond these ordinary statistics and aggregations to determine what the needs of sophisticated scientists and business analysts are and how these needs are affecting the structure and usage of XLDBs.

Many examples of complex analytical tasks in industry were in the area of understanding and finding patterns in customer behavior. Such patterns, or in some cases the exceptions to the patterns, can be used for many business purposes. These tasks may include targeting advertising and promotions, predicting churn, detecting spam, finding fraud, and analyzing social networks. In combination with controlled experiments, complex analytics may be used to improve products by determining the effects of a change on both behavior and revenue.

In science, similar tasks include analyzing astronomical spectra and positions; crunching high-throughput genomics and proteomics data; pulling together climate data from networks of hundreds of thousands of sensors; comparing computational simulations of the ocean, earthquakes or combustion dynamics; and sifting through the results of fusion and collider experiments. Not all of these sciences are using databases extensively today for storage of raw and derived data, but they all have large datasets in the terabyte or larger range with some collections of datasets reaching the petabyte scale. A wide variety of techniques is applied to the data, ranging from coordinate transformations on the raw data to advanced machine learning algorithms applied to derived attributes.

Scientific and industrial analytical methods overlap substantially. Both groups use statistical techniques, classification algorithms and time series analysis. Both are interested in finding outliers that do not fit patterns, while also using the data to determine those patterns.

The next two sections describe common issues across science and industry with regard to complex analytics on XLDBs. Section 4 discusses issues related to the analyst's view of the database in terms of data representation and the query interface. Section 5 continues with topics related to the processing of data internal to the database.

# 4 COMPLEX ANALYTICS — DATA REPRESENTATION

## 4.1 Scale and cost

Not only are dataset sizes growing dramatically, but even the rate of growth seems to be increasing. Industrial dataset sizes are reaching tens of petabytes per year, with raw data in the tens of terabytes per day. Scientific dataset sizes are in the same range, with CERN planning to store 15 petabytes per year at similar daily data rates. While these are the largest databases, virtually all participants are working with existing and planned systems in the 0.1 to 10 petabyte range.

There are several approaches to scalability of data warehouse systems available from various providers, but the scalability of analysis tools may not be keeping pace. Many of the analyses that are desired are the kind that can be done by statistical tools like R or SAS, or even Excel, which is perhaps the most popular, but these can only be applied if the massive data can be reduced to a set of derived statistics that fit in memory. The cost of doing analysis may actually be increasing per unit of data as algorithms get more complex and vendor licenses fail to take these scales into account.

At these sizes, maintaining complex normalized relational schemas can be difficult and expensive. Many projects have compromised by storing data in files or unstructured strings with the metadata and sometimes derived data being the only components stored in a traditional RDBMS.

It was pointed out that even free software is not free. All software entails maintenance and operations costs, in the form of contracts or personnel that can be substantial.

## 4.2 Complexity

Just as scale is increasing, the complexity of analyses is also increasing.

First, the structure of the data is becoming more complex, engendering similarly complex processing. Observations are acquiring a host of attributes to capture conditions that cannot be reconstructed; storing transient search results for Internet queries or postage-stamp images for astronomical detections are examples of this. Time series, for which the order and spacing of events are significant, are becoming prominent. Much scientific data and increasing amounts of industrial data have spatial attributes, so multi-point correlations in space and time are needed. Uncertain data adds uncertainty ranges and interval calculations to queries, further increasing complexity.

Second, the analyses themselves are becoming more complex. Transformations between coordinate systems (re-gridding) may be required. In many cases, well-understood derived data products such as key performance metrics are being produced by highly-optimized processes to allow simple tools to do basic analysis, but this is insufficient for today's analysts, who need to be able to explore the data in more detail, integrating raw data with the derived data in an *ad hoc* manner, in order to discover new patterns and new metrics.

## 4.3 Flexibility

Complete, accurate requirements are rarely known ahead of time for any usage of complex analytics. Scientists often do not know exactly what they are looking for. Industrial needs can rapidly change. Analytical systems must accordingly be built very flexibly in order to handle unknown requirements. In many ways, the new, unprecedented analysis is precisely what XLDBs are designed to enable.

As the data being stored become more complex and more structured, their variability also increases. Accordingly, schema flexibility, in particular the ability to add new data attributes easily and cheaply, is a critical aspect of XLDB operations. These new attributes may stem from the raw data, from new derived metrics, or from end-user annotations.

Capabilities of systems at the leading edge of technology are further enhanced as that technology advances. New developments such as multi-core computing, access to large amounts of RAM, flash storage, fast networks, or vastly increased disk I/O bandwidth can enable radically different analytical techniques. The systems must be able to cope with the introduction of these new methods over time.

## 4.4 Data models

A variety of data models could be used to represent information in XLDBs. Possibilities include relational tables, objects, streams, arrays, graphs, meshes, strings (e.g. of DNA or amino acid sequence), unstructured text, and XML. None of these is perfect for any given dataset, and it is very hard for a single system to support all of these well. The relational model has succeeded for decades as a reasonable compromise between, on one hand, flexibility and representational power and, on the other, limiting the set of operators to permit an optimizer to work well. Object databases and MapReduce systems move further to one side of the spectrum, permitting great representational capability and infinitely flexible operations at the cost of little optimization.

Many participants felt that representing data in the form of arrays could be a useful step forward beyond the relational model. First-class arrays can perform much better than arrays simulated on relational tables. Arrays are a natural, intuitive data model for many sciences, including astronomy, fusion physics, remote sensing, oceanography and seismology. Typically, a small number of (physical) dimensions is needed, often no more than four (x/y/z/time or right ascension/declination/spectral frequency/time). It will be important to support irregular or ragged arrays that have varying numbers of elements in each dimension.

Arrays inherently have ordering properties for their elements. This is of some interest to industry, which can use the ordering and spacing of array elements to represent event sequences.

Some sciences require more specialized data structures. Biology works with sequences; chemistry uses graph and network structures. Both of these could be simulated using tables or arrays, but likely at some cost.

## 4.5 Interfaces

The final aspect of data representation is the interface by which the analyst gains access to the data. As mentioned above, end-user statistical tools such as R, MATLAB and Excel, which can only be used with small, summary datasets today, need to be integrated with databases to make use of their exploratory capabilities while fully exploiting the scalability of the XLDB back-end.

Beyond a tools interface, science and industry both use procedural languages such as C++, IDL, and even FORTRAN in order to write advanced analytics. Interfaces to these languages that are more natural than low-level ODBC or JDBC would substantially enhance analyst productivity. Microsoft's LINQ was mentioned as an interesting example. An alternative path is to build a new language, such as Sawzall or Pig Latin, that provides a procedural structure, but concern was expressed over whether such a language could achieve widespread adoption.

Participants recognized the need to define basic operators for manipulating arrays, especially if they are to be the basis of a new data model. They foresee that the right set of powerful primitives for tasks such as time series analysis could enable a few simple lines to replace pages of unwieldy SQL code. The specialized graph processing operations available for AT&T's Daytona system are an example.

Furthermore, an interface must be defined internal to the database to allow new operators to be defined. This interface, which might take the form of an existing language, must allow exploitation of the parallelism of the XLDB system by moving computation to the location of the data. Techniques such as the translation of Daytona's Cymbal language to C code for compilation and execution may be useful in this area.

# 5 COMPLEX ANALYTICS - PROCESSING

## 5.1 Architecture

For the largest-scale datasets, there is no debate that computation must be moved close to where the data resides, rather than moving the data to the computation. On a micro level, this suggests a shared-nothing style architecture, which is common to many existing XLDBs, e.g. those implemented using Teradata or the Pan-STARRS GrayWulf system.

On a macro level, there are sometimes difficulties, because, as one project put it, data must move to where the funding is. Prevailing cultural attitudes in both industry and science desire ownership of and control over data, sometimes for competitive reasons. Nevertheless, having a central shared analysis platform can provide great benefits. It avoids a proliferation of data marts, each of which requires valuable administrative resources. Providing analytics on the platform as a service seems to be an exciting way forward. It allows dynamic adjustment of usage on demand as workload changes driven by external factors such as conferences or quarterly reports. It can help maximize utilization of expensive resources while allowing the full capabilities of those resources to be devoted to important problems, thereby reducing the time to discovery. Further centralization does not preclude having rigorous access controls.

There continue to be debates between a "brute force" camp that emphasizes the MapReduce framework and full table (or column) scans and a "database" camp that emphasizes the capabilities of an optimizer. MapReduce has advantages in fault tolerance, progressive results that allow mistakes to be caught rapidly and simplicity of the programming model, but it requires programming, is more suited for batch processing than interactive queries, and can be inefficient in its use of resources. Both sides agreed that there appears to be a convergence between them, with aspects of each model being adopted by the other.

## 5.2 Reproducibility

As analytics become more complex, it is becoming even more important to be able to reproduce an analytical procedure and its results. This involves more than just tracking what happened or maintaining the lineage of metadata; the lineage of the data itself and the ability to use older versions of them is critical, as running an old process on updated data can generate different output. This capability also allows erroneous derived data to be tracked to its source or, vice versa, erroneous raw data to be tracked to all values derived from them. While provenance and reproducibility are typically associated with science, industry is increasingly seeing the need for this feature, in some cases due to new legal requirements.

It is easiest to capture provenance for operations occurring within the database. It is rare for all operations of interest to be handled by the database, however. In many cases, external systems are used for processing raw data sources or as "black box" computational packages. In these cases, provenance information from outside and inside the database must be joined. Loading external provenance into the database's internal structures is one possible approach that offers a unified query capability; another approach would be to export provenance information from the database into a standard format usable by external provenance tools.

Maintaining the provenance of data at this level comes with a price. It was pointed out that XLDB systems typically have large numbers of disks in order to have enough spindles to maintain sufficient I/O bandwidth. In typical usage, only 10% or so of those disks are actually useful for "hot" data. The remainder of the disk space may be used for storing provenance and versioning information.

At the same time, a strong argument was made that perfect reproducibility is a mirage that is not necessary all the time, particularly during exploratory analysis. First, the computational environment outside the database such as the hardware, the operating system, compiler libraries and so forth may affect results. While this environment can be recorded, reproducing a given configuration may be unduly expensive or even impossible. Second, there may be great cost savings in relaxing the accuracy guarantees. Given the probability of failures in hardware in these large-scale systems, the ability to execute analyses on incomplete data, dropping a few elements in an unbiased fashion, may be highly desirable. Similarly, it may be possible to provide greater responsiveness (see below) and performance by relaxing the standard ACID (atomicity, consistency, isolation and durability) criteria used with transactional relational databases. In such cases where incomplete or potentially inconsistent data are used, the system must give an indication of the uncertainty in the result.

## 5.3 Workflows

There are several occasions in the course of a complex analysis when workflow management may be necessary. Initially, as data is loaded into a system, processing may occur to transform the data from their raw form into a

suitable long-term persistent format. A "cooking" process may be used to turn raw data into more-easily-analyzed derived data. Finally the analysis itself may require multiple steps.

Tracking these workflows is essential for provenance and reproducibility, as in the previous section. Allocating resources to them in an appropriate fashion is also important. Industrial systems often have strict requirements in order to meet service level agreements, but even scientific systems must ensure fair sharing of the available resources. Managing workflows inside the database is the most powerful paradigm in terms of maintaining provenance and allowing for potential optimizations such as bringing the computation to the data. In many cases, however, bringing all of the processing inside the database will be impossible, and so the database must be able to integrate with external workflow management systems.

## 5.4 Responsiveness

Despite the immense scale of XLDBs, many end users need rapid response capabilities. Businesses need to react to events happening in real time, and scientists need to respond to interesting transients that may have limited observation windows. In many cases, processing raw data as they are collected or loaded into an XLDB is used to trigger these responses, but it would be desirable to bring the full power of the database system to bear on fresh data.

Another aspect of responsiveness is the ability of a system to deliver partial and progressively more accurate results. Having such a capability, rare if not nonexistent in traditional SQL RDBMSes, would allow erroneous queries to be recognized faster and correct queries to be terminated early when their accuracy reaches desired levels.

## 5.5 Administration

At extremely large scales, administrative costs that increase with the size of the data are unsustainable. Systems must be self-sustaining, self-healing, self-load-balancing and self-adjusting to avoid requiring a large number of database administrators. In the experience of the participants, detailed monitoring of large systems at every level is essential. Applying the power of the database to the task of analyzing its own logs is highly rewarding. The feedback loop between database performance metrics and database configuration helps to automate index management and reduce data skew, but the loop should not totally exclude human oversight, as transient load anomalies could otherwise cause long-term problems.

Resources used by a database need to be managed carefully. Because accurate cost estimation is often difficult due to the presence of many correlated factors and data dependencies, it is desirable for a system to be able to catch unproductive queries rapidly during execution and also to pause and resume queries that are temporarily exceeding their resource quotas. This naturally leads to a fine-grained, operating-system-like priority scheme, as implemented in several systems of workshop participants.

Fault tolerance is essential when large numbers of computers are involved in a single system. A particular concern was failures of complex, long-running user-defined functions. While a database may handle fault tolerance in its own operations, providing fault tolerance to user code, perhaps by including checkpointing capabilities in the programmer interface, may be more difficult.

# 6 SCIDB

The 1st XLDB workshop clearly exposed the lack of shared infrastructure needed by large-scale database users. Each data-intensive community, with very few exceptions, "rolls its own" software on top of the bare operating system. This results in building software with almost no applicability to other projects. While data-intensive industrial users with extensive financial resources can afford building custom solutions, such an approach is not sustainable inside science. To address this issue, the recommendations from the 1st workshop included:

a) improving collaboration between science and database research and
b) defining common database requirements shared by different science domains.

A follow-up Science-Database mini-workshop organized at Asilomar, CA, in March of 2008 resulted in a decision to build a new open source science database, called *SciDB*. Since then, the SciDB founders have identified initial partners, assembled a database research "brain trust," collected detailed use cases, completed an initial design, organized funding, founded a non-profit corporation and started recruiting technical talent. This chapter highlights the initial SciDB design and some of the aforementioned activities.

## 6.1 Science needs

The 1st XLDB workshop, the Science-Database mini-workshop, and the use cases supplied by the initial partners and lighthouse customers all highlighted the fact that science database users are almost universally unhappy with relational DBMSes. The following main reasons include many described above:

**Wrong data model:** science data very rarely naturally fits into a relational table-based model. The ideal model varies somewhat by science, but in each case, it is far from pure tables, and simulating it on top of tables is extremely inefficient.

**Wrong operators:** the most frequently performed operations, for example regridding or Fourier transforms are nearly hopeless in relational DBMSes. Similarly, frequently performed complex analytics such as time series analysis are impossible to express in SQL.

**No provenance:** all scientists want a DBMS to support provenance and reproducibility.

**No time travel:** science users must be able to reproduce published results, hence overwriting previous data as is done in current DBMSes is not an option.

**Insufficient scalability:** some scientific projects have already reached the petabyte level, and others are quickly approaching it. None of the existing DBMSes offers multi-petabyte level scalability. The situation worsens when the costs of licensing and, in some cases, specialized hardware are taken into account.

## 6.2 Design

Building a system supporting all of the features requested by science requires substantial development at the lowest levels of the system, such as the storage manager. This requirement makes it difficult to build on top of existing open source systems (DBMS or MapReduce). While it is expected that some relevant pieces of existing appropriately-licensed software can be reused, the overall design of SciDB has to be from the ground up. This section highlights the key decisions made, the technological challenges to be overcome, and future plans in the area of SciDB design.

## 6.3 Data model

It is clear there is no single universal data model that would make every science happy. After evaluating which models are applicable to the largest number of users, which models are most realistic to implement and which models can be easily implemented on top of other models, the array model has been selected. This model is a very good fit for most sciences, including astronomy and many branches of geoscience including oceanography, remote sensing and atmospheric sciences.

SciDB will support nested multi-dimensional arrays. Two types of arrays will be supported:

- basic arrays (MATLAB style), with integer dimensions starting with 0. Dimensions can be bounded or unbounded.
- enhanced arrays, where a user-defined shape function defines the outline of the array. Enhanced arrays can be irregular in any dimension.

Each element of an array will contain a tuple of attributes, similar to a row in a relational database. Any of the attribute values can be a nested array.

It is expected array-based data will compress well. Techniques such as storing deltas or run-length encoding will be used to provide maximum possible loss-less compression.

## 6.4 Query language and operators

A parse-tree representation will be used to define the SciDB query language. Bindings to commonly used tools, such as MATLAB, C++, Python, IDL and others are expected to be built. The C++ binding will likely be the first one available because of its usefulness for internal development. It is expected that the community will help with some of the additional bindings.

SciDB will support standard relational operators, such as filter or join, plus many other commonly-used array operators. The complete list of such operators has not been determined yet and will require polling different scientific communities. Frequently mentioned examples include regridding and Fourier transforms. In addition to natively supported operators, users will be able to define their own operators, PostgreSQL-style.

Some of the research topics in this area include how to represent array operations in languages like C++ or Python. On one hand it would be useful if the syntax would look similar in all languages, but on the other hand some tool languages already have well defined array-based operators.

## 6.5 Infrastructure

SciDB will run on incrementally scalable clusters or clouds of industry standard hardware, ranging from a single laptop through small clusters managed by individual projects and laboratories to very large commercial clouds. It will have built in high availability and failover and disaster recovery. Data will be partitioned across the available hardware to maximize throughput.

**In situ processing**
SciDB will optionally operate on data "in situ", i.e. on external data not loaded into SciDB. Such data will not have some of the SciDB services such as replication or crash recovery. SciDB will provide a means for describing the contents of this type of data. In addition, adapters for the most popular array formats such as HDF-5 or NetCDF will be implemented.

**Uncertainty**
Based on input received from scientists, virtually all sciences make some use of uncertainty. Beyond that, the uncertainty-related requirements vary significantly. For that reason, SciDB will initially support the specification of uncertainty and their use in simple computations such as comparisons. Even this level of support is non-trivial to implement. More sophisticated error models may be implemented later; once more commonalities across different sciences are identified.

**Provenance**
To capture provenance, the SciDB engine will record every update it executes. In addition, it will be possible to load external provenance through special interfaces, and export of provenance will be supported. In combination with versioning features for data, it will be possible to reproduce the conditions of any operation and trace its inputs and effects. It is expected that special facilities for querying provenance will be developed.

## 6.6 Project organization and current status

The SciDB organization is based on a partnership between three groups:

- Science and high-end commercial users. Their tasks include providing use cases and reviewing the design.
- A database research brain trust. The tasks of this group include designing the system, performing necessary research and overseeing software development.
- A non-profit foundation. The tasks of the foundation include managing the open source project and providing long-term support for the resulting system.

The initial science partners include Large Synoptic Survey Telescope/Stanford Linear Accelerator Center (now SLAC National Accelerator Laboratory) (LSST/SLAC), Pacific Northwest National Laboratory (PNNL), Lawrence Livermore National Laboratory (LLNL), and the University of California at Santa Barbara (UCSB). The first lighthouse customers are LSST and eBay. Some of these teams have already provided an initial set of use cases. A science advisory board has been created to coordinate input from science, translate use cases from scientific terminology into database terminology and prioritize the requested features.

Industrial partners include eBay, Vertica, and Microsoft.

The brain trust team members include Mike Stonebraker (MIT), David DeWitt (University of Wisconsin → Microsoft), Jignesh Patel (University of Wisconsin), Jennifer Widom (Stanford), Dave Maier (Portland State University), Stan Zdonik (Brown Institute), Sam Madden (MIT), Ugur Cetintemel (Brown University), Magda Balazinska (University of Washington) and Mike Carey (UC Irvine). The brain trust has already produced an initial design and is currently in the process of refining and improving it.

## 6.7 Timeline

It is expected that a demo version of SciDB system will be available in late 2009, and the first production version ("V1") in late 2010.

## 6.8 Summary

The SciDB project sparked a lot of interest among workshop attendees. It was unanimously agreed that it is an ambitious project, and therefore in order for it to succeed it must initially stay focused on a minimum set of well-defined core features. It is to be built from the ground up, so users should not expect "Teradata-like bells and whistles" in one or two years — it may take a long time to get to truly good performance.

# 7 NEXT STEPS

The last section of the workshop was devoted to discussions about the future. The two main topics discussed were (1) the future relationship between SciDB and XLDB and (2) the future of the XLDB workshops.

## 7.1 SciDB and XLDB

SciDB came about as a result of the 1st XLDB workshop. It is clear there is substantial overlap between XLDB and SciDB, but it was also obvious that SciDB design and development is not appropriate for a big and diverse environment like XLDB and should thus be an independent activity. The attendees agreed that it was very valuable to devote a large fraction of this workshop to SciDB to bring the XLDB community up to date. It was also agreed that it would be very useful to keep the XLDB community aware of SciDB activities on a regular basis but that this need not be as great a proportion of the workshop time in the future.

It was suggested that the SciDB team should publish its collected use cases for other vendors to study. The attendees also advised that SciDB should reach out to more sciences.

One of the hot topics related to the future of SciDB was a measure of success. The conclusion was that some metrics of success should be defined early on and that the design should be periodically validated against such measures.

## 7.2 Science challenge

It was agreed we should define a science-oriented database challenge. Because this was envisioned to be different from benchmarks such as the TPC-x series, the word *challenge* was preferred over *benchmark*. Some believed a challenge based on the Sloan Digital Sky Survey data set and queries, which are well characterized and understood,

would be a comprehensive measure. In the end, however, it was believed that this task has been overly optimized for relational engines. Mike Stonebraker and David DeWitt agreed to take on the task of defining a science challenge.

## 7.3 Wiki

The XLDB Wiki was briefly discussed. The attendees felt it should be made more visible and publicized better. An active moderator with some degree of journalistic skill will likely be required to maintain an attractive site. We will attempt to recruit one.

## 7.4 Next workshop

There was unanimous agreement that we should continue with XLDB workshops. The workshops have a very well-defined, unique focus and are a "great place for interactions" between database users and database community members (researchers and vendors). It was also pointed out there is no overlap between XLDB and other well established conferences and workshops; while other groups (such as SSDBM) have tried to create similar forums, they have never fully succeeded.

Most attendees felt that the XLDB workshops should continue as annual meetings. Two days turned out to be an ideal length. We will continue with the by-invitation-only rule to foster discussion and frank exchange. XLDB3 should continue to seek to be interactive, although it was suggested that we might introduce several presentations or papers to drive the discussion.

Some felt it might be advisable to attach the next XLDB to another database venue like SIGMOD or VLDB, however in the end the conclusion was to continue with an independent workshop. It was clear to everybody that the larger database community gathered around VLDB/SIGMOD conferences would benefit from hearing about real science requirements, and therefore we should try to organize a tutorial at the next VLDB conference, but that would not substitute for a real XLDB workshop.

Participants agreed that we should try to reach out to XLDB communities outside of the USA. Large scale, database-focused activities are happening in Europe (e.g., MonetDB), and in Asia (with Japan and China specifically mentioned). For that reason, locating XLDB3 in Europe or Asia was considered. Most felt the best location would be at CERN immediately before or after the VLDB conference which will be held in August 2009 in Lyon, France, near CERN. The XLDB3 location will be finalized once CERN officially confirms it can host the workshop. Assuming these plans hold, Europeans Maria Girone/CERN and Martin Kersten/CWI would take charge of the organizing committee; at least one of the two key organizers of the first two XLDB workshops should also be involved.

The attendees pointed out that we have not had sufficient representation from several major scientific communities. In particular biology was underrepresented; the biology representatives present at XLDB2 were all database people who had worked with biologists.

Two important goals for XLDB3 were identified:

- Connect with non-US XLDB efforts.
- Connect with more science disciplines and communities.

XLDB3 should also include a short report from the SciDB project. Another possible agenda item mentioned was "embedded sensors and RFIDs." One possible structure for the workshop might be to spend the first day on existing engines and solutions with a focus on systems developed in Europe and then spend the second day on discussing how to move the state of the art forward.

Participation of DBMS vendors in future XLDB workshops was encouraged; keeping the vendors aware of large scale problems and needs stimulates research and ultimately will result in more DBMSes supporting useful XLDB features.

It was also agreed that it would be beneficial to make the XLDB3 agenda available well in advance, e.g. around December 2008.

## 8 ACKNOWLEDGMENTS

The organizers gratefully acknowledge support from the sponsors:
- eBay
- Greenplum
- Facebook
- LSST Corporation.

## 9 GLOSSARY

CERN – The European Organization for Nuclear Research
DBMS – Database Management Systems
HEP – High Energy Physics
LHC – Large Hadron Collider
LLNL - Lawrence Livermore National Laboratory
LSST – Large Synoptic Survey Telescope
MIT – Massachusetts Institute of Technology
Pan-STARRS – Panoramic Survey Telescope & Rapid Response System
RDBMS – Relational Database Management System
SDSS – Sloan Digital Sky Survey
SLAC – SLAC National Accelerator Laboratory, previously known as Stanford Linear Accelerator Center
SSDBM – Scientific and Statistical Database Management Conference
UCSB – University of California in Santa Barbara
VLDB – Very Large Databases
XLDB – Extremely Large Databases

## 10 APPENDIX – AGENDA

| Time | Duration | Speaker/Moderator | Title | Type |
|---|---|---|---|---|
| **Monday, September 29** | | | | |
| 09:00 AM | 00:10 | Steven Kahn | Welcome | |
| 09:10 AM | 00:30 | Jacek Becla | XLDB1, status update, XLDB2 goals | Presentation |
| 09:40 AM | 00:20 | Maria Girone | War stories (LHC) | Presentation |
| 10:00 AM | 00:20 | Oliver Ratzesberger | War stories (eBay) | Presentation |
| 10:20 AM | 00:15 | | Coffee break | |
| 10:35 AM | 00:20 | Jim Heasley | War stories (PanSTARRS) | Presentation |
| 10:55 AM | 00:20 | Kian-Tat Lim | Science commonalities and SciDB intro | Presentation |
| 11:15 AM | 01:00 | Mayank Bawa | Complex analytics - industry perspective | Panel |
| 12:15 PM | 01:00 | | Lunch (provided) | |
| 01:15 PM | 01:45 | Jacek Becla | Complex analytics – science perspective (astronomy, bio, fusion, remote sensing, HEP. Each gets 20 min) | Panel |
| 03:00 PM | 00:15 | | Coffee break | |
| 03:15 PM | 00:30 | Kian-Tat Lim | Complex analytics – data models | Moderated discussion |
| 03:45 PM | 00:30 | Michael Carey | Complex analytics – interfaces | Moderated discussion |
| 04:15 PM | 00:40 | David DeWitt | Complex analytics – specific topics TBD | Moderated discussion |
| 04:55 PM | 00:05 | Jacek Becla | Dinner logistics | |
| 05:00 PM | | | Adjourn | |
| 07:00 PM | | | Dinner at the Sheraton in Palo Alto | |
| **Tuesday, September 30** | | | | |
| 09:00 AM | 00:20 | Oliver Ratzesberger | Admin features needed – list and priorities | |
| 09:20 AM | 01:00 | Oliver Ratzesberger | Admin features – specific topics TBD | Moderated discussion |
| 10:20 AM | 00:15 | | Coffee break | |
| 10:35 AM | 01:40 | Oliver Ratzesberger | Admin features – specific topics TBD | Moderated discussion |
| 12:15 PM | 01:00 | | Lunch (provided) | |
| 01:15 PM | 00:45 | Michael Stonebraker | SciDB – design | Presentation |
| 02:00 PM | 00:45 | | SciDB | Q&A from audience |
| 02:45 PM | 00:15 | | Coffee break | |
| 03:00 PM | 01:20 | Richard Mount | Future | Moderated discussion |
| 04:20 PM | 00:10 | Jacek Becla | Closeout | |